Some Invariants of Directed Topology towards a Theoretical Base for a Static Analyzer Dealing with Fine-Grain Concurrency

– Habilitation thesis –

Emmanuel Haucourt

October 27, 2016

Foreword

This memoir is an account of my work and centers of interest during the last decade. As a research engineer at CEA LIST, I've always been alternating between theory and applications. The ALCOOL software would not have existed without the thorough theoretical work upon which it is based. Conversely, many theoretical problems I am interested in arise from practical issues.

The practical roots of my work lie in parallel automata (*i.e.* parallel composition of sequential processes that communicate via a shared pool of resources). In this context no instance of a process can be spawned nor killed at runtime. Nor the characteristic of a resource can be altered during the execution of the program. Pointer arithmetic is moreover forbidden. These restrictions allow us to define an analog of the control flow graph for any program of a certain language. The programs written in this language are actually special cases of higher dimensional automata (*cf.* Pratt (1991)). From the precursory work of Dijkstra (1965) and the previous observation, one is led to consider directed topology as a natural place for studying the control flow structures of such programs.

I owe much of my knowledge about local pospaces and fundamental categories to Lisbeth Fajstrup, Eric Goubault, and Martin Raußen with whom I regularly collaborate.

I was initiated to the subtleties of streams by their inventor himself, Sanjeevi Krishnan, during his post-doctoral stay at the CEA.

My interest in unique decomposition theorems goes back to a collaboration with Thibaut Balabonski in 2006 during his three months internship. In 2012, Nicolas Ninin also wrote his master thesis upon related questions.

Contents

Introduction

This chapter sketches a rough picture of my work: which branch of computer science it is related to, what practical situation it applies to, and what theoretical ground it is based on.

Static Analyzers and Compilers

Static analysis encompasses all the methods by which one extracts information on the behaviour of a program from its source code. It relies on software, namely static analysers, that builds mathematical abstractions of the program under study. Any static analyser is thus associated with a programming language, that is to say a parser (*i.e.* a syntaxic analyzer which recognizes the grammar of the language and generates the abstract syntaxic trees – AST for short), and a semantics (*i.e.* the mathematical description of the language), which provide the raw material for building the models. A chain of transformations (handling internal abstract representations aka middle-end representations) is indeed applied to the AST at the end of which one obtains the expected abstraction. In particular one may substitute the parser with another one so another programming language can be taken into account without necessarily changing the whole chain of transformations. Therefore in some sense, static analyzers and compilers belong to the same family: the latter generate executable files E while the former return mathematical objects M yet both work by successive transformations of middle-end representations. One even finds domain specific languages allowing the user to alter the generation process by directly handling the middle-end representations (*e.g.* the MELT language allows one to deal with the GCC internals aka "Gimples" – see Starynkevitch (2012)).

Control Flow Graphs and Flowcharts

The concept of Control Flow Graph (or CFG for short) was introduced by Allen (1970) with a view toward optimization of sequential programs, and quickly became a founding notion in compiler middle-ends design. For example the internals of the current versions of both LLVM and GCC follow this approach. According to the analogy between compilers and analyzers, control flow graphs became a key ingredient of static analysis. The collection T of all possible execution traces of a program is finitely described by nature. However its obvious and faithful description, the source code itself, may not be convenient for automated handling. The CFG is thus intended to be an overapproximation of T in the following sense: any execution trace is associated with a path on the CFG. In the sequential realm the definition of the CFG is rather clear: a node is

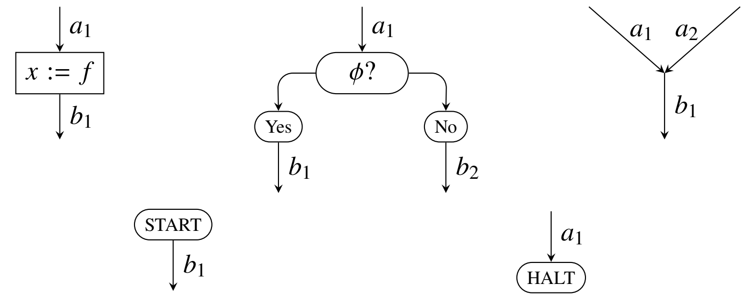


Figure 1: Basic elements of the flowchart language.

provided for each control instruction so the execution is entirely determined from the initial state of the system (*e.g.* the input provided by the user). A sequential program is in particular characterized by the fact that it has a single instruction pointer. From an abstract point of view a sequential program can therefore be defined as a pointed graph G labelled by endomorphisms of a given set X which should be thought of as the state space of the system. In some sense the structure G is intrinsic, indeed any programming language proposes at least two features to define the control flow: the **branching** and the **loop** constructions which often follow the syntax thereafter. Loosely speaking, these instructions shape the control flow graph.

```
if condition                       while condition do
then code                            code
else code                          done
```

Flowcharts, which were introduced by Floyd (1967), are intended to provide programs with semantics. Formally, they are very close to control flow graphs though they are related to the static analysis of programs instead of their compilation. Flowcharts are built from five basic elements shown on Figure 1. Each of them is associated with one of the basic features shared by all the sequential programming languages, namely the assignments, the conditional branchings, the join, and the entry and terminating points.

Dynamics

The structure described in the preceding section comes with its dynamics. The distinguished vertex of the control flow graph is the **initial state** (*i.e.* the one upon which the **instruction pointer** stands at the beginning of the execution). The current state of the system is then updated according to the action carried by the first arrow whose condition is satisfied: the set of arrows is supposed to be totally ordered. If no condition is satisfied then the execution is over. As for dynamical systems, the behaviour (*i.e.* the sequence of states visited during the execution) of the program is entirely fixed by the choice of the initial point. Sequential programs are therefore said to be **deterministic**. The state of the system at the end of the execution is the result of the **computation**.

What matters here is that not only two terminating executions starting at the same vertex will return the same result, but they will go through the same sequence of states.

Concurrency

From the mid sixties it became clear for computer scientists that concurrency would be both a major feature of programming languages and the source of severe and unpredictable errors – see Naur and Randell (1969); Randell (1979). Indeed Dijkstra (1968) points out several practical problems that are seemingly simple but whose "obvious" solutions are drastically wrong for subtle reasons. According to (Hansen, 2002, p.7-12) the previous article appears on the top of the list of publications that founded concurrent programming theory which is thoroughly explained in a book by Michel Raynal (2013). A system is said to be concurrent when it is made of several agents acting simultaneously. Applying this loose definition in the context of the previous section, a program is said to be concurrent when it may have several instruction pointers wandering on separate control flow graphs. This point of view was implicit in Dijkstra (1968) before Coffman et al. (1971) and later Carson and Reynolds Jr. (1987) definitely formalize the notion of **progress graphs**. We thus represent a concurrent program by the collection of the control flow graphs associated with its many processes. A separate analysis of the constituent processes cannot however produce a fine analysis of the global behavior of the program. This is to be compared with the law of probability of a tuple of random variables which cannot, in general, be fully recovered from the laws of probability of its components – unless they are independent. The control flow graphs of the constituent processes then should be gathered in a single structure giving an account of the interactions between them. From the idea that each process has its own instruction pointer wandering on its own control flow graph emerges a correspondence between the number of concurrent processes and the number of control flow graphs. The representation of a concurrent program with d processes then should be some mathematical structure M of dimension d. One of the seminal remarks of the domain is that, under mild hypotheses, the interplay between the processes of the program are represented by holes in M. The study of such structures is at the core of algebraic directed topology. For a detailed account of the rise a topological methods in concurrency theory, see Goubault (2000).

Directed Topology

Therefore it seems flourishing to go in search of a generalization of the notion of graph that enables the concept of higher dimension. Mathematics provide an obvious candidate, namely the precubical sets, which can be defined as cubical sets without degeneracies (*cf.* Goubault (2003)). In particular they led Pratt (1991) and van Glabbeek (1991) to introduce higher dimensional automata, a model of concurrency which encompasses many (if not all the) standard combinatorial models of concurrency – Winskel (1995); van Glabbeek (2006); Goubault and Mimram (2012). Shortly after they were introduced, higher dimensional automata became the subject of theoretical papers (Goubault and Jensen (1992); Goubault (1993)) and were used to provide toy languages like λ-Linda with a semantics – Cridlig and Goubault (1993). Nevertheless the combinatorial nature of precubical sets make them poorly fitted with the fundamental speed independence hypothesis made in Dijkstra (1968). Modelling concurrency by means of combinatorial structures implicitly imposes the existence of a global clock.

A way of tackling the problem is the use of continuous models (*i.e.* topological spaces built from the real line $\mathbb{R}$ and its iterated Cartesian products). With respect to modelling concurrency, topology however lacks expressivity: any programming language comes with an implicit notion of direction which derives from the fact that a sequence of instructions in a source code is executed as it is read, namely from the top to the bottom and from left to right. Graphs (and precubical sets) possess their own canonical notion of direction, on the contrary topological spaces do not. Directed topology was introduced to overcome this difficulty, however there is no consensus about a specific formalism. A part of my research work has consisted of comparing these formalisms, in particular I have thoroughly studied pospaces – Nachbin (1965), local pospaces – Fajstrup et al. (2006); Haucourt (2014), d-spaces – Grandis (2003, 2009), and streams – Krishnan (2009); Haucourt (2012). The notion of isothetic region has emerged from these investigations – Haucourt (2014).

Invariants of Directed Topology

After one has admitted that concurrency would favourably be modelled by directed topology, one may ask if there exist relevant invariants in the manner of fundamental groupoid for topological spaces – Higgins (1971); Brown (2006). The fundamental category construction has arisen from this question – Fajstrup et al. (2006). The path-connected components of a topological space are in canonical correspondence with the connected components of its fundamental groupoid. In fact the fundamental groupoid of a topological space is equivalent (as a category) to the disjoint union (*i.e.* coproduct in the category of small groupoids) of the fundamental groups – Hatcher (2002) of its path-connected components. In many concrete cases one can thus substitute a groupoid having uncountably many objects with a groupoid having finitely many ones. Is the same substitution possible when dealing with directed topology? The question has been investigated first in Fajstrup et al. (2004) and gave birth to categories of components. Nevertheless, at present time, this concept is fully understood only in the case of directed topology without directed loops – Haucourt (2006). Another important part of my research work has been dedicated to extending this concept to a broader framework: though it may seem anecdotal, this problem is actually related to the hypothetical notion of directed homotopy type. My work on the subject convinced me that model categories – Quillen (1967); Hovey (1999) do not fit with directed topology, though they naturally cross your mind when trying to define a notion of homotopy type over a category. A more promising approach is provided by Dwyer et al. (2004), see also Section 8.6.

Another problem I have been concerned with is the decomposition of processes – Milner and Moller (1993); Luttik (2003); Luttik and Oostrom (2005); Dreier et al. (2013). Balabonski and Haucourt (2010) indeed provides the decomposition of any PV program – Dijkstra (1968), as a parallel composition of subprograms that run independently of each others. This decomposition is moreover unique and obtained from the geometric model of the program. During his internship in our lab, Thibaut Balabonski proved a unique decomposition theorem for finite connected loop-free categories. They arise as categories of components of directed topologies without directed loops. Moreover the category of components construction preserves cartesian products. Then one asks whether the unique decomposition of a geometric model is sent to the unique decomposition of its category of components. This problem even has further connections with decomposition of metric spaces – Foertsch and Lytchak (2008). I have actively worked on the subject.

Practical Situation of Concurrency

Starting from control flow graphs and concurrent programming, we have arrived at the study of invariants of directed topology (*i.e.* algebaic directed topology). However, to be fully honest, we have to shed light on the limitations of this approach. We describe some explicit situations of concurrency to locate the range of applications of our methods.

– Distributed systems are made of a potentially large population of clones of a given process. The individuals of this population can be seen as ants performing a little bit of a huge task, their efforts being gathered on the fly. The "Map-Reduce" model of parallel computation, which derives from the celebrated Divide and Conquer strategy, is a typical example of this situation. Among its easiest applications there is the computation of the product of a huge collection of elements of a commutative monoid. Many real world systems are based on this idea. For example SETI@home performs spectral analysis on signals from space. The size of the population may vary depending on how much resources are available. In particular, life and death of a small group of processes does not matter as long as the product of the work already done is kept coherent.

– Operating Systems consist of a group of distinguished processes (windows and memory managers, task scheduler etc) that manage a heterogeneous population of processes which are spawned and killed according to the user requirements. The main problem here is to keep the "fundamental" processes active, the other ones being expandable with respect to the global stability of the system.

– Control command systems are made of a given group of processes, each of them being devoted to a specific activity, tightly communicating with each other so they efficiently drive some device. Every process of such systems tends to be important if not vital. It is then crucial to ensure that a command control system will not freeze (*i.e.* that it is has no deadlock). Also, it is often important to guarantee that its decision is deterministic (*i.e.* only depends on its current state instead of the execution trace that led to it). The other specific feature of this family of systems is that their available resources as well as their population of threads are both constant and known after a syntactic analysis of the source code.

Our methods mostly apply to the last kind of systems, more precisely to the asynchronous ones, as they strongly depend on the property that the population and the available resources be constant and declared in the header of the source code. They also require that all the processes of the system be conservative (*cf.* Definition 3.4.1).

Organization and Description of the Chapters

The Parallel Automata Meta Language (or just Paml in the sequel) is introduced in Chapter 1. It is based on the toy language introduced by Edsger Wybe Dijkstra (1968). Unlike its ancestor, it does not allow processes to be spawned, killed, or overwritten at runtime. Moreover the parameters of a synchronisation mechanism cannot be dynamically changed, and pointer arithmetic on shared variables is forbidden. Abstract machines are described and two notions of independence are introduced: the first one is based on a syntactical analysis of the program while the other one relies on the observation of the execution traces. The chapter finishes with the notion of control flow graph.

The relation between presheaves over a small category and their realization/nerve is discussed at the beginning of Chapter 2 in a very broad sense. The purpose is to introduce two instances of this setting, namely the cubical and the precubical sets, and also to explain why they better match the requirements of directed topology than the celebrated simplicial sets, which are the standard choice for combinatorial homotopy theory. Chapter 3 is devoted to the precubical models of Paml programs. They are built from tensor products of control flow graphs.

There is actually no consensus about the way directed topology should be formalized to fit with algebraic directed topology. Some approaches are studied in Chapter 4 from a categorical point of view. They are compared through the realization of (pre)cubical sets. For this purpose, we provide in Section 4.2 an abstract framework in which the constructions described in the subsequent chapters make sense.

The fundamental categories, which are the directed counterpart of fundamental groupoids, are the main concern of Chapter 5. A generic construction is given so it can be applied to all the categories described in Chapter 4. The resulting fundamental categories are compared.

Isothetic regions are introduced in Chapter 6. They provide a good compromise between genericity and tractability. In practice, they had been implemented in the ALCOOL software before they were formally defined since the continuous model of any Paml program (*cf.* Definition 7.1.2) naturally lies in their class. In theory, one checks that isothetic regions canonically embed in all the categories defined in Chapter 4, and several abstract constructions from algebraic directed topology can be automated in the context of isothetic regions. Hence they are put forward to provide the "right" framework for studying the kind of concurrency depicted in the introduction. From a combinatorial point of view, isothetic regions are related to labelled tensor products of 1-dimensional precubical sets (*i.e.* graphs), and as such, to special higher dimensional automata.

Categories of components are studied in Chapter 8. They are defined in a purely categorical context which makes the construction independent from directed topology and concurrency. Nevertheless, the intuition is that categories of components are to fundamental categories (in directed topology) as fundamental groups are to fundamental groupoids (in topology). From a technical point of view, the idea is to find classes of morphisms that contain the classes of isomorphisms and have similar properties.

In Chapter 9 we relate parallelization of code and unique decomposition results that hold for some of the invariants introduced in the preceding chapters.

Open problems and perspectives are discussed in the last chapter.

1

The Parallel Automata Meta Language

Classical Semantics

The programming language handled in Dijkstra (1968) is an extension of ALGOL60 providing the 'parallel compound' construction `parbegin ...parend`. Dijkstra specifies its semantics in the following terms: "Initiation of a parallel compound implies simultaneous initiation of all its constituent statements, its execution is completed after the completion of the execution of all its constituent statements." In other words the parallel compound construction is a synchronisation mechanism. On that basis, Dijkstra generalizes Dekker's mutual exclusion algorithm so that it works for more than two concurrently running processes. Starting from that, he explains how to implement general semaphores and introduces the primitives $P(\_)$ and $V(\_)$. Strictly speaking, Dijkstra's PV language refers to that ALGOL60 extension. It is not hard to see that any reasonable sequential language can be extended the same way. For example, Parallel Pascal (Cridlig (1995, 1997)) and the C programming language together with POSIX Threads can be seen as direct descendants of this idea. Nevertheless, all these languages are general purpose ones[1] while the methods that we will describe are in the first place dedicated to asynchronous control command systems. In this memoir, we therefore impose restrictions that are commonly met in the design of the latter. For example, the original PV language allows parallel compounds to occur anywhere in a program so they can be nested. On the contrary, asynchronous control command systems are often built on the following paradigm: a collection of sequential processes, known at compile time, is launched at the beginning of any execution of the program. Consequently, we only allow parallel compound in outermost position[2]. We also suppose that a pool of resources, also fixed at compile time, is shared by the running processes which might therefore get stuck because of a lack of available resources. In other words, the PV programs we consider have the form shown on Figure 1.1, where all the processes are sequential. In order to take these limitations into account, we only consider a fragment of the original Dijkstra's PV language by introducing the Parallel Automata Meta Language, or just Paml in the sequel. Since we aim at modelling programs more than

[1] The C language has been created to develop operating systems. The most widespread ones, namely Windows, UNIX, and Linux, are largely written in it.

[2] A similar assumption is made in (Miné, 2013, p.19).

```
resource declarations;
parbegin
   $process_1$;
   $process_2$;
   ...
   $process_N$;
parend
```

Figure 1.1: The general form of the PV programs to which the methods described in this memoir apply.

parsing them, Paml is not, strictly speaking, a restriction of the PV language. Instead, it is much closer to the intermediate language of a compiler like GCC or LLVM than to a human friendly one. The grammar of Paml is made explicit in Figure 1.2 following the Extended Backus-Naur Form[3]. The reason why we describe its syntax in detail is that Paml is the input language of the ALCOOL static analyzer. In particular, a complete description would require us to give, explicitly, a lexer and a parser (*cf.* (Aho et al., 2007, Chap. 3 and 4)). We will not do so and assume that we already have a "middle-end" representation at our disposal.

An informal description of the Paml syntax is given in Section 1.1. The middle-end representation of such programs is described in Section 1.2. They provide a structure that is easier to handle than raw source code. In particular, they are used to define a semantics for Paml in Sections 1.3 and 1.6. A naive approach to Paml programs parallel decomposition is presented in Section 1.5. The chapter ends with the description of an abstract machine for Paml (Section 1.6).

1.1 Syntax

The source code of a Paml program consists of a sequence of statements. Each of them is either the declaration of a resource, the description of a sequential process associated with an identifier, or a linear combination of identifiers with coefficients in $\mathbb{N}$. The latter is to be interpreted as the multiset of processes that are simultaneously launched at the beginning of any execution of the program.

Definition 1.1.1. Following the C programming language convention (Kernighan and Ritchie, 1988, p.25), **identifiers** are made up of letters and digits; the first character must be a letter, and the underscore character counts as a letter. A **constant** is a nonempty finite sequence of digits, possibly followed by a dot and another nonempty finite sequence of digits. In mathematical terms, constants are (decimal representations of) decimal numbers.

Definition 1.1.2. The **expressions** are finite trees whose nodes carry identifiers, constants, unary and binary operators taken from the following sets.

$$\{\,-\,,\,\neg\,\}\qquad\{\,\wedge\,,\,\vee\,,\,+\,,\,-\,,\,*\,,\,/\,,\,<\,,\,>\,,\,\leqslant\,,\,\geqslant\,,\,=\,,\,\%\,\}$$

The number of branches of a node being the arity of the operator it carries. The arity of any identifier or constant is null. Any expression e comes with the set $\mathcal{F}(e)$ of **free**

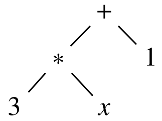
[3] The Extended Backus-Naur Form is described in the ISO/IEC 14977:1996(E) norm reference.

```
digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' ;
lowercase = 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't'
            | 'u' | 'v' | 'w' | 'x' | 'y' | 'z' ;
uppercase = 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R'
            | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z' ;
letter = lowercase | uppercase ;      spc = {' '}- ;
comma = ',' , [spc] ;      semicolon = ';' , [spc] ;      dot = '.' , [spc] ;      nat = {digit}- , [spc] ;
id = (letter | '_') , {letter | digit | '_'} , [spc] ;      id list = [nat] , id , {spc , [nat] , id}
int = ['-' | '+'] , nat ;      cst = ['-' | '+'] , { digit }- , ['.' , nat] ;
equal = '=' , [spc] ;      assign = ':=' , [spc] ;
left bracket = '(' , [spc] ;      right bracket = ')' , [spc] ;
unary op = ( 'not' | 'abs' ) , [spc] ;
binary op = ('+' | '-' | '*' | '/' | '&' | '|' | '<' | '<=' | '=' | '>=' | '>' | '<>') , [spc] ;
expr = cst | id | left bracket , expr , right bracket | unary op , expr | expr , binary op , expr ;
variable = ('variable:' | 'var:') , spc , id , equal , expr , {comma , id , equal , expr} ;
mutex = ('mutex:' | 'mtx:') , spc , id list ;
semsync = ('semaphore:' | 'sem:' | 'synchronization:' | 'sync:') , spc , [int , spc] , id list ;
jpvw = ('J' | 'P' | 'V' | 'W') ,  left bracket , id , right bracket ;
instruction = id , assign , expr | jpvw | sum | left bracket , [sum] , right bracket ;
sequence = instruction , {semicolon , instruction} ;
plus = '+' , [spc] ;      left square bracket = '[' , [spc] ;      right square bracket = ']' , [spc] ;
sum = sequence , {plus , [left square bracket , expr , right square bracket , plus] , sequence} ;
proc = ('process:' | 'proc:') , spc , id , equal , sequence , {comma , id , equal , sequence} ;
declaration = {variable | mutex | semsync | proc} ;
program = [spc] , declaration , ['init:', spc , id list] ;
```

Figure 1.2: The Paml grammar – following EBNF standard

variables of the e, viz all the identifiers appearing in it. The expressions are provided by the rule *expr* on Figure 1.2.

Example 1.1.3. The expression $3 * x + 1$.



Definition 1.1.4. According to rule *instruction* on Figure 1.2, an **instruction** is either

- an assignment <*identifier*>`:=`<*expression*>,
- a jump `J(`<*identifier*>`)`,
- a resource request `P(`<*identifier*>`)` / deallocation `V(`<*identifier*>`)`,
- a forced synchronisation `W(`<*identifier*>`)`,
- a conditional branching as below, where all the e_k (resp. i_k) are expressions (resp. instructions),

$$i_1 + [e_1] + \cdots + i_n + [e_n] + i_{n+1}$$

- the instruction that "does nothing", namely `Nop`.
- a sequence of instructions between grouping parentheses, and separated by semi-colons,

The keywords `P(_)` and `V(_)` stand for the Dutch terms *prolaag*, short for *probeer te verlagen* (*i.e.* "try to reduce"), and *verhogen* (*i.e.* "increase"). The keywords J and W stand for "jump" and "wait". The last kind of instructions is said to be **compound**. By opposition, the first ones are called **single instructions**. The set of free variables of a single instruction is defined below.

$$\mathcal{F}(x\colon =e) = \mathcal{F}(e)$$

$$\mathcal{F}(i_1 + [e_1] + \cdots + i_n + [e_n] + i_{n+1}) = \bigcup_{i=1}^{n} \mathcal{F}(e_i)$$

$$\mathcal{F}(\mathtt{P}(x)) = \mathcal{F}(\mathtt{V}(x)) = \mathcal{F}(\mathtt{W}(x)) = \mathcal{F}(\mathtt{J}(x)) = \emptyset$$

Definition 1.1.5. A **resource declaration** (*cf.* rule *declaration* on Figure 1.2) is one of the following statement:

- `sem:` <*int*> <*set of identifiers*> (*e.g.* `sem: 3 a b`, declares the **semaphores** a and b, both of them being of **arity** 3),
- `sync:` <*int*> <*set of identifiers*> (*e.g.* `sem: 7 a b c`, declares the **synchronisation barriers** a, b, and c, all of them being of **arity** 7), or
- `var:` <*identifier*> `=` <*constant*> (*e.g.* `var: x = 0`, declares variable `x` and set its initial value to 0).

A **process description** is a statement of the form

- `proc:` <*identifier*> = <*instruction*>. In most cases the instruction is compound (*e.g.* `proc: p = (P(a);V(a))` associates the identifier `p` with the compound instruction `(P(a);V(a))`). The right-hand part of the declaration is called the **body** of the process.

A **program bootup** is a statement of the form

- `init:` <*multiset of identifiers*> (*e.g.* `init: a 2b 3c`, indicates that one copy of a, two copies of b, and three copies of c are simultaneously launched at the beginning of each execution of the program).

A program is thus made of a set of sequential processes executing their instructions in parallel, and sharing a pool of resources. The arity of a semaphore a indicates the total number of tokens of type a. At the beginning of the execution of the program, all of them are in the common pool of resources. Simultaneous execution is constrained by limited resources which are taken and released by means of instructions $P(\_)$ and $V(\_)$. Given a semaphore a, the former (resp. the latter) applied to a should be read as "take a token of type a from the common pool of resources" (resp. "put it back to the pool"). A process which performs the instruction $P(a)$ is granted with a token of type a provided at least one is available in the pool, otherwise its execution is stopped until it is. In doing so, the process becomes the *owner* of a token. A process which performs the instruction $V(a)$ put a token of type a back in the common pool of resources, provided it owns some; otherwise the instruction is just ignored and the process execution keeps on going. It is worth noticing that Paml semantics differs from the PV language one as the latter does not take the concept of *owner* of a token into account. Also compare with the definition given in (Raynal, 2013, p.63). On the way, the notion of semaphore is quantitative while the notion of **mutex** (a short for "mutual exclusion") is qualitative. It means that the same process can hold several tokens of the same semaphore, but not several tokens of the same mutex. In practice, there is a single token for each mutex, if its owner tries to take it again, it is not stopped, the $P(\_)$ instruction being ignored. On the contrary, a process that tries to acquire a token of some unavailable semaphore is blocked regardless of the amount of tokens of that type it already owns. For the sake of simplicity, Paml mutices are just semaphores with a single token (*i.e.* of arity 1), hence they are quantitative. Adding genuine mutices (*i.e.* the qualitative ones) to Paml would not have raised any problem but a longer description of the semantics.

Parallel execution is also constrained by the 'wait' instruction $W(\_)$ which is intended to synchronise a given group of *existing* processes. The instruction $W(b)$, which should be read as "wait behind the *barrier b*", indeed stops any process that meets it until a certain number of processes get blocked by it. When the threshold, viz the arity of the barrier b, is crossed, the executions of all the processes stalled by the instruction resume. The 'Wait' instruction can be seen as a weakened form of parallel compound since the latter *creates* the processes that it synchronises while the former does not.

Remark 1.1.6. The loops are provided by the jump instruction. With the aid of parentheses and conditional branchings, an instruction can actually be a tree whose nodes carry conditional branchings. This is to be related to the standard notion of **extended basic blocks** which is used, for example, in both GCC and LLVM compilers – see also (Cooper and Torczon, 2011, p.418). In other words, the process declarations define tree-like extended basic blocks. The Paml version of the Hasse/Syracuse algorithm is given below, the PV language version is given on Figure 3.4. The question of whether this algorithm terminates for all initial values of `x` is still open.

```
var:  x = 7
proc:
   does_it_continue = J(continue)+[x<>1]+() ,
   continue = (x:=x/2 ; J(does_it_continue))+[x % 2 = 0]+
             (x:=3*x+1; J(does_it_continue))
init:  does_it_continue
```

Definition 1.1.7. A Paml **source code** is a finite sequence of resource declarations and process descriptions ended with a program bootup. The multiset given by the program bootup is called the **bootup multiset**.

Example 1.1.8 (Race Condition)**.** In concurrent programming, a **race condition** occurs when the execution of several instructions, that are meant to be performed simultaneously, results in an output that depends on the order in which the instructions have been executed. In the program below, two concurrent processes try to modify the content of the same variable. The assignments are enclosed between the instructions $P(a)$ and $V(a)$, which prevents a race condition to occur (*i.e.* that several processes access the variable simultaneously) because there is a single token of type a. OCaml threads are implemented that way: a global mutex ensures that two threads never access memory at the same time. This approach is certainly safe, but it is also the most inefficient one because it forbids parallel execution. JoCaml (*cf.* Fournet et al. (2003)) and ReactiveML (*cf.* Mandel and Pouzet (2005)) are OCaml extensions allowing efficient parallelism.

```
var:  x = 0
sem:  1 a
proc:
   p1 = P(a); x:=1; V(a) ,
   p2 = P(a); x:=2; V(a)
init:  p1 p2
```

In general, a part of code that must be executed by a single process at the time is called a **critical section** (*cf.* (Raynal, 2013, p.9)).

Example 1.1.9 (Producer *vs* Consumer)**.** In this program, a cyclic process called "producer" increments a variable at each turn. Another cyclic process called "consumer" decrements the same variable at each turn. Using a synchronisation barrier, we ensure that the variable content remains non-negative. In case of a race condition, which corresponds to the situation where the producer increments the variable while the consumer decrements it, we assume that the variable content is left unchanged. One easily check that the variable varies in $\{0, 1, 2\}$.

```
sync:  1 b
var:  amount = 0
proc:
   producer = amount := amount + 1 ; W(b) ; J(producer) ,
   consumer = W(b) ; amount := amount - 1 ; J(consumer)
init:  producer consumer
```

The race condition can even be avoided by considering the following program. In this case the variable alternatively takes the values 0 and 1.

```
sync:  1 b c
var:  amount = 0
proc:
   producer = amount := amount + 1 ; W(b) ; W(c) ; J(producer) ,
   consumer = W(b) ; amount := amount - 1 ; W(c) ; J(consumer)
init:  producer consumer
```

Example 1.1.10 (Deadlock)**.** Homer and Barney are at Moe's to have a drink. Unfortunately, all that's left is one bottle of beer, and its cap cannot be removed without the unique bottle cap opener available in the tavern. Homer rushes to the beer while Barney decides to get the bottle cap opener first. The situation is formalized by the Paml program below.

```
sem:  1 bottle opener
proc:
   Homer = P(bottle);P(opener);open bottle;V(opener);drink;V(bottle) ,
   Barney = P(opener);P(bottle);open bottle;V(opener);drink;V(bottle)
init:  Homer Barney
```

If Homer grab the bottle while Barney catch the opener then both of them got stuck and the beer is not drunk. Such a scenario appears in Dijkstra (1968) where two processes ask for additional memory space to the operating system, the problem being that none of the requests can be satisfied unless one of the two processes free the memory space it occupies. Dijkstra originally called such a situation a *deadly embrace* though it is nowadays better known as a *deadlock*. Hoare (1978) tells a similar story with "dining philosophers" and Chandy and Misra (1984) generalize it to "drinking philosophers".

Example 1.1.11 (Simultaneity)**.** Instructions $P(\_)$ and $V(\_)$ are used to prevent simultaneous executions from happening while instructions $W(\_)$ synchronise processes. Consequently, the left hand program below cannot avoid deadlock. On the contrary, no deadlock can occur during the execution of the right hand one.

```
sem:  1 a                         sem:  1 a
sync:  1 b                        sync:  1 b
proc:  p = P(a);W(b);V(a)         proc:  p = W(b);P(a);V(a)
init:  2p                         init:  2p
```

Remark 1.1.12. A Paml program is thus made of several sequential processes running in parallel, with the following restrictions:

- the number of running processes as well as the amount of available resources of the ambient system are known and fixed at compile time,
- there are no pointer arithmetics nor aliases. In particular the name of a variable should be thought of as an absolute address in the memory, therefore $x := y$ copies the content of the "memory cell" y in the "memory cell" x.

These restrictions arise from the fact that compilers/static analyzers need for generating the control flow structure of a program statically. We should keep in mind that the control flow structure is intended to be an overapproximation of the set of execution traces of the program. In order to be efficient, this structure has to take into account the interplay between processes due to synchronisation mechanisms. For example, suppose that a group of processes sharing some resource can nevertheless work concurrently without competing because the stock is large enough to supply them all. Now suppose that a process is created at runtime which also needs the common resource, then tension between processes could appear thus affecting the control flow structure. In fact any dynamical alteration of the global amount of resources may create or remove conflicts between processes. The ban on pointer arithmetics and aliases arises from another issue: we would like to know at compile time when two memory accesses refer to the same location. Under these assumptions, the answer becomes straightforward: different variables refer to different memory locations. Otherwise the question cannot be decided.

1.2 Middle-End Representation of Paml Programs

The middle-end representation of a program is a generic term that encompasses all the structures actually handled by compilers (resp. static analyzers) in the process of turning source codes (which are just texts) into binary files (resp. information about them). The control flow graphs of programs form an essential part of their middle-end representations but they are defined for sequential programs (*i.e.* Paml processes in our context) only. Still, they are the bricks from which we will make up the higher dimensional control flow structures of Paml programs. As the name suggests, the conditional branching instructions are represented by the branchings of the control flow graphs.

As we have already mentioned, dealing with Paml programs from their source codes requires to apply to the latter some routine compilation transformations that are summarized in Figure 1.3 (Aho et al., 2007, p.41). Roughly speaking, the lexical analyzer recognizes the keywords, the identifiers, and the constant appearing in the source code which is, strictly speaking, just a text. The parser then check that the text matches the grammar of the language and builds the so-called abstract syntax tree. On the way, it relates identifiers with syntactic constructions. In our case, the intermediate model generator builds the control flow grahs of the sequential processes. The symbol table is updated accordingly. At the output of the grayed part of Figure 1.3, which will be referred to as “the black box” in the sequel, we have the middle-end representation of the Paml program under consideration. The purpose of this section is to formalize the latter notion so we can get rid of the source code of a Paml program and identify it with its middle-end representation.

Definition 1.2.1. The black box readily extracts from the source code all the variables, the semaphores, and the barriers occurring in a program. They are respectively collected in the finite sets $\mathcal{V}$, $\mathcal{S}$, and $\mathcal{B}$.

Definition 1.2.2. According to the Paml grammar, each variable occurring in the source code is initialized when it is declared. The **initial valuation** is the mapping

$$\text{init} : \mathcal{V} \to \{\text{constants}\}$$

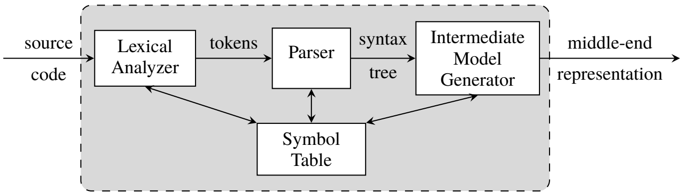


Figure 1.3: Producing the middle-end representation from the source code

which assigns a constant (*cf.* Definition 1.1.2) to every variable appearing in the source code. The set of expressions occurring in the program is denoted by $\mathcal{E}$.

Definition 1.2.3. Each semaphore or barrier comes with an element of $\mathbb{N} \cup \{\infty\}$ called its arity. Each Paml source code thus gives the mapping

$$\text{arity} : \mathcal{S} \sqcup \mathcal{B} \to \mathbb{N} \cup \{\infty\}$$

which associates each resource (semaphore or barrier) with its **arity**.

Definition 1.2.4. A **process identifier** (or just **pid** for short) is an ordered pair made of an identifier and a natural number. From the bootup (*cf.* Definition 1.1.5) appearing in a source code, the black box readily determines the set $\mathcal{P}$ of process identifiers of the (running processes of the) program. For example, if the bootup is the following line

```
init:  2p 3q
```

then $\mathcal{P}$ is the following set.

$$\big\{(p,0),(p,1),(q,0),(q,1),(q,2)\big\}$$

The black box also provides the mapping

$$\text{body} : \mathcal{P} \to \{\text{bodies of instructions}\}$$

which assigns a body of instructions (*cf.* Definition 1.1.4) to each process identifier. Given $p \in \mathcal{P}$ it is often useful to consider the set $\mathcal{P}(p)$ of processes involved in p, which is defined as the least subset of $\mathcal{P}$ containing p, and such that if the instruction $\mathtt{J}(q)$ appears in body(q') for some $q' \in \mathcal{P}(p)$, then $q \in \mathcal{P}(p)$.

The sets $\mathcal{V}$, $\mathcal{S}$, and $\mathcal{B}$ as well as the mappings init, arity, and body, are actually part of the symbol table of Figure 1.3. In fact, they are almost produced by the parser.

Before going further, we have to point out that a text that matches the grammar described on Figure 1.2 may not be a correct source code. For example using a variable (resp. calling a process) which has not been defined should be considered as an error. Correctness is partially checked by the black box so we assume that all the process and variable identifiers met in the source code of any program under consideration have been duly initialized.

Definition 1.2.5. The **middle-end representation** of a Paml program is the output of the black box, namely:

- the sets $\mathcal{V}$, $\mathcal{S}$, $\mathcal{B}$, and $\mathcal{E}$ (*cf.* Definitions 1.2.1 and 1.2.2)
- the mappings init : $\mathcal{V} \to \{\text{constants}\}$ (*cf.* Definition 1.2.2) and arity : $\mathcal{S} \sqcup \mathcal{B} \to \mathbb{N} \cup \{\infty\}$ (*cf.* Definition 1.2.3), and
- the mapping body : $\mathcal{P} \to \{\text{bodies of instructions}\}$ (*cf.* Definition 1.2.4).

1.3 Interpreting Multi-Instructions

From a theoretical point of view, an abstract machine is a map $[\![\_]\!]$ which assigns a (possibly infinite) sequence $[\![P]\!]$ of elements of a set Σ to (the middle-end representation of) every Paml program P. The elements of Σ are the (internal) states of the machine. The map $[\![\_]\!]$ is defined inductively according to the structure of Paml programs, starting from its most simple elements, namely its expressions (*cf.* Definition 1.1.2).

Definition 1.3.1. An **abstract valuation** is a mapping from $\mathcal{V}$ to $\mathbb{R}$.

Recall that $\mathcal{V}$ is the set of variables appearing in the program under consideration. Hence, a valuation should be thought of as a "memory state". In particular the variables take their **values** in $\mathbb{R}$ (due to the restricted set of operators we could have worked with $\mathbb{Q}$).

Definition 1.3.2. An **abstract expression** is a partial mapping

$$\varepsilon : \{\text{abstract valuations}\} \longrightarrow \mathbb{R}$$

together with a subset $\mathcal{F}(\varepsilon) \subseteq \mathcal{V}$ such that if the valuations v and v' match on $\mathcal{F}(\varepsilon)$ then $\varepsilon(v) = \varepsilon(v')$. The **free variables** of ε are the elements of $\mathcal{F}(\varepsilon)$.

We produce an abstract expression (*cf.* Definition 1.3.2) from a "concrete" one (*cf.* Definition 1.1.2).

Definition 1.3.3. The **evaluation** $[\![\_]\!]$ of an expression of the Paml language is an abstract expression defined inductively for all abstract valuations v:

- $[\![x]\!] = v(x)$ for all $x \in \mathcal{V}$, undefined if x has not been declared,
- $[\![c]\!]$ is the element of $\mathbb{R}$ whose decimal representation is the constant c,
- each of the operator is evaluated as an operator in $\mathbb{R}$:
  - $[\![-]\!]$ returns the opposite of its argument,
  - $[\![\neg]\!]$ returns 0 if its argument is nonzero, and 1 otherwise,
  - $[\![\%]\!]$ is the modulo operator,
  - $[\![+]\!]$ and $[\![-]\!]$ are the addition and the substraction operators,
  - $[\![*]\!]$ and $[\![/]\!]$ are the multiplication and the division operators,
  - $[\![\wedge]\!]$ and $[\![\vee]\!]$ are the minimum and the maximum operators,
  - the evaluations of the Boolean operators return 1 for "true" and 0 for "false",
- for all unary operators *op* and all expressions e, $[\![op\ e]\!] = [\![op]\!]([\![e]\!])$
- for all binary operators *op* and all expressions e_1, e_2, $[\![e_1\ op\ e_2]\!] = [\![op]\!]([\![e_1]\!], [\![e_2]\!])$

Remark 1.3.4. The operators $[\![\%]\!]$ and $[\![/]\!]$ are the only ones that are not defined on the entire domain $\mathbb{R}^2$ since division by zero is not allowed.

Remark 1.3.5. In most cases, the evaluation of an expression depends on v so we write $[\![\_]\!]_v$ to emphasize this fact when it is necessary. Nevertheless, if no variable occurs in the expression e, then $[\![e]\!]_v$ does not depend on v. In particular, if init : $\mathcal{V} \longrightarrow \{\text{constants}\}$ is the initial valuation of a Paml program (*cf.* Definition 1.2.2), then we define the corresponding abstract valuation as below.

$$[\![\text{init}]\!] : v \in \mathcal{V} \mapsto [\![\text{init}(v)]\!] \in \mathbb{R}$$

In order to interpret the instructions of the language, we need a partial description of the internal states of the abstract machines.

Definition 1.3.6. A **distribution of resources** is map sending each element of $\mathcal{S}$ to a multiset over $\mathcal{P}$. Given such a mapping σ and $s \in \mathcal{S}$, the multiset $\sigma(s)$ should be understood as the mapping that binds each process p with the number of tokens of type s that it owns. In particular for all $s \in \mathcal{S}$ the mass

$$|\sigma(s)| \quad = \quad \sum_{p \in \mathcal{P}} \big(\sigma(s)\big)(p)$$

is the total amount of tokens of type s held by (all the running processes of) the program. Consequently, Paml takes the notion of owner into account. On the contrary, the original PV language does not.

Definition 1.3.7. A **context of interpretation** is a map defined over $\mathcal{V} \sqcup \mathcal{S}$ whose restriction to $\mathcal{V}$ (resp. $\mathcal{S}$) is an abstract valuation (resp. a distribution of resources). Given the middle-end representation of a Paml program, the interpretation of the initial valuation $[\![\text{init}]\!]$ (*cf.* Remark 1.3.5) and the map which sends every $s \in \mathcal{S}$ to the empty multiset over $\mathcal{P}$, form the associated **initial context**.

A language enabling concurrent execution has to offer the programmer a way to tell the compiler/static analyzer which part of the code should/should not be executed in parallel. There are two opposite approaches: "parallelize unless otherwise stated" vs "parallelize only if explicitly stated". The former paradigm is the most commonly met (*e.g.* POSIX threads), so it is the one we have chosen to base the Paml language on (the restriction on parallel execution arising from instructions $P(\_)$, $V(\_)$, and $W(\_)$). This choice is reflected by the convention made in the subsequent definitions.

Since Paml is a parallel language, there are many cases where several instructions should be allowed (not forced) to be performed at the same time. Indeed the Paml language was designed to enable simultaneous execution, so its abstract machine should be adapted accordingly. Yet, it is clear from a quick examination of Example 1.6.3 that we must put a limitation on what can be done simultaneously. In this spirit, we define and interpret multi-instructions (*cf.* Definition 1.3.8), which should be compared to the *multisets of actions* introduced by Cattani and Sassone (1996).

Definition 1.3.8. A **multi-instruction** is a partial map μ from $\mathcal{P}$ to the set of single instructions. It is said to be **trivial** when, as a (partial) map, it always returns `Nop`. Two multi-instructions are said to be **disjoint** when so are their domains of definition.

Following many parallel architectures so far the memory is shared. That assumption is reflected here by the fact that all the variables can be seen by all the running processes. Hence we have to deal with the problem of race conditions (*cf.* Example 1.1.8). We could choose any arbitrary rule of precedence, without specifying it, so the result of any parallel execution leading to a race condition would not be specified. We forbid such executions instead.

Definition 1.3.9. Two instructions **conflict** when both modify the same variable (one says that it is a **write-write conflict**) or one of them alters a free variable of the other (one says that it is a **read-write conflict**).

Remark 1.3.10. In the first case, both instruction are assignments $x{:=}e$ and $x{:=}e'$ regardless of e and e' being identical. In the second case, one of the instructions is an assignment $x := e$ while the other is a conditional branching or an assignment in which x occurs as a free variable.

Remark 1.3.11. Definition 1.3.9 proposes a simple criterion allowing parallel execution of assignments. Nevertheless this approach fails if pointer arithmetics and index arrays are allowed since in that case, one is no longer able to decide whether assignments read/write the same part of the memory. It is the reason why we refrain from providing the Paml language with such features.

Definition 1.3.12. A multi-instruction μ defined over $M \subseteq \mathcal{P}$ is said to be **admissible** in the context σ when:

- for $i, j \in M$ with $i \neq j$, $\mu(i)$ and $\mu(j)$ do not conflict,
- for all semaphores $s \in \mathcal{S}$, $|\sigma(s)| + \mathrm{card}\{i \in M \mid \mu(i) = \mathtt{P(}s\mathtt{)}\} \leqslant \mathrm{arity}(s)$, and
- for all synchronising barriers $b \in \mathcal{B}$, $\mathrm{card}\{i \in M \mid \mu(i) = \mathtt{W(}b\mathtt{)}\}$ is either null or greater than arity(b) in the strict sense.

The empty multi-instruction is admissible. Any multi-instruction μ defined on a singleton $\{i\}$ is admissible unless $\mu(i) = \mathtt{W(}b\mathtt{)}$ with $\mathrm{arity}(b) \geqslant 1$. The arity of a semaphore s is the number of available occurrences of s, while the arity of a barrier is the maximal number of processes that it can stop. Therefore, one should always have $|s|_\sigma \leqslant \mathrm{arity}(s)$, and a barrier of arity n can be gone through iff at least $n + 1$ processes "push" on it. These constraints are reflected by the last two conditions. In particular any process trying to acquire an occurrence of a semaphore of null arity is definitely blocked. The same happens when a process comes across a synchronisation barrier whose arity is infinite. On the contrary, semaphores of infinite arity and barriers of null arity are harmless: the latter are roughly speaking ignored and the former cannot bring about the process stalling. It is worth noticing that our semaphore semantics slightly differs from the one found in Dijkstra (1968) which does not take the concept of "owner" of a semaphore occurrence into account. Moreover the arity of a semaphore s in the sense of the article (Dijkstra (1968)) is $\mathrm{arity}(s) + 1$.

As an immediate by-product of Definition 1.3.12, we define a partial action of the set of multi-instructions on the right side of the set of contexts.

Definition 1.3.13. Assuming that the multi-instruction μ is admissible in the context σ, the context $\sigma \cdot \mu$ is defined as follows.

- For all $x \in \mathcal{V}$ there is at most one $i \in \{1, \ldots, n\}$ such that the instruction $\mu(i)$ modifies x. In that case the instruction is an assignment $x\mathtt{:=}e$ and the value

  $$\sigma \cdot \mu(x) \quad = \quad [\![e]\!](\sigma|_\mathcal{V})$$

  does not depend on the instructions $\mu(j)$ for $j \in \{1, \ldots, n\} \setminus \{i\}$ since they do not alter the free variables of e. Nevertheless $[\![e]\!](\sigma|_\mathcal{V})$ may not be well-defined (*e.g.* division by zero). In that case we say that the multi-instruction μ "crashes" in the context σ. If no component of the multi-instruction μ alters the variable x, then $\sigma \cdot \mu(x) = \sigma(x)$.
- For all semaphores $s \in \mathcal{S}$, the multiset $\sigma \cdot \mu(s)$ is defined as follows.

  $$\sigma(s) + \big\{p \in M \,\big|\, \mu(p) = \mathtt{P(}s\mathtt{)}\big\} - \big\{p \in M \,\big|\, \mu(p) = \mathtt{V(}s\mathtt{)}\big\}$$

We are considering the arithmetic of the multisets hence negative values are not allowed. In particular the multiset $\sigma \cdot \mu(s)$ is the mapping from $\mathcal{P}$ to $\mathbb{N}$ defined below.

$$p \quad \mapsto \quad \begin{cases} \big(\sigma(s)\big)(p) + 1 & \text{if } p \in M \text{ and } \mu(p) = \mathtt{P}(s) \\ \max\{0, \big(\sigma(s)\big)(p) - 1\} & \text{if } p \in M \text{ and } \mu(p) = \mathtt{V}(s) \\ \big(\sigma(s)\big)(p) & \text{in all other cases} \end{cases}$$

The **interpretation** $[\![\mu]\!]$ of the multi-instruction μ is thus a partial mapping from the set of contexts to itself defined by $[\![\mu]\!](\sigma) = \sigma \cdot \mu$. Its domain of definition is the collection of contexts in which μ is admissible.

Definition 1.3.14. A finite family of multi-instructions is said to be **summable** in the context σ when its members are pairwise disjoint and its union is still admissible in the context σ.

Remark 1.3.15. Given a summable family $\{\mu_1, \ldots, \mu_n\}$ in the context σ, and a permutation π of $\{1, \ldots, n\}$ the contexts $\sigma \cdot \mu_1 \cdots \mu_k$ and $\sigma \cdot \mu_{\pi(1)} \cdots \mu_{\pi(k)}$ are well-defined for all $k \in \{1, \ldots, n\}$. Moreover, if $k = n$, they are equal. Nevertheless, due to the barriers of synchronisation, a restriction μ' of a multi-instruction μ may not be admissible in the context σ even if μ is.

1.4 Small Step Semantics

We describe an abstract machine which provides Paml with a structural operational semantics (Plotkin (2004)). The first step consists of defining its internal states so that the control flow can be managed. We suppose that we have the middle-end representation of the program to execute (*cf.* Definition 1.2.5). Given a nonempty sequence $(x_1, x_2, \ldots)$ we introduce the following notation.

$$\text{head}(x_1, x_2, \ldots) = x_1 \qquad \qquad \text{tail}(x_1, x_2, \ldots) = (x_2, \ldots)$$

Definition 1.4.1. Given a finite sequence of instructions $(i_1, \ldots, i_n)$, the finite sequence of instructions $\text{unbox}(i_1, \ldots, i_n)$ is defined recursively as:

- $(i_1, \ldots, i_n)$ if i_1 is a single instruction, and
- $\text{unbox}(j_1, \ldots, j_m, i_2, \ldots, i_n)$ if i_1 is the compound instruction $\mathtt{(}j_1\mathtt{;}\ldots\mathtt{;}j_m\mathtt{)}$

A finite sequence of instructions which is left invariant by the unbox operator is said to be **unboxed**.

Definition 1.4.2. An **internal state** of the abstract machine is a map σ defined over the set $\mathcal{V} \sqcup \mathcal{S} \sqcup \mathcal{P}$ such that the restriction of σ to $\mathcal{V} \sqcup \mathcal{S}$ is a context of interpretation (*cf.* Definition 1.3.7) and $\sigma(p)$ is an unboxed finite sequence of instructions for all $p \in \mathcal{P}$. Such a sequence is called a **stack of instructions**. The initial context of the program (*cf.* Definition 1.3.7) together with the map (*cf.* Definition 1.2.5)

$$p \in \mathcal{P} \mapsto \text{unbox}(\text{body}(p))$$

form the **initial state** of the machine.

We extend the partial action of multi-instructions on contexts (*cf.* Definition 1.3.13) to an action on internal states.

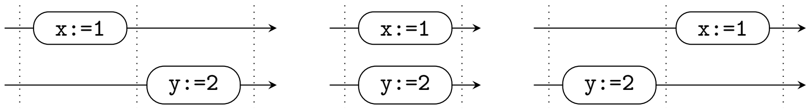


Figure 1.4: Execution traces as time lines

Definition 1.4.3. Let μ be a multi-instruction that is admissible in the context of the internal state σ. Applying Definition 1.3.13, the context of $\sigma \cdot \mu$ is the following one.

$$(\sigma|_{\mathcal{V}\sqcup\mathcal{S}}) \cdot \mu$$

Then given $p \in \mathcal{P}$, the stack of instructions $(\sigma \cdot \mu)(p)$ is:

- $\sigma(p)$ if $p \notin \operatorname{dom}\mu$,
- unbox(body(q)) if $\mu(p) = \mathtt{J}(q)$,
- unbox(i_k) · tail($\sigma(p)$) if $\mu(p) = i_1 + [e_1] + \cdots + i_n + [e_n] + i_{n+1}$ and

  $$k = \min\left(n+1\,,\ \inf\left\{\,k \;\middle|\; [\![e_k]\!]_\sigma \neq 0\,\right\}\right)$$

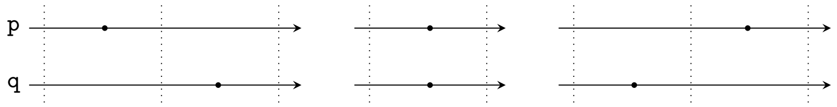
  with the usual convention that $\inf \emptyset = \infty$, and
- unbox(tail($\sigma(p)$)) in all other cases.

The **interpretation** $[\![\mu]\!]$ of the multi-instruction μ given in Definition 1.3.13 thus extends to a partial mapping from the set of internal states to itself. It is defined by $[\![\mu]\!](\sigma) = \sigma \cdot \mu$. By definition, its domain of definition is the collection of contexts in which μ is admissible. It provides Paml with an operational semantics allowing several processes to execute their current instruction simultaneously.

Remark 1.4.4. From an internal state σ and a subset M of $\mathcal{P}$ such that for all $p \in M$, the sequence $\sigma(p)$ is nonempty, one defines a multi-instruction μ over M by $\mu(p) = \operatorname{head}(\sigma(p))$. We insist that if $\sigma(p)$ is empty for some $p \in M$, then μ is undefined. Extending Definition 1.3.12, the set M is said to be admissible in the state σ when μ is admissible in the context $\sigma|_{\mathcal{V}\sqcup\mathcal{S}}$. In that case we define the action of M of the right of σ, denoted by $\sigma \cdot M$, as the action of μ of the right of σ (*i.e.* as the internal state $\sigma \cdot \mu$).

Definition 1.4.5. An **execution trace** starting at the internal state σ (*cf.* Definition 1.4.2) is a (possibly infinite) sequence $M_0 \ldots M_n \ldots$ of nonempty subsets of $\mathcal{P}$ such that for all n, if M_n is defined then the associated multi-instruction is well-defined and admissible in the internal state $\sigma \cdot M_0 \cdots M_{n-1}$ (*cf.* Remark 1.4.4). An execution trace is said to be **maximal** when it is infinite or when there exists $n \in \mathbb{N}$ such that every nonempty set that is admissible in the state $\sigma_0 \cdot M_0 \cdots M_n$ induces a multi-instruction that crashes (*cf.* Definition 1.3.13). An execution trace is said to be **interleaving** when every multi-instruction it contains is either a single instruction (*i.e.* M is a singleton) or has a range reduced to the singleton $\{\mathtt{W}(b)\}$ for some barrier b. An execution trace of a Paml program P is an execution trace starting at the initial state of P.

There are three reasons why an execution trace cannot be extended.



```
var:  x = 0, y = 0
proc:  p = x:=1
proc:  q = y:=2
init:  p q
```

Figure 1.5: Execution traces as time lines (abstract form)

Definition 1.4.6. Let σ be the output internal state of a finite maximal execution trace of a program P:

- if all the stacks $\sigma(p)$ are empty, then we say that the execution properly finishes,
- if some stack $\sigma(p)$ is not empty and the only admissible set is the empty one, then we say that the execution ends in a **deadlock**,
- in all other cases, the execution **crashes**.

The notion of deadlock is illustrated by Example 1.1.10. We will look again at the notion of deadlock in a more abstract context (*cf.* Definition 3.4.10). The last case occurs when the only available instructions lead to division by zero.

It is often useful to visualize execution traces as parallel time lines along which single instructions are pinned. Vertically aligned instructions should therefore be understood as the components of a multi-instruction. An illustration is given on Figure 1.4 with two running processes (*i.e.* the set $\mathcal{P}$ has two elements). Sometimes we are only interested in the domain of definition of the multi-instructions, in that case we simplify the visual representations by just pinning dots on time lines instead of instructions, see Figure 1.5.

1.5 Independence of Programs

This section provides syntactic criteria ensuring that a parallel composition of Paml programs make sense and introduces two standard notions of independence of programs.

Definition 1.5.1. A (finite) family of Paml programs is said to make **coherent** declarations when it satisfies the following properties:

- any variable identifier declared in two programs of the family is initialized with the same value (*i.e.* the initial valuations of any two programs of the family are compatible (*cf.* Definition 1.2.2))
- any semaphore (resp. synchronisation barrier) identifier declared in two programs of the family have the same arity (*i.e.* the arity maps of any two programs of the family are compatible (*cf.* Definition 1.2.3)), and
- any process identifier appearing in two programs of the family is associated with the same body of instructions (*i.e.* the body maps of any two programs of the family are compatible (*cf.* Definition 1.2.4)).

Definition 1.5.2. Given a finite sequence of Paml programs $P_1, \ldots, P_N$ that make coherent declarations, it makes sense to define their **parallel composition** $P_1|\cdots|P_N$ as the following Paml program:

- any variable, semaphore, or synchronisation barrier declared in one of the programs is declared the same way in the parallel composition
- any identifier associated with a body of instruction in one of the programs is associated with the same body in the parallel composition.
- the bootup multiset of the parallel composition is the sum of all the bootup multisets of the programs.

In that case, the middle-end representation of the parallel composition $P_1|\cdots|P_N$ is deduced from the middle-end representations of the programs P_i in the obvious way.

Definition 1.5.3. A family of Paml programs is said to be **syntactically independent** when any variable, semaphore, or synchronisation barrier appearing in one of them does not appear in the others.

Remark 1.5.4. The intuition behind Definition 1.5.3 is rather clear: since the sets of resources required by the programs are pairwise disjoint, they should be able to run concurrently, the ones independently from the others. This idea will be formalized in two other ways in Definitions 1.5.7 and 7.3.1. In practice, one is interested in **separating** Paml programs, that is to say obtain them as parallel compound of "independent" Paml programs. For the moment, it is done by syntactic means only. We will provide a semantic approach in Section 9.3 which will turn out to be more accurate.

Example 1.5.5. The following Paml programs are syntactically independent.

```
sem:  2 a                            sem:  2 x
var:  x = 0                          var:  a = 0
proc:  p = P(a);x:=1;V(a)            proc:  q = P(x);a:=1;V(x)
init:  2p                            init:  2q
```

Remark 1.5.6. Renaming is a standard manipulation on source codes which, formally speaking, consists of applying to a source code a permutation of the set of all existing identifiers in the obvious way. When a family of Paml source codes is under consideration for parallel composition, applying the same permutation to all the members of the family neither alters the coherence of its declarations nor its syntactic independence. Because the resources are shared, a variable (resp. semaphore, barrier) declaration appearing in one the programs of the family is global in the whole family. On the contrary, a process declaration is local to the program in which it appears. As an illustration, the programs below are syntactically independent though they do not make coherent declarations due to a process naming conflict. But they clearly do so if, for example, one changes `p` into `q` in the right-hand program.

```
sem:  1 a                            sem:  1 b
proc:  p = P(a);V(a)                 proc:  p = P(b);V(b)
init:  p                             init:  p
```

More generally, one can soundly rename the processes in each member of a family of programs so that no conflicting process declaration occurs. Indeed, that renaming does not change the meaning of the parallel composition. In particular, any Paml program

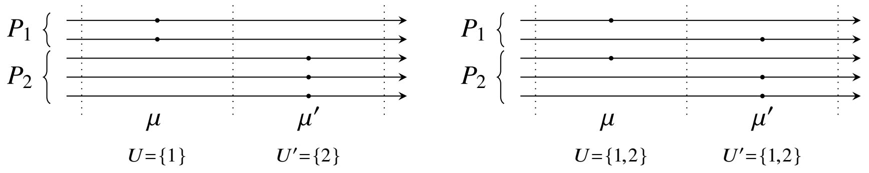


Figure 1.6: Disjoint vs not disjoint multi-instructions

P with n running processes can be written as a parallel compound of n Paml programs with a single running process: it suffices to declare each running process individually. An example is given below.

```
sem:  1 a
proc:  p = P(a);V(a)
init:  2p
```

```
sem:  1 a
proc:  p = P(a);V(a)
proc:  q = P(a);V(a)
init:  p q
```

Consequently, Definitions 1.5.1 and 1.5.2 could be weakened by dropping the constraints on process declarations. The only drawback in using the weak form of Definitions 1.5.1 is that the middle-end representation of the parallel compound of a family of programs is no longer canonically obtained from the middle-end representations of the members of the family. Indeed, one may need to apply some unspecified process renaming to make these representations compatible.

The notion of execution trace (*cf.* Definition 1.4.5) allows us to introduce another notion of independence. Suppose that the programs $P_1, \ldots, P_N$ make coherent declarations (*cf.* Definition 1.5.1) and that $\mathcal{P}_n$ is the set of (process identifiers of the) running processes of P_n for all $n \in \{1, \ldots, N\}$. The set $\mathcal{P}$ of (process identifiers of the) running processes of the parallel composition $P_1|\cdots|P_N$ is the disjoint union $\mathcal{P}_1 \sqcup \cdots \sqcup \mathcal{P}_N$. Extending Definition 1.3.8, we say that the multi-instructions μ and μ' are **disjoint** with respect to the parallel composition $P_1|\cdots|P_N$ when the sets of programs they trigger are disjoint. Formally, if U (resp. U') is the set of indices $n \in \{1, \ldots, N\}$, such that $\operatorname{dom}\mu \cap \mathcal{P}_n \neq \emptyset$ (resp. $\operatorname{dom}\mu' \cap \mathcal{P}_n \neq \emptyset$), then $U \cap U' = \emptyset$. On Figure 1.6, we provide an example of disjoint multi-instructions (on the left-hand side) and an example of not disjoint ones (on the right-hand side). Given $K \in \mathbb{N}$, a permutation π of $\{0, \ldots, K\}$ is said to be **compatible** with the parallel compound $P_1|\ldots|P_N$ and the finite sequence of multi-instructions $\mu_0, \ldots, \mu_K$ when it is order preserving for all $k, k' \in \{0, \ldots, K\}$ such that μ_k and $\mu_{k'}$ are not disjoint with respect to $P_1|\ldots|P_N$.

$$k \leqslant k' \quad \wedge \quad \mu_k \text{ and } \mu_{k'} \text{ not disjoint with respect to } P_1|\ldots|P_N \quad \Rightarrow \quad \pi(k) \leqslant \pi(k')$$

For the sake of readability, we often omit the dependency to $P_1|\ldots|P_N$. Recall that the symmetric groups $\mathfrak{S}_{K+1}$ acts on the left of the sequences $\overline{\mu}$ of length $K+1$ as follows:

$$\pi \cdot \overline{\mu} \quad = \quad \overline{\mu} \circ \pi^{-1} \quad = \quad \mu_{\pi^{-1}(0)} \cdots \mu_{\pi^{-1}(K)}$$

where $\overline{\mu}$ is interpreted as a mapping defined over $\{0, \ldots, K\}$.

Definition 1.5.7. Denoting by σ the initial state of the parallel compound $P_1|\cdots|P_N$, the programs $P_1, \ldots, P_N$ are said to be **observationally independent** when for all

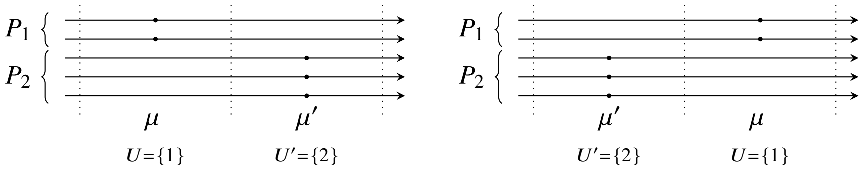


Figure 1.7: Compatible transposition

internal states σ' such that $\sigma'|_{\mathcal{P}} = \sigma|_{\mathcal{P}}$, for all execution traces $\overline{\mu} = \mu_0, \ldots, \mu_K$ starting at σ', and for all permutations π compatible with it, the sequence $\pi \cdot \overline{\mu}$ is still an execution trace starting at σ' and has the same action as $\overline{\mu}$ on σ'.

$$\sigma' \cdot \mu_0 \cdots \mu_K \quad = \quad \sigma' \cdot \mu_{\pi^{-1}(0)} \cdots \mu_{\pi^{-1}(K)}$$

In less formal words, a family of programs is observationally independent when up to the initial values of the variables, the output of any execution trace of the parallel compound does not depend on the order in which disjoint multi-instructions were performed. The permutation shown on Figure 1.7 is compatible. Hence, if P_1 and P_2 are observationally independent, the actions on internal states σ' of the two execution traces on Figure 1.7 are the same.

Remark 1.5.8. The notion of independence given in Definition 1.5.7 is related to the notion of deterministic Paml programs from Remark 1.6.2. It is weak in the sense that n copies of a program can be observationally independent. For example, consider the program whose unique process has a unique instruction, namely the assignment `x:=1`. A family of n copies of this program is observationally independent, but not syntactically.

```
var:  x = 0                         var:  x = 0
proc:  p = (x := 1)                 proc:  p = (x := 1)
init:  p                            init:  np
```

Replacing the assignment `x:=1` by a request `P(`s`)` (resp. a synchronisation `W(`b`)`) also provides an example of program that is observationally independent from itself, regardless of the arity of the semaphore s (resp. the barrier b).

Remark 1.5.9. Syntactically independent programs are observationally independent but the converse is false as shown by Remark 1.5.8.

Syntactical independence is decidable yet too restrictive. On the contrary observational independence is purely theoretic because it cannot be stated until all the execution traces have been tested or a mathematical proof has been given. We will introduce a third notion of independence (*cf.* Definition 7.3.1) which is strictly weaker that the syntactic one yet still decidable. We complete this section with a technical result that will be used in the proof of Theorem 7.3.4. We denote the set of fixpoints of a map f by fix (f).

Definition 1.5.10. Let us define a **rolling** as a cyclic permutation ρ of the form

$$(\, x \quad x+1 \quad \cdots \quad x+y \,)$$

for some $x, y \in \mathbb{N}$, that is to say defined by $\rho(x+t) = x + (t+1 \mod y+1)$. A rolling is canonically written as a composite of elementary transpositions.

$$(\, x \quad x+1 \quad \cdots \quad x+y \,) \quad = \quad (x \quad x+1)\,(x+1 \quad x+2) \cdots (x+y-1 \quad x+y)$$

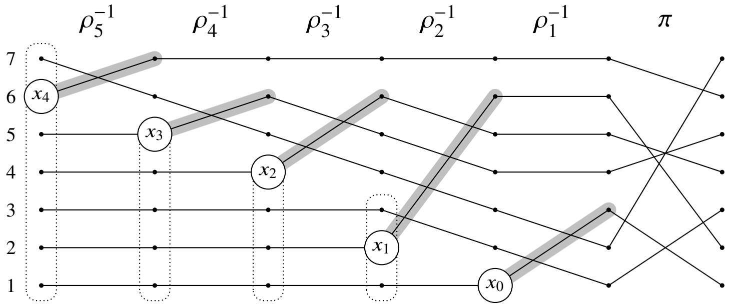


Figure 1.8: Rolling decomposition of a permutation

Since elementary transpositions (*i.e.* permutations of the form $(x\ x+1)$) generate the symmetric groups, the rollings generate them too. Nevertheless, among all the ways of writing a given permutation π as a composite of rollings, one can be considered as "canonical". More precisely one has a finite sequence of rollings $\rho_1, \dots, \rho_D$ with $D \in \mathbb{N}$ such that, denoting $\pi_d = \pi \circ \rho_1^{-1} \circ \cdots \circ \rho_d^{-1}$ for all $d \in \{0, \dots, D\}$, we get $\pi_D = \mathrm{id}$ (*i.e.* $\pi = \rho_D \circ \cdots \circ \rho_1$) and the greatest initial segment of $\mathrm{fix}\,(\pi_d)$ is strictly contained in the greatest initial segment of $\mathrm{fix}\,(\pi_{d+1})$. If π_d is not an identity and x_d is the least element of its support, then ρ_{d+1} is the rolling below.

$$\rho_{d+1} \quad = \quad (x_d \quad x_d+1 \quad \cdots \quad \pi_d^{-1}(x_d))$$

The induction obviously stops and the sequence $\rho_1, \dots, \rho_D$ is called the **rolling decomposition** of $\pi = \rho_D \circ \cdots \circ \rho_1$. The construction is illustrated on Figure 1.8. It plays a special role as it preserves compatibility.

Remark 1.5.11. If $\rho_1, \dots, \rho_D$ is the rolling decomposition of π, then $\rho_d, \dots, \rho_D$ is the rolling decomposition of π_{d-1}.

Lemma 1.5.12. If $\rho_1, \dots, \rho_D$ is a rolling decomposition and $\rho_D \circ \cdots \circ \rho_1$ is compatible with the sequence of multi-instructions $\overline{\mu}$, then ρ_1 is compatible with $\overline{\mu}$ and $\pi_1 = \rho_D \circ \cdots \circ \rho_2$ is compatible with $\rho_1 \cdot \overline{\mu}$.

Proof. Let x be the least element of the support of π. Given $0 \leqslant k < k' \leqslant K$, one has $\rho_1(k) > \rho_1(k')$ iff $k' = \pi^{-1}(x)$ and $x \leqslant k < \pi^{-1}(x)$. Then $\pi(k') = x < \pi(k)$ by minimality of x. Because π is compatible with $\overline{\mu}$, the multi-instructions μ_k and $\mu_{k'}$ are disjoint.

Given $0 \leqslant k < k' \leqslant K$ such that $\pi_1(k) > \pi_1(k')$, we would like to prove that $\mu_{\rho_1^{-1}(k)}$ and $\mu_{\rho_1^{-1}(k')}$ are disjoint. Note that $\pi_1 = \pi \circ \rho_1^{-1} = \rho_D \circ \cdots \circ \rho_2$ and apply the following change of variables.

$$k = \rho_1(k'') \qquad\qquad k' = \rho_1(k''')$$

We need to prove that the multi-instructions $\mu_{k''}$ and $\mu_{k'''}$ are disjoint under the hypothesis that $\pi(k'') > \pi(k''')$. Because π is compatible with $\overline{\mu}$ it suffices to check that $k'' < k'''$. Assume that it is not the case (*i.e.* $\rho_1^{-1}(k) > \rho_1^{-1}(k')$ since k'' and k''' cannot be equal). Then by definition of ρ_1^{-1} we have $k = x$ and $x + 1 \leqslant k' \leqslant \pi(x)$. As a consequence we get

$$\pi_1(k) \quad = \quad \pi_1(x) \quad = \quad \pi(\rho_1^{-1}(x)) \quad = \quad \pi(\pi^{-1}(x)) \quad = \quad x$$

because $\pi(x) = \rho_1(x)$ by definition of ρ_1. For the same reason we have $\pi_1(k') = \pi(k'-1)$. Because $k'-1 < \pi^{-1}(x)$ and by minimality of x, we deduce the equalities

$$\pi_1(k') \quad = \quad \pi(k'-1) \quad > \quad x \quad = \quad \pi_1(k)$$

which is a contradiction. □

1.6 Abstract Machine

From a general point of view, an **abstract machine** can be seen as a device that produces maximal execution traces of programs using the action of the multi-instructions on the internal states (*cf.* Definition 1.4.3). Formally, it can be defined as a map

$$\mathcal{M} \quad : \quad \Omega \times \{\text{Paml programs}\} \quad \longrightarrow \quad \{\text{execution traces}\}$$

which returns the maximal execution trace of the second argument according to the parameters provided by the first argument. Hence the set Ω gathers all the parameters that drive the behaviour of the machine. For example, the abstract machine is said to be **interleaving** when all the execution traces it produces are interleaving. Therefore an interleaving abstract machine cannot execute more than one single instruction at each tick of the clock. The unique exception comes from the instruction `W(_)` which forces simultaneous execution. More concretely, an abstract machine comes with a **scheduler** that is to say a map

$$\tau \quad : \quad \Omega \times \{\text{internal states}\} \quad \longrightarrow \quad \{\text{subsets of } \mathcal{P}\}$$

such that the set $\tau(\omega, \sigma)$ is admissible in the state σ (*cf.* Remark 1.4.4) and such that $\tau(\omega, \sigma) \neq \emptyset$ unless no other admissible subset is available. The scheduler is thus a parametrized and constrained function of choice. From its scheduler, an abstract machine produces a maximal execution trace of a program P in the obvious way: σ_0 is the initial state associated with P, and for all $n \in \mathbb{N}$, if $\mu_n = \tau(\sigma_n)$ is not empty and the related multi-instruction μ_n does not crash (*cf.* Definition 1.3.13), then $\sigma_{n+1} = \sigma_n \cdot \mu_n$; otherwise the abstract machine stops either on a deadlock or a crash.

Remark 1.6.1. A realistic scheduler should be concerned about all the previous steps of execution. Moreover it should be designed to be fair (*i.e.* to give each non finished process a chance to execute an instruction). Even further it should take a priority scale into account, thus giving the user a reasonable control over its behaviour. For reasons of effectiveness, we could also assume that the scheduler selects maximal sets with respect to inclusion. In doing so, we would maximize parallelism. Unfortunately this requirement cannot be ensured in practice. In fact the behaviour of a scheduler used to test a program should even be partially random.

Remark 1.6.2. The execution of a program heavily depends on the scheduler, over which the programmer has limited control at best. Whenever the program has at least two running processes, it is submitted to this inherent phenomenon in concurrent programming, and there is no reasonable way to rule it out. Still, we would like to guarantee that when the execution finishes, the valuation associated with the final internal state of the machine does not depend on the scheduler. A program satisfying this weaker form of determinism, which is eventually the only significant one, is said to be **deterministic** – see Example 1.6.3.

Example 1.6.3. The left hand program below is not deterministic in the sense that the final content of the variable x depends on which of the processes p and q performs the last assignment. The right hand one is deterministic because the synchronisation barrier forces p to modify x first so its final content is 2.

Non deterministic

```
var:  x = 0

proc:
  p = x := 1
  q = x := 2
init:  p q
```

Deterministic

```
var:  x = 0
sync:  2 b
proc:
  p = x := 1 ; W(b)
  q = W(b) ; x := 2
init:  p q
```

2

Combinatorial Structures

One of the striking achievements of algebraic topology is the relation between the combinatorial and the topological approaches to homotopy theory. In modern mathematics, this fact is formalized by a Quillen equivalence between the model category of simplicial sets and the model category of topological spaces. There are actually several versions of the latter depending on the class of topological spaces one considers (k-spaces, compactly generated spaces – Lewis (1978)) yet they are all Quillen equivalent. The Quillen equivalence between simplicial sets and topological spaces is actually given by the geometric realization – (Hovey, 1999, Theorem 3.6.7). We expect an analogous result in directed topology. In this regard however, the cubical sets (and their realizations in the categories described in Chapter 4) better fit with directed topology than the simplicial sets, see Example 2.2.9. More precisely the realizations of cubical sets in the category of complete filled d-spaces (*cf.* Definition 4.5.14 and Definition 4.5.31) (or the corresponding streams) are fully satisfactory. However, in the directed context, no notion corresponding to Quillen equivalences has been discovered yet. Moreover, from the computer science point of view, the concept of cubical set is too wide so that of precubical sets is preferred.

In Section 2.1, one finds a topology *aide mémoire* gathering up the definitions of compact Hausdorff space and homotopy. The realization/nerve construction is recalled in Section 2.2. The requirements for both functors to exist are extremely weak: from any small category of any cocomplete category with large isomorphism classes (*i.e.* the isomorphism class of any non-initial object is not a set) the nerve functor admits a left adjoint which is, by definition, the realization functor. From the homotopy theory point of view, the overall idea of this approach is that certain presheaf categories can be taken as models of homotopy types – see Cisinski (2006). We loosely borrow from this idea and dedicate the last two sections of the chapter to presheaf categories that both fit with directed topology and concurrency theory. The exposition remains “down to earth”. In Section 2.3, we recall some complete and cocomplete Cartesian closed subcategories of **Top** whose associated simplicial realizations are left exact (*i.e.* preserve all the finite limits). The special cases of cubical and precubical sets are treated in Section 2.4. Both are special instances of the construction described in Section 2.2. Precubical sets are used in Section 3.4 to define the precubical model of Paml programs. The role played by higher dimension mathematical structures emerges on this occasion.

2.1 A Topology Reminder

Since topological spaces are pervasively used in this text, we briefly recall their definition as well as some well-known facts that will be referred to in the subsequent chapters.

Definition 2.1.1. A **topological space** is a set X together with a collection of subsets of X, called the open subsets, such that: the union (resp. the intersection) of any family (resp. any finite family) of open subsets is open. Since the union (resp. the intersection) of the empty family is $\emptyset$ (resp. X), both $\emptyset$ and X are open. The complement of an open subset of X is said to be closed. It is common to denote both the topological space and its underlying set by X. A continuous map (*i.e.* a morphism of topological spaces) is a map between the underlying sets such that the inverse image of an open subset is open. The topological spaces and the continuous maps form the category **Top**.

Example 2.1.2. Any metric space (X, d) induces a topological space whose open subsets are those subsets U such that for all $u \in U$ there exists $\varepsilon > 0$ such that for all $x \in X$, $d(u, x) < \varepsilon \Rightarrow x \in U$. In particular $\mathbb{R}$ is the real line equipped with the distance $d(x, y) = |y - x|$ and $\mathbb{S}^1$ is

$$\left\{ z \in \mathbb{C} \;\middle|\; |z| = 1 \right\}$$

together with the distance $d(z, z') = \min\{\theta, 2\pi - \theta\}$ with θ being the unique element of $[0, 2\pi[$ such that $z' = e^{i\theta} z$. The metric space described above is actually a length space (Burago et al., 2001, Chapter 2). All the topological spaces we will meet in concrete situations arise from a metric space, and most of them are actually length spaces.

Definition 2.1.3. Any subset A of a topological space X inherits a topology from X, its open subsets being of the form $A \cap U$ with U being an open subset of X. We say that A is a **subspace** of X. The **interior** of A is the greatest open subset of X contained in A. The **closure** of A is the least closed subset of X containing A.

Definition 2.1.4. A topological space is said to be Hausdorff when for all $x, y \in X$, $x \neq y$ implies the existence of disjoint open subsets U and V such that $x \in U$ and $y \in V$. The full subcategory of Hausdorff spaces is denoted by **Haus**.

Proposition 2.1.5. The category **Haus** is complete and cocomplete and the inclusion functor **Top** $\hookrightarrow$ **Haus** has a left adjoint.

Proof. See (Mac Lane, 1998, p.135, Proposition 2). □

A collection $\mathcal{U}$ of subsets of X is called a **covering** of X when the union of its elements is X. A **subcovering** of $\mathcal{U}$ is a subcollection of $\mathcal{U}$ whose union of the elements is still X. A covering is said to be open when so are its elements.

Definition 2.1.6. A topological space X is said to be **compact** when one can extract a finite subcovering from any open of its open coverings. A topological space is said to be **locally compact** when all its points admit a compact neighborhood. The full subcategory of **Top** whose objects are the compact spaces is denoted by **Comp** and the one whose objects are the compact Hausdorff spaces is denoted by **CHaus**.

The compact subsets of $\mathbb{R}^n$ are its bounded closed subsets and the compact intervals of $\mathbb{R}$ are the sub-spaces $[a, b]$ with $a \leqslant b$.

Definition 2.1.7. A continuous map from some compact interval of $\mathbb{R}$ to X is called a **path** on X.

Definition 2.1.8. Given two continuous maps $f, g : X \to Y$ and a subspace $A \subseteq X$, a **homotopy** from f to g relative to A is a mapping $\eta : X \times [0, r] \to Y$ with $r \in \mathbb{R}_+$ such that $\eta(\_\,, 0) = f$, $\eta(\_\,, r) = g$, and for all $a \in A$, the mapping $h(a, \_)$ is constant. When $A = \emptyset$ one just writes that η is a homotopy from f to g. Note that:

- the mapping $(a, t) \in X \times [0, r] \mapsto f(a) \in Y$ if a homotopy from f to f relative to all the subsets of X.
- the mapping $(a, t) \in A \times [0, r] \mapsto \eta(a, r - t) \in Y$ is a homotopy from g to f relative to A.
- if $h : X \to Y$ is another map and $\eta' : X \times [0, s] \to Y$ is a homotopy from g to h relative to A, then the mapping $(a, t) \in X \times [0, r + s] \to Y$ defined by $\eta(a, t)$ if $t \leqslant r$ and by $\eta'(a, t - r)$ if $r \geqslant t$, is a homotopy from f to h relative to A.

So the binary relation defined by $f \sim_A g$ when there exists a homotopy from f to g relative to A is an equivalence relation. We write that f and g are homotopic relatively to A or just that f and g are homotopic when $A = \emptyset$. A map is said to be null homotopic when it is homotopic with a constant map.

Definition 2.1.9. A mapping $f : X \to Y$ is called a **homotopy equivalence** when there exists a mapping $g : Y \to X$ such that $g \circ f$ and $f \circ g$ are respectively homotopic with id_X and id_Y.

2.2 Realization and Nerve

Most mathematical formalization of a given concept come with representatives whose properties do not match the intuition. The notion of topology is a striking example of this situation. While it is meant to generalize metrics, not to say $\mathbb{R}^n$, there is a plethora of topologies which fail to satisfy the most elementary properties of metric spaces (*e.g.* order topologies, Zariski topologies, stably compact spaces, and so many others). Denoting the category related to a given formalization by C, a way to prevent these "pathologies" from happening is to choose a class of "nice" objects of C and to consider only those which can be obtained by assembling "nice" objects in a restricted way. Formally speaking, it amounts to choosing a functor $B : \Theta \to C$ and to narrowing the class of objects under consideration to those which can be obtained as colimits of diagrams whose arrows are in the image of B. For technical reason the category Θ should be small. Intuitively, the collection of objects of Θ is the catalog of all the building blocks available in a construction-toy system. Its morphisms indicate all the ways of assembling these blocks allowed by the system. Choosing a functor B specifies which material the building blocks are made of. That intuition is formalized by means of **presheaves**, *viz* functors from the opposite of a small category, in this instance Θ, to **Set**. A presheaf K over Θ plays the role of an assembly manual which explains how to build a certain object from nice ones. An element of K is an ordered pair (x, a) such that $a \in Kx$ for some object x of Θ. A morphism from (x, a) to an element (y, b) is a morphism $f \in \Theta^{\mathrm{op}}(x, y)$ such that $Kf(a) = b$, it should be read as "assemble a with b according to f".

Definition 2.2.1. The elements of K and the morphisms between them form the category $\mathbf{Elts}(K)$ (*cf.* (Borceux, 1994a, p.22) or (Mac Lane and Moerdijk, 1994, p.41)). The B-realization of K in C is, when it exists, the colimit of the functor $F_K : \mathbf{Elts}(K)^{\mathrm{op}} \to C$

defined by $F_K(x,a) = Bx$ and $F_K(f) = Bf$ for all objects (x,a) and all morphisms f of $\mathbf{Elts}(K)^{\mathrm{op}}$. It is denoted by $|K|_B$.

Due to the smallness assumption, presheaves over Θ form, together with natural transformations between them, the category $\mathbf{Set}^{\Theta^{\mathrm{op}}}$. A natural transformation $\eta : K \to K'$ of $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ induces a functor $\overline{\eta} : \mathbf{Elts}(K) \to \mathbf{Elts}(K')$ by setting $\mathbf{Elts}(\eta)(x,a) = (x, \eta_x(a))$ and $\mathbf{Elts}(\eta)(f) = f$ for all morphisms $f : (x,a) \to (y,b)$. This definition makes sense because $K'(f) \circ \eta_x(a) = \eta_y(b)$, the latter equality resulting from naturality of η and $K(f)(a) = b$. The initial cocone over $F_{K'}$, when it exists, can be seen as a natural transformation μ from $F_{K'}$ to the constant functor $|K'|_B$. Then $\mu{\cdot}\overline{\eta}$ is a natural transformation from $F_{K'} \circ \overline{\eta}$ to the constant functor $|K'|_B$. Then observe that $F_{K'} \circ \overline{\eta} = F_K$ hence $\mu \cdot \overline{\eta}$ can be seen as a cocone over F_K. The universal property of colimits gives a morphism $|\eta|_B : |K|_B \to |K'|_B$. A routine verification proves that $|\eta' \circ \eta|_B = |\eta'|_B \circ |\eta|_B$ for any morphism $\eta' : K' \to K''$ of $\mathbf{Set}^{\Theta^{\mathrm{op}}}$. We have extended Definition 2.2.1 to a functor from $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ to C.

Definition 2.2.2. If all the functors F_K have a colimit in C, the B-realization functor is well-defined. It is denoted by $|\_|_B : \mathbf{Set}^{\Theta^{\mathrm{op}}} \to C$.

Let us examine the case where C is **Top** or **Haus**. In particular we have a forgetful functor to **Set** whose left adjoint embeds **Set** in C. Moreover the inclusions of the components of a coproduct are monomorphisms and if X_0, X_1 are the images of two distinct such inclusions, then $X_0 \cap X_1 = \emptyset$. In categorical terms, it more or less amounts to assuming that C has both a terminal object and an initial one, and that all its coproducts are **disjoint** in the sense of (Borceux, 1994c, Def.3.4.8, p.216). Then C canonically contains a copy of **Set** and for all sets S and all objects X of C, the Cartesian product $S \times X$ exists and is isomorphic to the coproduct of S copies of X (*i.e.* the coproduct of the family $\{a\} \times X$ for $a \in S$). In particular, for all $f \in \Theta^{\mathrm{op}}(x,y)$ and all $a \in Kx$, we define $G_K(x,a)$ as $\{a\} \times Bx$ and $G_K(f)$ as below.

$$\{Kf(a)\} \times By \xrightarrow{!\times Bf} \{a\} \times Bx$$

Then $|K|_B$ is again the colimit of G_K and we can carry the toy building set metaphor further saying that K is a construction kit whose building blocks are the objects $\{a\}\times Bx$, and that they are assembled according to the morphisms $G_K(f)$ defined above. Having these additional hypotheses in mind, the relation $Kf(a) = b$ for $(a,b) \in Kx \times Ky$ should be interpreted as: the copy of By labelled by b is identified with some "subspace" (specified by Bf) of the copy of Bx labelled by a.

The realization functor assembles the elements of a construction kit according to an instruction manual, *viz* an object of $\mathbf{Set}^{\Theta^{\mathrm{op}}}$. We are now interested in the reverse procedure: drawing an assembly manual from an object X of C in such a way that its realization is as close as possible to the original object.

Definition 2.2.3 (Nerve)**.** Given an object X of C, the B-**nerve** of X is the presheaf $N(X)$ defined as follows:

- for all objects x of Θ^{op}, $NX(x) := C(Bx, X)$,
- for all morphisms $f \in \Theta^{\mathrm{op}}(x,y)$, $NX(f)$ is the precomposition by Bf. Indeed, if $\gamma \in C(Bx, X)$ then $\gamma \circ Bf \in C(By, X)$.

Given a morphism $g \in C(X,Y)$, one obtains a natural transformation from NX to NY by post-composition. Indeed, if $\delta \in C(\Theta(x), X)$ then $g \circ \delta \in C(\Theta(x), Y)$. The functor N is called the B-**nerve** functor. It is worth noticing that no assumption on C were required during the construction.

Assuming that the B-realization functor is well-defined, the realization of the nerve of X is indeed as close to X as possible. This idea is formalized by the following standard result.

Theorem 2.2.4. (Mac Lane and Moerdijk, 1994, p.41, Theorem 2)**.**
The B-realization and the B-nerve form a pair of adjoint functors.

$$|\_|_B \dashv N_B$$

Remark 2.2.5. In Definition 2.2.2 we stress that only certain colimits are required. Our purpose is to emphasize that Theorem 2.2.4 actually remains valid when C only has the colimits of the functors F_K, for all the objects K of $\mathbf{Set}^{\Theta^{\mathrm{op}}}$. This subtlety will matter when it comes to realizing presheaves in the category of local pospaces (*cf.* Definition 4.3.17).

Remark 2.2.6. When C is cocomplete the B-realization functor is well defined (*cf.* Definition 2.2.2) and by Theorem 2.2.4, it can be defined as the adjoint to the left of the B-nerve. This approach, however concise, relies on abstract results which do not say a word about the intuition on which realizations and nerves are based. The preceding discussion is intended to fill that gap.

The range of applications of Definition 2.2.2 is almost unlimited. We give some examples including the standard and degenerated ones.

Example 2.2.7 (Trivial Realization)**.** The degenerated case where C is the terminal category stresses that the colimits in C might induce identifications in the realization of $K \in \mathbf{Set}^{\Theta^{\mathrm{op}}}$ that are not specified by K.

Example 2.2.8 (Graphs and Bipartite Graphs)**.** Let B be the inclusion of the subcategory Θ of **Top** generated by $\{*\} \to \{0\} \hookrightarrow [0,1]$ and $\{*\} \to \{1\} \hookrightarrow [0,1]$. The category Θ is readily isomorphic to the category freely generated by the graph $v \rightrightarrows a$. An object G of $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ is thus defined by two sets $G(a)$ and $G(v)$ together with two set maps from $G(a)$ to $G(v)$. One easily checks that the category $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ is that of graphs. The realization of the graph with a single vertex and a single arrow is the circle $\mathbb{S}^1$. If we had considered the subcategory Θ of **Top** generated by the graph $\{0\} \hookrightarrow [0,1]$ and $\{1\} \hookrightarrow [0,1]$, then the category Θ would have been isomorphic to the category freely generated by $v_0 \to a \leftarrow v_1$ and $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ would have been the category of bipartite graphs (*i.e.* the graphs with a partition $V^- \cup V^+$ of the set of vertices such that the source (resp. the target) of any arrow belongs to V^- (resp. V^+)). Going back to the case where $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ is the category of graphs, if we had substituted the inclusion functor with the functor B that sends $[0,1]$ to a point, the B-realization of a graph would have been the set (seen as a discrete space) of connected components of the graph.

Example 2.2.9 (Simplicial Realization)**.** The standard n-dimensional simplex can be defined as the convex subset of $\mathbb{R}^n$ generated by the totally ordered set of points $P^n = \{O^n < p_1^n < \ldots < p_n^n\}$ with O^n being the origin of $\mathbb{R}^n$ and p_k^n being the point all coordinates of which are 0 except the k^{th} one which is 1 (*e.g.* the 2-dimensional simplex is generated by $P^2 = \{(0,0) < (1,0) < (0,1)\}$). Erasing a single element of

P^{n+1} defines an affine inclusion from the n-dimensional simplex to one of the $n+1$ faces of the $n+1$-dimensional simplex, namely the one that induces a one-to-one order-preserving map from P^n to P^{n+1}. Moreover duplicating a single element of P^n defines an affine projection from the $n+1$-dimensional simplex to one of its $n+1$ faces, namely the one that induces an order-preserving map from P^{n+1} onto P^n. The common practice in algebraic topology is to consider the category generated by the affine inclusions and the affine projections that we have previously described. This category is isomorphic to the category of non-decreasing maps between finite initial segments of $\mathbb{N}$ (*i.e.* $\{0, \ldots, n-1\}$ for $n \in \mathbb{N}$). It is called the **simplicial** category and usually denoted by Δ. The category $\mathbf{Set}^{\Delta^{\mathrm{op}}}$ is denoted by **SSet** and its objects are called the **simplicial sets**. The left adjoint given by Theorem 2.2.4 in the case that simplices are taken in **Top**, is called the simplicial realization in **Top**. For all $n \in \mathbb{N}$, the simplicial set $\Delta(\_, n)$ is called the standard n-simplex, and its realization in **Top** is the n-dimensional simplex. The adjunction between simplicial realization and simplicial nerve is given by (Goerss and Jardine, 1999, Prop.2.2, p.7) and thoroughly studied in (May, 1967, Chap.III) and (Gabriel and Zisman, 1967, Chap.III) while a more general approach can be found in (Riehl, 2014, Section 1.5, p.12-16).

Example 2.2.10 (Cubical Realization)**.** As the category Δ naturally springs up by considering the n-dimensional simplices for all $n \in \mathbb{N}$, the **box** (or **cubical**) category $\square$ arises by focusing on the n-dimensional cubes (*i.e.* $[0,1]^n$). It is indeed the subcategory of **Top** generated by the **face** maps $\delta^n_{i,\varepsilon}$ and the **degeneracy** maps σ^n_i with $i \in \{0, \ldots, n\}$ and $\varepsilon \in \{0,1\}$.

$$\delta^n_{i,\varepsilon} : (x_0, \ldots, x_{n-1}) \in [0,1]^n \mapsto (x_0, \ldots, x_{i-1}, \varepsilon, x_i, \ldots, x_{n-1}) \in [0,1]^{n+1}$$

$$\sigma^n_i : (x_0, \ldots, x_n) \in [0,1]^{n+1} \mapsto (x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in [0,1]^n$$

The category of **cubical sets**, denoted by **CSet**, is the presheaf category $\mathbf{Set}^{\square^{\mathrm{op}}}$. The adjunction between cubical realization and cubical nerve is given by (Brown et al., 2011, Th.11.1.15, p.371).

Example 2.2.11 (Idle Realization)**.** The presheaf category $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ is complete and cocomplete (Borceux, 1994a, p.89, Corollary 2.15.4) so it is a potential setting for defining a realization functor (*i.e.* $C = \mathbf{Set}^{\Theta^{\mathrm{op}}}$). In this case, the functor B is the Yoneda embedding $\mathcal{Y} : \Theta \to \mathbf{Set}^{\Theta^{\mathrm{op}}}$. In regard with Proposition 2.2.12 the realization functor $|\_|_{\mathcal{Y}}$ is the left Kan extension of $\mathcal{Y}$ along itself. As one can easily guess $|\_|_{\mathcal{Y}} = \mathrm{id}_{\mathbf{Set}^{\Theta^{\mathrm{op}}}}$.

By Theorem 2.2.4, each functor $B : \Theta \to C$ can be associated with a pair of adjoint functors between $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ and C. In fact, that construction extends to an equivalence of categories to which the theoretical content of this section actually boils down. The special case where Θ is the simplicial category (*cf.* Example 2.2.9) is given by (Hovey, 1999, p.76, Proposition 3.1.5).

Proposition 2.2.12. *Let Θ and C be categories, with Θ small and C cocomplete. The category of functors C^{Θ} is equivalent to the category $\mathbf{Adj}(\mathbf{Set}^{\Theta^{\mathrm{op}}}, C)$ of adjunctions from $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ to C.*

We give a brief overview of the proof of Proposition 2.2.12 using all the abstract machinery provided by category theory. In particular we introduce the celebrated Yoneda embedding $\mathcal{Y} : \Theta \to \mathbf{Set}^{\Theta^{\mathrm{op}}}$ (*cf.* (Borceux, 1994a, Theorem 1.3.3, p.11), (Mac Lane, 1998, p.59-62)). Given an object x of Θ the presheaf $\mathcal{Y}x$ is described on Figure 2.1.

$$\begin{tikzcd}
\Theta^{\mathrm{op}} \arrow[r, "\mathcal{Y}x"] & \mathbf{Set} & \mathcal{Y}x \arrow[r, "\mathcal{Y}g"] & \mathcal{Y}x' \\
y \arrow[d, "f"', ""{name=F}] & \Theta(y,x) \arrow[d, "(-\circ f)", ""'{name=G}] & \Theta(y,x) \arrow[r, "(g\circ -)"] \arrow[d, "(-\circ f)"'] & \Theta(y,x') \arrow[d, "(-\circ f)"] \\
y' & \Theta(y',x) & \Theta(y',x) \arrow[r, "(g\circ -)"'] & \Theta(y',x')
\arrow[mapsto, from=F, to=G]
\end{tikzcd}$$

Figure 2.1: The Yoneda embedding in a nutshell.

Moreover any $g \in \Theta(x, x')$ induces, by post-composition, a natural transformation $\mathcal{Y}g : \mathcal{Y}x \to \mathcal{Y}x'$ because the composition law of any category is associative (*cf.* Figure 2.1). Then any functor $G : \mathbf{Set}^{\Theta^{\mathrm{op}}} \to C$ is associated with the object $G \circ \mathcal{Y}$ of C^{Θ}. This association actually extends to a functor which provides the quasi-inverse of the equivalence announced in Proposition 2.2.12. Conversely, we have to build an adjunction from a functor $B : \Theta \to C$. Its left part, namely the B-realization functor (*cf.* Definition 2.2.2), is the left Kan extension of B along $\mathcal{Y}$ (*cf.* (Borceux, 1994a, Theorem 3.7.2 and Proposition 3.7.3, p.123-125)) while the right one is the B-nerve functor (*cf.* Definition 2.2.3).

The category of simplicial sets is the most common combinatorial setting in algebraic topology. From the homotopy theory point of view we can indifferently work with either topological spaces or simplicial sets. Making this statement precise would require that we introduce the notion of model category – Quillen (1967); Hovey (1999). Let us just say that a model category provides a categorical setting for homotopy theory and not surprisingly, **Top** admits a model structure that matches the usual homotopy theory. The interesting point is that **SSet** is also a model category from which a notion of homotopy emerges. The striking fact is that the model categories **SSet** and **Top** induce the same notion of homotopy. In other words they are related by a Quillen equivalence (*i.e.* a pair of adjoint functors that preserves the model category structures). As one can guess the Quillen equivalence is given by the realization functor $|\_| : \mathbf{SSet} \to \mathbf{Top}$ (Hovey, 1999, Theorem 3.6.7). The relation between simplicial sets and topological spaces, and more generally its importance in algebraic topology, is explained in several monographs: see for example Gabriel and Zisman (1967), Quillen (1967), May (1967), (Hovey, 1999, Chapter 3), Goerss and Jardine (1999).

With respect to the Quillen equivalence between **SSet** and **Top**, a computer scientist may ask whether there exist a sound and complete collection of “rewriting rules” such that K and K' are homotopic in **SSet** iff one can turn K into K' by finitely many applications of these rules. In the language of model categories, it amounts to finding a collection of morphisms $\mathcal{A}$ such that the homotopy category of **SSet** is equivalent to the localization $\mathbf{SSet}[\mathcal{A}^{-1}]$. Moreover we would like to have $\mathcal{A}$ as simple as possible. The class $\mathcal{A}$ of **anodyne extensions** (Gabriel and Zisman, 1967, p.60, Chapter 4, section 2) fulfills the above requirements: define Λ^k_n, the k^{th} horn of the standard n-simplex, as the boundary of Δ_n from which the k^{th} face has been removed. The horn inclusions $\Lambda^k_n \hookrightarrow \Delta_n$ are monomorphisms of **SSet** which generate $\mathcal{A}$ in the following sense: it is the least **saturated** collection of monomorphisms (Gabriel and Zisman, 1967, p.60) that contains all of them. Then we deduce from (Hovey, 1999, p.80, Proposition 3.2.3) and (Joyal and Tierney, 2008, p.59, Proposition 3.4.2), that the anodyne extensions are the injective weak equivalences of **SSet**.

Interpreting $\mathscr{A}$ as a rewriting system based on the horn inclusions, the anodyne extensions can be understood as the "contexts" in which one is allowed to "fill" horns and "reduce" simplices without changing the homotopy type of the geometric realization. However the problem of the homotopy between simplicial sets is undecidable (*i.e.* there is no algorithm to decide whether two simplicial sets are homotopic or not).

There is an analogy between the notion of saturated class of monomorphism and the notion of class of weak isomorphisms – see Definition 8.2.4. We will see (in Section 8.6) that any one-way or loop-free category equipped with a system of weak isomorphisms is a homotopical category in the sense of Dwyer et al. (2004).

2.3 A Topological Digression

The categorical structure of $\mathbf{Set}^{\Theta^{\mathrm{op}}}$ is, by construction, strongly related to the one of **Set**. It is thus natural to ask to what extent one can, in the language of categories, express the fact that a category behaves like **Set**. The answer is given by the notion of **topos** (*i.e.* Cartesian closed categories with a subobject classifier (*cf.* Mac Lane and Moerdijk (1994); Goldblatt (1984))). Toposes are especially well-behaved categories so one naturally tries to get as close to them as possible. In particular the morphisms of toposes, also called **geometric morphisms**, are the left adjoint left exact (*i.e.* preserving finite limits) functors. Not surprisingly, any presheaf category is a topos (*cf.* Mac Lane and Moerdijk (1994)). Moreover Theorem 2.2.4 tell us that all realization functors are left adjoint. Then one naturally asks how far a realization functor is from being a geometric morphism. First the category **Top** is not a topos because it has no subobject classifier and it seems that no reasonable subcategory of **Top** can have one. Moreover, the simplicial realization in **Top** does not preserve binary products, nevertheless it remains that if K and K' are finite simplicial sets then their simplicial realization in **Top** satisfies the following relation (*cf.* Milnor (1957), Gabriel and Zisman (1967) or (May, 1967, Th.14.3 p.57)).

$$|K \times K'| \cong |K| \times |K'|$$

Even further the category **Top** is not closed. Actually, there are only few exponentiable objects in the category of topological spaces. However there are cocomplete Cartesian closed full subcategories of **Top** that contains all the finite dimensional simplices and whose associated realization functor is left exact. The simplicial realization and the simplicial nerve in these "convenient" subcategories of **Top** are very well known, they are studied in the third chapter of Gabriel and Zisman (1967) and in the third chapter of May (1967) (to mention some textbooks only). The preceding discussion about the relation between realization and geometric morphisms can be found in (Mac Lane and Moerdijk, 1994, p.454-455). We now recall some common such "convenient" subcategories of **Top**.

Definition 2.3.1. A topological space X it is said to be

- **compactly generated** when any subset U of X is open when $f^{-1}(U)$ is open for all continuous maps f from a compact Hausdorff space K to X.
- **weakly Hausdorff** when the continuous image in X of any compact Hausdorff space K, is closed.

The full subcategory of **Top** whose objects are compactly generated is denoted by **CG**. The full subcategories of **CG** whose objects are Hausdorff (resp. weakly Hausdorff) are respectively denoted by **CGH** and **CGWH**. The compactly generated Hausdorff spaces appeared in Gale (1950) which actually attributes the definition to Hurewicz.

The categories **CG**, **CGH**, and **CGWH** are Cartesian closed, contain all the finite dimensional simplices, and their associated realization functors are left exact. The Cartesian closedness is proven in (Kelley, 1955, p.229-231), Steenrod (1967), (Lewis, 1978, Appendix), (Engelking, 1989, p.148), or (Hovey, 1999, p.58) while proofs of the left exactness can be found in (Gabriel and Zisman, 1967, p.49) and (Goerss and Jardine, 1999, Prop.2.4, p.9). An even more general approach to Cartesian closed subcategories of **Top** is given by (Goubault-Larrecq, 2013, Section 5.7, p.180-194).

Definition 2.3.2. In the sequel, a convenient subcategory of **Top** will refer to either **CG**, **CGH**, or **CGWH**. More generally it could be any complete and cocomplete Cartesian closed subcategory of **Top** in which the simplicial realization preserves finite limits.

Note that the terminology is not standardized: “Kelley spaces” or “k-space” may refer to either of the three notions. Because the underlying space of a partially ordered spaces is Hausdorff, we prefer the category **CGH** as a convenient one. Proposition 2.3.3 and Figure 2.2 (where the solid arrows are inclusions and the dashed ones are their adjoints) gather some common results about these categories, see Lewis (1978), (Engelking, 1989, p.152-155), (Borceux, 1994b, p.359, Section 7.2), and (Hovey, 1999, p.58-59). Weak Hausdorff compactly generated spaces are thoroughly treated in (Strickland (2009)).

Proposition 2.3.3.

1. The inclusion functor **CG** $\hookrightarrow$ **Top** has a right adjoint k. For all topological spaces X, kX is the least compactly generated topology (on the underlying set of X) that contains the topology of X. If X is compactly generated, then $kX = X$. The result still holds if we substitute **CG** and **Top** with **CGH** and **Haus** (resp. **CGWH** and **WH**, the category of weak Hausdorff spaces).

2. The inclusion functor **CGWH** $\hookrightarrow$ **CG** has a left adjoint w. For all compactly generated spaces X, wX is the maximal weak Hausdorff quotient of X. If X is weak Hausdorff, then $wX = X$.

3. The inclusion functor **CGH** $\hookrightarrow$ **CGWH** has a left adjoint h. For all compactly generated spaces weak Hausdorff space X, hX is the maximal Hausdorff quotient of X. If X is Hausdorff, then $hX = X$.

4. The category **CG** is complete and cocomplete: the colimits are taken in **Top**, the limits are obtained by applying k to the limits in **Top**.

5. The category **CGWH** is complete and cocomplete: the limits are taken in **CG**, the colimits are obtained by applying w to the colimits in **CG**.

6. A topological space X is Hausdorff compactly generated iff $X \cong Y/\sim$ for some locally compact space Y and some closed equivalence relation $\sim$ on Y.

7. If X is compactly generated and Y is locally compact, then $X \times Y = X \times_{\mathbf{CG}} Y$

$$\begin{tikzcd}
\mathbf{CG} \arrow[r, shift left] \arrow[d, dashed, shift left, "w"'] & \mathbf{Top} \arrow[l, dashed, shift left, "k"] \arrow[d, dashed, shift left, "w"'] \\
\mathbf{CGWH} \arrow[r, shift left] \arrow[u, shift left] \arrow[d, dashed, shift left, "h"'] & \mathbf{WH} \arrow[l, dashed, shift left, "k"] \arrow[u, shift left] \arrow[d, dashed, shift left, "h"'] \\
\mathbf{CGH} \arrow[r, shift left] \arrow[u, shift left] & \mathbf{Haus} \arrow[l, dashed, shift left, "k"] \arrow[u, shift left]
\end{tikzcd}$$

Figure 2.2: Some remarkable subcategories of **Top**.

8. If X is compactly generated Hausdorff and Y is locally compact Hausdorff, then $X \times Y = X \times_{\mathbf{CGH}} Y$

9. For X, Y topological spaces, U an open subset of Y, and $f \in \mathbf{Top}(K, X)$ with K being compact Hausdorff, define

$$W(f, U) := \{g \in \mathbf{Top}(X, Y) \,|\, g \circ f(K) \subseteq U\}$$

Then $C(X, Y)$ is the topological space over $\mathbf{Top}(X, Y)$ with the topology generated by the family $\{W(f, U); f, U\}$. Define Y^X as $kC(X, Y)$. If X, Y, and Z are compactly generated then

$$\mathbf{CG}(X \times_{\mathbf{CG}} Y, Z) \cong \mathbf{CG}(X, Z^Y)$$

with $X \times_{\mathbf{CG}} Y$ being the image under k of the product $X \times Y$ in **Top** (*i.e.* the product in **CG**). Therefore **CG** is Cartesian closed.

10. Any locally compact Hausdorff space X is compactly generated, exponentiable in **Top**, and for all topological spaces Y, the space Y^X is the homset $\mathbf{Top}(X, Y)$ equipped with the compact-open topology. If Y is compactly generated Hausdorff, then so is the latter space.

From the eighth point of Proposition 2.3.3 a homotopy (*cf.* Definition 2.1.8) can be understood as a path (*cf.* Definition 2.1.7) on the function space Y^X. We have seen (*cf.* Example 2.2.9) that **Top** and **SSet** are (Quillen equivalent) model structures. We also have seen that some full subcategories of **Top** are more tractable than **Top** in practice. However one can reasonably ask whether the notion of homotopy is altered by this replacement. The answer is simple: **CG**, **CGH**, and **CGWH** admit model category structures that are Quillen equivalent to the one of **Top**. From the homotopy theory point of view, we can therefore indifferently work with either of the categories **Top**, **CG**, **CGH**, or **CGWH**. More precisely:

– the inclusion functor $\mathbf{CG} \hookrightarrow \mathbf{Top}$ is a Quillen equivalence, (Hovey, 1999, p.58-59), and

– the left adjoint functor $w : \mathbf{CG} \to \mathbf{CGWH}$ is a Quillen equivalence (Hovey, 1999, p.58-59).

Note that Hovey (1999) does not explicitly treat the case of **CGH**, yet one can reasonably guess that it is also Quillen equivalent to **Top**. Since the first models of directed topology that we will meet are the partially ordered spaces (*cf.* Definition 4.1.1) and since the underlying space of a pospace needs to be Hausdorff (*cf.* Remark 4.1.2), we prefer the category **CGH**.

The remaining of the chapter is divided into two sections respectively dedicated to cubical sets and precubical sets, both of them being examples of the construction made in Section 2.2.

2.4 Cubical Sets

This section is devoted to the presheaf category $\mathbf{Set}^{\square^{\mathrm{op}}}$ introduced in Example 2.2.10. Most of the results presented in this section can be found in (Brown et al., 2011, Chap. 11 Sect. 1). For practical purpose, the box category $\square$ can be extensively described as follows – Crans (1995); Haucourt (2012). We write $[n]$ for $\{0, \dots, n-1\}$; therefore $[0]$ is empty. The set of objects is $\mathbb{N}$ and the homset $\square[n, m]$ is the (finite) set of ordered pairs (n, w) where w is a word of length m on the alphabet $\{0, 1\} \cup \left\{x_0, \dots, x_{n-1}\right\}$ such that for all $i, j \in [n]$ if $w(i) = x_{i'}$, $w(j) = x_{j'}$ and $i < j$, then $i' < j'$. The composition being defined as follows:

$$(w' \circ w)(k) = \begin{cases} w'(k) & \text{if } w'(k) \in \{0, 1\} \\ w(k') & \text{if } w'(k) = x_{k'} \end{cases}$$

The identity of n is represented by the word $(n, x_0 \cdots x_{n-1})$. The face and degeneracy morphisms are therefore represented, for $i \in \{0, \dots, n\}$ and $\varepsilon \in \{0, 1\}$, by the following words:

$$\delta^n_{i,\varepsilon} = (n, x_0 \cdots x_{i-1} \varepsilon x_i \cdots x_{n-1}) \qquad \sigma^n_i = (n+1, x_0 \cdots x_{i-1} x_{i+1} \cdots x_n)$$

We recover the usual cubical relations – see Figure 2.3, and any morphism of $\square$ can be written as a composite of faces and degeneracies in a unique way (provided we impose some extra constraints). For example one has the following decomposition.

$$(5, 01x_0 0x_4 111) \circ (7, x_1 01x_3 0) = (7, 01x_1 00111) = (\delta_{3,0})^2 (\delta_{3,1})^3 \delta_{0,0}\, \delta_{0,1} (\sigma_1)^5 \sigma_0$$

The box category also enjoys a monoidal structure (*cf.* (Mac Lane, 1998, p.161)) given by

$$(n, w) \otimes (n', w') = (n + n', w \cdot (w'|_{x_i \leftarrow x_{i+n}}))$$

where the dot $\cdot$ is the word concatenation and $w'|_{x_i \leftarrow x_{i+n}}$ is obtained from w' replacing each occurrence of x_i by x_{i+n} for all $i \in \{0, \dots, n'-1\}$. However this structure is not symmetric. As a monoidal category $\square$ is generated by $\delta_0 = (0, (0))$, $\delta_1 = (0, (1))$, $\sigma = (1, ())$, and $\mathrm{id} = (1, (x_0))$; while $(0, ())$ is the neutral element of the monoidal product. Note that $\delta_\varepsilon \otimes \sigma = \sigma \otimes \delta_\varepsilon$ for $\varepsilon \in \{0, 1\}$ but $\sigma \otimes \mathrm{id} = (2, (x_1))$ while $\mathrm{id} \otimes \sigma = (2, (x_0))$. For example

$$(7, 01x_1 00111) = \delta_0 \otimes \delta_1 \otimes \sigma \otimes \mathrm{id} \otimes \delta_0^{\otimes 2} \otimes \delta_1^{\otimes 3} \otimes \sigma^{\otimes 5}$$

Remark 2.4.1. Any morphism $(n, w) \in \square(n, m)$ (with m being the length of w) is canonically associated with a non-decreasing map $\phi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ that is defined (for $k \in \{0, \dots, m-1\}$) by

$$\mathrm{proj}_k(\phi(t_0, \dots, t_{n-1})) \quad = \quad \begin{cases} t_{k'} & \text{if } w_k = x_{k'} \\ w_k & \text{if } w_k \in \{0, 1\} \end{cases}$$

This remark is one of the motivations for introducing the notion of framework for directed topology in Section 4.2.

for all $n \in \mathbb{N}$, $i \in [n]$, $j \in [n+1]$ and $\alpha, \beta \in \{0, 1\}$

$$\delta^{(n+1)}_{j,\beta} \circ \delta^{(n)}_{i,\alpha} = \begin{cases} \delta^{(n+1)}_{i,\alpha} \circ \delta^{(n)}_{j-1,\beta} & \text{if } i < j \\ \delta^{(n+1)}_{i+1,\alpha} \circ \delta^{(n)}_{j,\beta} & \text{if } i \geqslant j \end{cases}$$

for all $n \in \mathbb{N}$, $i \in [n+2]$, and $j \in [n+1]$

$$\sigma^{(n)}_{j} \circ \sigma^{(n+1)}_{i} = \begin{cases} \sigma^{(n)}_{i} \circ \sigma^{(n+1)}_{j+1} & \text{if } i \leqslant j \\ \sigma^{(n)}_{i-1} \circ \sigma^{(n+1)}_{j} & \text{if } i > j \end{cases}$$

for all $n \in \mathbb{N}$, $i \in [n+1]$, $j \in [n]$, and $\varepsilon \in \{0, 1\}$

$$\delta^{(n)}_{j,\varepsilon} \circ \sigma^{(n)}_{i} = \begin{cases} \sigma^{(n+1)}_{i} \circ \delta^{(n+1)}_{j+1,\varepsilon} & \text{if } i \leqslant j \\ \sigma^{(n+1)}_{i+1} \circ \delta^{(n+1)}_{j,\varepsilon} & \text{if } i \geqslant j \end{cases}$$

for all $n \in \mathbb{N} \setminus \{0\}$, $i \in [n]$, $j \in [n+1]$, $\varepsilon \in \{0, 1\}$

$$\sigma^{(n)}_{j} \circ \delta^{(n)}_{i,\varepsilon} = \begin{cases} \delta^{(n-1)}_{i,\varepsilon} \circ \sigma^{(n-1)}_{j-1} & \text{if } i < j \\ \mathrm{id}_n & \text{if } i = j \\ \delta^{(n-1)}_{i-1,\varepsilon} \circ \sigma^{(n-1)}_{j} & \text{if } i > j \end{cases}$$

Figure 2.3: Cubical relations

Definition 2.4.2. Mimicking Example 2.2.11, we define the standard n-cube $\square_n$, for $n \in \mathbb{N}$, as the functor $\square(\_, n)$. Given a cubical set K the elements of $K(n)$ are the n-dimensional cubes (or elements) of K. A cube of K is said to be **degenerated** when it is in the image of some map $K(\sigma_i^n)$, and it is said to be **generating** when it is not the face of some non-degenerated cube. The **dimension** of K, which takes its values in $\mathbb{N} \cup \{\infty\}$, is defined as

$$\sup\{\dim c \mid c \text{ is a cube of } K \text{ that is not degenerated}\}$$

Remark 2.4.3. One technical thing to note about cubical sets is that the face and the degeneracy maps go the opposite ways. Consequently, if any of the sets K_n is nonempty, then so are all the others: there is actually much redundancy due to the degeneracy maps. All the relevant information about a cubical set K can however be obtained from its generating cubes and the relations expressed in terms of face and degeneracy maps that are not derived from the cubical relations. All the cubical sets K we will be interested in satisfy the following property: for all cubes x there exists a generating cube y and a morphism f of $\square$ such that $K(f)(y) = x$. We give an example of cubical set, namely $\square_\infty$, that violates it: its n-cubes, for $n \in \mathbb{N}$, are the ordered pairs (n, s) where s is a sequence indexed by $\mathbb{N}$ of elements of $\{0, 1\} \cup \{x_0, \ldots, x_{n-1}\}$ such that:

- $i < j$, $s(i) = x_{i'}$, and $s(j) = x_{j'}$ implies $i' < j'$, and
- there exists a rank beyond which s_k is 0.

Given $(m, w) \in \square(m, n)$ we define $K(w)(n, s)$ as $(m, s \circ w)$ that is to say by updating the dimension and by replacing the occurrence of x_i in the sequence s by the letter of index i in the word w. We note that $\square_d$ canonically embeds into $\square_{d+1}$ (by adding a '0' at the end of each word of length d) and that one actually has

$$\operatorname{colim}\big(\square_0 \subseteq \square_1 \subseteq \cdots \subseteq \square_n \subseteq \square_{n+1} \subseteq \cdots\big) = \square_\infty$$

We also note that, dropping the constraint on the asymptotic behaviour of sequences s, we obtain another cubical set which canonically contains $\square_\infty$ but which, for cardinality reasons, is not isomorphic to it.

Example 2.4.4. Applying the dogma of Remark 2.4.3, the n-standard cube is entirely described by its unique generating element which is n-dimensional. From $K = \square_2$ and its unique generating element s we can describe the directed compact cylinder by adding the relation

$$K(\delta^1_{0,0})(s) = K(\delta^1_{0,1})(s) \text{ [rel. 1]}$$

which identifies two opposite edges of the square. The torus is then obtained by adding the relation

$$K(\delta^1_{1,0})(s) = K(\delta^1_{1,1})(s) \text{ [rel. 2]}$$

Remark 2.4.5. One can easily get convinced that the standard n-cube is not the only cubical set whose realization is $[0, 1]^n$. The subdivision of cubical sets (Krishnan (2013)) is one of the many phenomena that imply the existence of a plethora of (non isomorphic) cubical sets whose realizations are some given (non empty) space.

Anticipating the next chapter, we claim that **CSet** is a more natural setting for algebraic directed topology because the canonical geometric realization of the standard n-cube (*i.e.* $[0, 1]^n$) naturally inherits from the product order of $\mathbb{R}^n$. On the contrary, there is no canonical way to provide the n-simplex with an order. Indeed we could have chosen our n-dimensional simplex as the convex hull of a chain (resp. an antichain) of points of $\mathbb{R}^n$ instead of the one we have chosen in Example 2.2.9. The induced orders are obviously not isomorphic. In classical algebraic topology, the category of simplicial sets **SSet** is often preferred to **CSet** for several technical reasons – see Maltsiniotis (2009). One of them is that the simplicial geometric realization preserves finite limits (if one accepts to work in a convenient subcategory of **Top** – see Section 2.3) while this property fails for the cubical geometric realization (*i.e.* the B-realization where B is the functor that sends $n \in \mathbb{N}$ to $[0,1]^n$ considered as an object of **Top**). The defect is intrinsically due to the Cartesian product of cubical sets (*cf.* Example 2.4.6) and therefore cannot be fixed by substituting **Top** with some of its convenient subcategories.

Example 2.4.6. The standard 1-dimensional cube $\square_1$ is supposed to play the role of the "unit interval" of **CSet** but its 2-fold Cartesian product in **CSet** is rather puzzling since $\square_1 \times \square_1$ is actually isomorphic to $\mathbb{S}^2 \vee \mathbb{S}^1$ (*i.e.* the union of the boundary of $[0,1]^3$ together with the diagonal joining $(0,0,0)$ to $(1,1,1)$). This fact is explained in (Maltsiniotis, 2009, Section 6) and according to it, was known by Daniel M. Kan and made him give up the cubical sets in favour of the simplicial ones. Let us give an intuition about this phenomenon. For $n \in \mathbb{N}$, one has by definition $\square_1(n) = \square(n,1)$. Therefore $\square_1(0) = \{0,1\}$ and $\square_1(n+1) = \square(n,1) \cup \{x_n\}$ for all $n \in \mathbb{N}$. Since the Cartesian product of presheaves is calculated pointwise, we have

$$\square_1 \times \square_1(0) = \{00, 01, 10, 11\}$$

and

$$\square_1 \times \square_1(1) = \square_1 \times \square_1(0) \cup \{0x_0, x_00, x_01, 1x_0, \ \underline{x_0x_0}\ \}$$

and also

$$\square_1 \times \square_1(2) = \square_1 \times \square_1(1) \cup \{0x_1, 0x_1, x_10, x_11, x_0x_1, x_1x_0, \ \underline{x_1x_1}\ \}$$

The "extra" copy of $\mathbb{S}^1$ comes from the elements x_0x_0 and x_1x_1 which are actually identified. The 1-skeleton of $\square_1 \times \square_1$ is depicted on the left hand side of Figure 2.4 while the copy of $\mathbb{S}^2$ is the union of the squares x_0x_1 and x_0x_1 whose boundaries are identified with the outer shape of the 1-skeleton.

Though the cubical approach is not as popular as the simplicial one, it has been the subject of (or a tool in) many publications during the last sixty years: Kan (1955); Brown and Higgins (1981); Antolini (2000, 2002); Jardine (2002); Isaacson (2009); Maltsiniotis (2009); Isaacson (2011) to cite only a few of them, and especially (Serre (1951)) which has introduced the cubical nerve and (Brown et al. (2011)) which gathers in a single book a new foundation for algebraic topology. The overall goal is to find an alternative to the notion of simplicial sets that enjoys all its nice properties without suffering its practical drawbacks (*e.g.* finding a simplicial set whose realization is a given topological spaces is hardly ever an easy task, even for the most common objects of algebraic topology, in particular the decomposition of $\Delta_n \times \Delta_m$ into standard $(n+m)$-simplices is a fundamental construction of the simplicial theory). Yet it remains that the Cartesian product of cubical sets is ill-behaved with respect to homotopy theory (*cf.* Example 2.4.6). An approach consists of adding morphisms to the category $\square$ in a way that one obtains a category $\square'$ so that:

- the mapping sending $n \in \mathbb{N}$ to $[0,1]^n$ extends to a functor B from $\square'$ to some convenient subcategory of **Top**, and

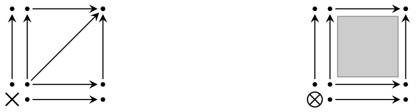


Figure 2.4: The 1-skeleton of $\square_1 \times \square_1$ and the tensor product $\square_1 \otimes \square_1 = \square_2$

– the B-realization is left exact (*i.e.* preserves finite limits).

The cubical sets with connections, introduced in Brown and Higgins (1981), have been proven to be a satisfactory candidate in a very strong sense by Maltsiniotis (2009). As another remedy one can consider the non-symmetric monoidal structure of **CSet** (Kan (1955); Brown et al. (2011)) whose tensor product reflects, through the cubical geometric realization, the Cartesian product of (nice) topological spaces. Provided one works in a convenient subcategory of **Top**, we have indeed

$$|K \otimes K'| \;\cong\; |K| \times |K'|$$

for all cubical sets K and K' – see (Brown et al., 2011, Prop.11.2.3).[1] The tensor product of $K, K' \in$ **CSet** in dimension n (*i.e.* $(K \otimes K')_n$) is the quotient of the union of the products of sets $K_p \times K'_q$ with $p + q = n$, by the equivalence relation that identifies $(K(\sigma_r^r)(x), y)$ and $(x, K(\sigma_0^s)(y))$ with $r + s = n - 1$, see Kan (1955); Brown et al. (2011), and also (Borceux, 1994a, Sect.3.8 p.128) for a general tensor product of set-valued functors.

Example 2.4.7. The two fold tensor product of $\square_1$ is shown on the right hand side Figure 2.4. More generally the n-fold tensor product of $\square_1$ is (isomorphic to) the standard n-cube $\square_n$, and one refers to $\square_2$ and $\square_3$ respectively as the square and the cube. This confirms that the tensor product of **CSet** takes up the role in which one has expected to see the Cartesian product.

Mathematically speaking, the precubical sets are the cubical sets without degeneracies. From the computer science point of view, their interest have appeared with the rise of higher dimensional automata in concurrency theory (*cf.* Remark 2.4.9).

Definition 2.4.8. The category **pCSet** of **precubical sets** is $\mathbf{Set}^{\square^{+\mathrm{op}}}$ with $\square^+$ being the subcategory of **Top** generated by the face maps defined in Section 2.4. We define, for all $n \in \mathbb{N}$, the standard n-cube $\square_n^+$ as the functor $\square^+(\_, n)$. Given a precubical set K the elements of $K(n)$ are the n-dimensional cubes (or elements) of K. For all $n \in \mathbb{N}$, the n^{th} truncature is the endofunctor of **pCSet** that removes all the elements of dimension strictly greater than n from precubical sets.

Remark 2.4.9. The category **Grph** of graphs is canonically isomorphic to the full subcategory of 1-dimensional precubical sets. The other way round, one can define a precubical set as a higher dimensional graph. This point of view leads to the concept of **Higher Dimensional Automata** or **HDA** for short – Pratt (1986, 1991); van Glabbeek (1991); Pratt (2000), which has been proved to encompass almost all the mainstream computer science models of concurrency (*cf.* van Glabbeek (2006); Goubault and Mimram (2012)). At this point we should also mention **asynchronous transition**

[1] the cellular isomorphism $\chi : |K| \otimes |L| \to |K \otimes L|$ between skeletal filtrations induces a homeomorphism $\chi : |K| \times |L| \to |K \otimes L|$ between the underlying topological spaces.

systems – Winskel (1995), which can be seen, in a loose sense, as the 2-dimensional higher dimensional automata.

Remark 2.4.10. The notion of degenerated cube is pointless for precubical sets. A cube of a precubical set is thus said to be **generating** when it is not the face of some other cube. The **dimension** of K, which takes its values in $\mathbb{N} \cup \{\infty\}$, is defined below.

$$\dim(K) \quad = \quad \sup\{n \in \mathbb{N} \mid K_n \neq \emptyset\}$$

Because the Cartesian product of precubical sets is calculated pointwise (*i.e.* $(K \times K')_n = K_n \times K'_n$ for all $n \in \mathbb{N}$), the following relation holds for all precubical sets K and K'.

$$\dim(K \times K') \quad = \quad \min\{\dim(K'), \dim(K)\}$$

Remark 2.4.11. Given a precubical set K, the 1-dimensional truncature of K is a graph. The category P that it freely generates is, by definition, the category of paths on K. Each 2-dimensional element x of K is a square which makes its lower path equivalent to its upper one. Denote by $\sim$ the resulting congruence over P and anticipating on Definition 5.2.10, define $\overrightarrow{\pi_1}K$, the **fundamental category** of K, as the quotient of P by $\sim$. According to this definition, note that a precubical set and its 2-dimensional truncature have the same fundamental category.

Since $\square^+$ is a subcategory of $\square$, the inclusion $\square^+ \hookrightarrow \square$ induces a functor **CSet** $\to$ **pCSet** by precomposition. Also note that $\square^+$ only contains monomorphisms and if the morphism (n, ω) is a composite of face maps, then denoting its codomain (*i.e.* the length of ω) by m, the set of variables occurring in ω is $\{x_0, \ldots, x_{n-1}\}$ with

$$n = m - \#\big\{\, i \in \{0, \ldots, m-1\} \,\big|\, \omega(i) \in \{0,1\} \,\big\}\ .$$

Any morphism (n, ω) of $\square^+$ is thus entirely characterized by ω and we have

$$\omega = \delta^{\omega(i_1)}_{i_1,\, m-n} \circ \cdots \circ \delta^{\omega(i_{m-n})}_{i_{m-n},\, m}$$

with $\{\, i_1 < \ldots < i_{m-n} \,\} = \big\{\, i \in \{0, \ldots, n-1\} \,\big|\, \omega(i) \in \{0,1\} \,\big\}$. By the way, we note that the second subscripts are implicitly given by the length of the word to decompose, so we omit them. Moreover we know from the description of $\square$ given in Section 2.4 that each element of $\{x_0, \ldots, x_{n-1}\}$ occurs in ω exactly once, and that if $\omega(i) = x_{i'}$ and $\omega(j) = x_{j'}$ with $i < j$, then $i' < j'$. The morphism (n, ω) is therefore entirely characterized by the word obtained by substituting each occurrence of a variable in ω with the same variable, let us say x. The homset $\square^+(n, m)$ can therefore be described as the collection of words of length m on the alphabet $\{0, 1, x\}$ with n occurrences of x. For example

$$01xx0x11 = 01x_0x_10x_211 = \delta_0^- \circ \delta_1^+ \circ \delta_4^- \circ \delta_6^+ \circ \delta_7^+$$

Consequently, the (non-symmetric) strict monoidal strcture of $\square^+$ is given by the mere concatenation of words, moreover $\square^+(n, m)$ is empty if $n > m$, and it is a singleton if $n = m$. In particular $\square^+$ is loop-free – see Definition 8.1.1.

Applying the results of Section 2.2, we define the precubical realization functor $|\_|_{\mathbf{pCSet}}$ and the precubical nerve functor $N_{\mathbf{pCSet}}$, the former being the adjoint on the left of the latter. We omit the subscript when there is no ambiguity. From a homotopy point of view, the Cartesian product of **pCSet** does not behave much better than its counterpart in **CSet** – see Example 2.4.13. Yet the category **pCSet** also enjoys a non-symmetric monoidal structure whose tensor product satisfies

$$|K \otimes K'| \;\cong\; |K| \times |K'|$$

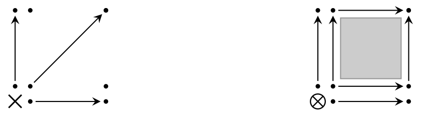


Figure 2.5: The Cartesian product $\square_1^+ \times \square_1^+$ and the tensor product $\square_1^+ \otimes \square_1^+$

for all K and K' in **pCSet** provided that the precubical sets are realized in a convenient subcategory of **Top** (if K and K' are finite, the isomorphism remains valid for realizations in **Top** and **Haus**). Its tensor product is actually even simpler to describe than the one of **CSet**:

Definition 2.4.12. The tensor product of $K, K' \in$ **pCSet** in dimension n (*i.e.* $(K \otimes K')_n$) is the union of the products of sets $K_p \times K'_q$ with $p + q = n$. The faces of $(x, y) \in K_p \times K'_q$ are defined below, for $i \in \{0, \ldots, n-1\}$ and $\varepsilon \in \{0, 1\}$.

$$(K \otimes K')(\delta_{i,\varepsilon}^{p+q-1})(x, y) \quad = \quad \begin{cases} \Big(K(\delta_{i,\varepsilon}^{p-1}) \quad , \quad y\Big) & \text{if } 0 \leqslant i < p \\ \Big(x \quad , \quad K'(\delta_{i-p,\varepsilon}^{q-1})\Big) & \text{if } p \leqslant i < p + q \end{cases}$$

Example 2.4.13. As in the cubical setting, the standard 1-dimensional cube $\square_1^+$ is supposed to play the role of the "unit interval" of **pCSet** and the n-fold tensor product of $\square_1^+$ is (isomorphic to) the standard n-cube $\square_n^+$ – see $\square_2^+$ on the right and side of Figure 2.5. Since the Cartesian product of presheaves is calculated pointwise, we have

$$\square_1^+ \times \square_1^+(0) = \{00, 01, 10, 11\}, \qquad \square_1^+ \times \square_1^+(1) = \{xx\},$$

and $\square_1^+ \times \square_1^+(n) = \emptyset$ for $n \geqslant 2$. In particular $\square_1^+ \times \square_1^+$ is 1-dimensional and disconnected.

Remark 2.4.14. By definition, the precubical sets avoid much redundancy (*e.g.* the set of k-dimensional elements of $\square_n^+$ is empty for $k > n$), yet the description of the cylinder and the one of the torus given in Example 2.4.4 are still valid. The expressiveness of the precubical sets with respect to the cubical ones is an interesting question: can the realization of any cubical set be realized as a precubical one? I have not been able to answer the question in the topological setting, however counter-examples abound in directed topology, and they are not pathological – see Example 4.3.39.

3

Precubical Semantics

of the Parallel Automata Meta Language

We use the precubical sets (*cf.* Definition 2.4.8) and the middle-end representations (*cf.* Definition 1.2.5) to provide each Paml programs with a model which prefigures the continuous one (*cf.* Definition 7.1.2). The mathematical structure carried by this model is a partial precubical set, that is to say a precubical set whose face maps might be partial. These objects naturally arise as we need to be able to remove elements x from a precubical set without removing the elements y whose boundaries contains x. Partial precubical sets were introduced by Fahrenberg and Legay (2015). Nevertheless, they come with a technical burden with which we do not want to deal here. This is also one of the reason why we advocate for using continuous models. Indeed, if the underlying set of a directed realization $\rceil K \lfloor$ of a precubical set K is the set[1]

$$\bigcup_{n\in\mathbb{N}} K_n \times]0,1[^n$$

and if we want to remove the elements contained in the set F, then the following set induces a subobject of $\rceil K \lfloor$.

$$\bigcup_{n\in\mathbb{N}} (K_n \setminus F) \times]0,1[^n$$

3.1 Exhaustive Models

From any Paml program, one can build a partial higher dimensional automata (in the sense of Fahrenberg and Legay (2015)) in a very natural way. Considering precubical sets as higher dimensional graphs, the latter can be seen as the "higher dimensional Cayley graph" (Karrass et al., 2004, p.57) of the action of subsets of $\mathcal{P}$ on internal states (*cf.* Definition 1.4.3). Recall that a program comes with its middle-end representation (*cf.* Definition 1.2.5) in which $\mathcal{P}$ denotes the set of running processes. For technical purpose, we assume that $\mathcal{P}$ is totally ordered so we can define the i^{th} element of any subset of $\mathcal{P}$ for any i less that its cardinal.

[1] This condition is almost always satisfied.

Definition 3.1.1. The **exhaustive model** of a Paml program is a partial higher dimensional automaton whose elements are the ordered pairs (σ, M) where M is an admissible subset of $\mathcal{P}$ in the state σ (*cf.* Remark 1.4.4). The dimension of such an element is the cardinal of M. Given a partial map $\varepsilon : M \to \{0, 1\}$, we define the element $\partial^{\varepsilon}_{M}(\sigma, M)$ as the ordered pair

$$\big(\sigma \cdot M', M \setminus \operatorname{dom} \varepsilon\big)$$

where M' is the set of elements x of $\operatorname{dom} \varepsilon$ such that $\varepsilon(x) = 1$. By definition, it is well-defined if and only if M' is admissible in the state σ and $M \setminus \operatorname{dom} \varepsilon$ is admissible in the state $\sigma \cdot M'$.

Remark 3.1.2. The operators introduced in Definition 3.1.1 encompasses the usual face operators: it suffices to the cases where $\operatorname{dom} \varepsilon$ is a singleton. In particular, a precubical relation at (σ, M) is satisfied when the two corresponding single instructions extracted from the multi-instruction associated with M in the state σ (*cf.* Remark 1.4.4) can be executed in any order without altering the final output. The latter condition is satisfied because for all the elements (σ, M) the set M is admissible in the state σ. However, because the operators are partial, the operators $\partial^{\varepsilon}_{M}(\sigma, M)$ such that $\operatorname{dom} \varepsilon$ is a singleton are no longer generating. Roughly speaking, a composite might exist though some intermediate composite does not. Nevertheless, one readily checks that if no synchronisation instructions appear in the program under consideration, then such a situation does not occur. In other words the exhaustive model is a higher dimension automata in the usual sense.

Remark 3.1.3. The exhaustive model of a Paml program mixes information about control flow, local concurrency, and variable content. Even if one can reduce the size of the exhaustive model of a program by obvious reachability argument, it remains, in almost all cases, infinite. Its interest is thus mainly theoretical. Yet, we will see that under reasonable assumptions, the information about control flow and local concurrency can be gathered in a single finite structure that is defined separately from any variable content consideration.

3.2 Control Flow Graphs

As already mentioned in the introduction of this memoir, (labelled) graphs play a key role in compilers and static analyzers. The control flow graphs introduced in Definition 3.2.1 are inspired from that of F. E. Allen (1970) and from the flowcharts of R. W. Floyd (1967). In particular the vertices of the control flow graphs are labelled with single instructions which are executed when the instruction pointer goes through them. We describe a simple algorithm which produces the control flow graph of every process defined in (the middle-end representation of) a Paml program. Following suit, we add to the middle-end representation of every Paml program a map which associates each of its running processes with its control flow graph.

Definition 3.2.1. A **control flow graph** is a finite graph

$$\begin{tikzcd} G : A \arrow[r, shift left, "\partial^{-}"] \arrow[r, shift right, "\partial^{+}"'] & V \end{tikzcd}$$

together with:

- a distinguished vertex, its **origin**,

- a total order $\sqsubseteq_v$ on $\left\{a \in A \,\middle|\, \partial^- a = v\right\}$ for all $v \in V$,
- a labelling $\lambda_V : V \longrightarrow \{\mathtt{P(\_)}, \mathtt{V(\_)}, \mathtt{W(\_)}, \mathtt{Nop}\}$ (*cf.* Definition 1.1.4), and
- a labelling $\lambda_A : A \longrightarrow \{\text{expressions}\}$ (*cf.* Definition 1.1.2).

These data are submitted to the following constraints: for all $v \in V$,

- if $\left\{a \in A \,\middle|\, \partial^- a = v\right\}$ is not a singleton, then $\lambda_V(v) = \mathtt{Nop}$,
- if $\left\{a \in A \,\middle|\, \partial^- a = v\right\} = \{a\}$, then $\lambda_A(a)$ is the constant 1 (and it should be understood as *true*).

We also impose that all vertices and all arrows of the graph are met by some path starting at its origin.

Remark 3.2.2. The distinguished vertex is the entry point of the process so it seems reasonable to require that all points and vertices can be reached from it. A vertex v with multiple outgoing arrows represents a branching, hence the instruction it carries "does nothing". However the expressions carried by the outgoing arrows together with the total order on them provide the information needed to decide which branch should be taken. In particular, we define the set of free variables $\mathcal{F}(a)$ of an arrow a as the set of free variables of $\lambda_A a$. We also define the set of free variables $\mathcal{F}(v)$ of a vertex v as the union of the sets of free variables of a for all arrows a such that $\partial^- a = v$. This definition will make sense in Definition 3.3.6.

Remark 3.2.3. Control flow graphs should be seen as abstract forms of processes, this fact is illustrated by the execution trace shown on Figure 3.1. It is important to note that in this interpretation, the arrows between nodes are are intermediate positions of the instruction pointer. Moreover, as suggested by Figure 3.1, we assume that the instruction carried by a vertex is executed at the moment the instruction pointer arrives on the vertex.

Definition 3.2.4. A **point** on a (control flow) graph is thus either a vertex or an arrow. A **path** on a (control flow) graph is a sequence of points $\overline{p} = p(0) \cdots p(K)$ whose first and last elements are vertices, and such that for all $k \in \{1, \ldots, K\}$, if $p(k-1)$ (resp. $p(k)$) is an arrow, then $p(k)$ (resp. $p(k-1)$) is its target (resp. its source). From any path on a control flow graph, one deduces a sequence of instructions in the obvious way. A path on a control flow graph is an execution trace when every expressions (carried by an arrow) met along the path is satisfied in the current internal state (*e.g.* Figure 3.1).

Remark 3.2.5. Both examples shown on Figure 3.2 actually produce the same execution traces (up to the occurrences of `Nop`). Referring to "the" control flow graph of a process is therefore an abuse of language. Transformations of control flow graphs are the basis of optimization techniques in compilation. We just provide a simple algorithm to build control flow graphs from basic blocks which we now define.

Definition 3.2.6. A **basic block** is a control flow graph whose underlying graph is a tree (*i.e.* it is acyclic and all its vertices but its origin have a unique ingoing arrow), and whose leaves (*i.e.* its vertices with no outgoing arrow) can be labelled with a jump instruction.

Following Remark 1.1.6, we suppose that we are able to produce the extended basic block of any body of instructions. From the latter, we build the control flow graph associated with the process identifier p as follows:

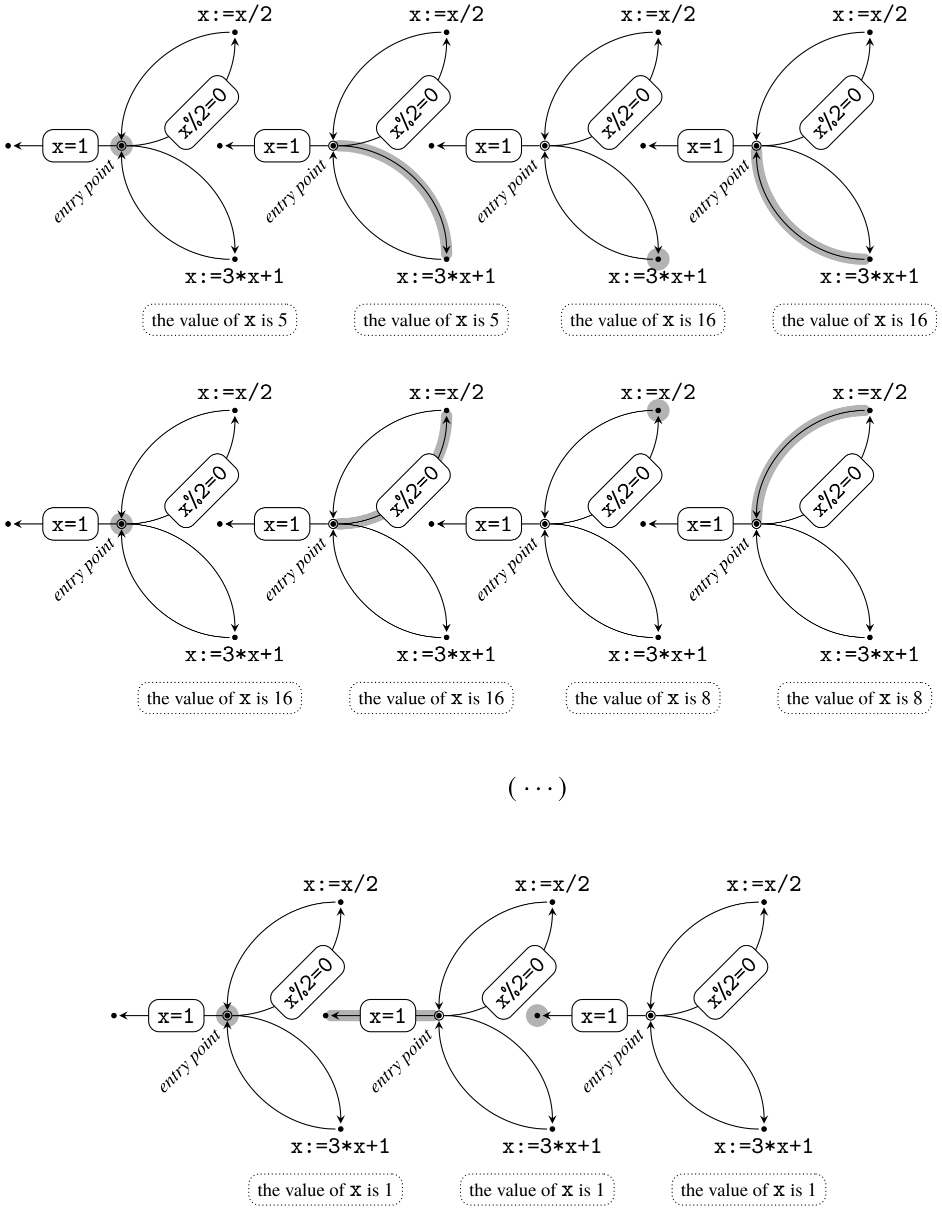


Figure 3.1: An execution trace on a control flow graph

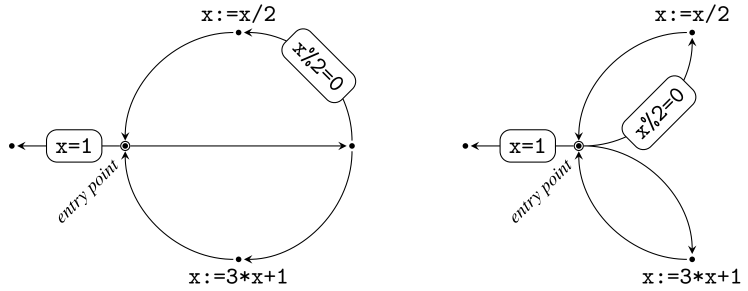


Figure 3.2: Two equivalent control flow graphs

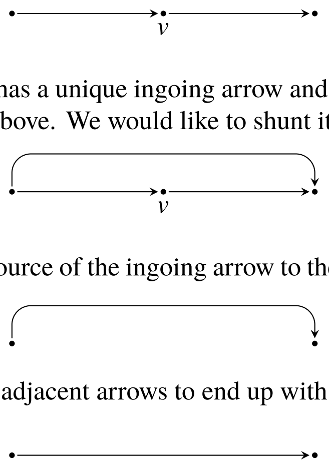
- we consider the disjoint union of all the basic blocks associated with all the elements of $\mathcal{P}$ (*cf.* Definition 1.2.4),
- we draw an arrow (labelled with `1`) from every vertex carrying a jump instruction `J(q)` to the entry point of the basic block of `q`,
- we keep the connected component containing the entry point of `p` (*i.e.* the other ones are removed),
- we "shunt" all the vertices containing a jump instruction (*cf.* Figure 3.3),
- the entry point is that of `p`.

Consider a vertex of the graph obtained at the second step and suppose that its label is a jump instruction. Then one can check that it has a unique ingoing arrow and a unique outgoing arrow, and that they are not equal. For this reason, such a vertex can be shunted.

Remark 3.2.7. The action of multi-instructions on internal states (*cf.* Definition 1.4.3) dynamically replaces the jumps by the corresponding bodies of instructions. In comparison, the control flow graph of a process is produced by connecting the vertices carrying jumps to the entry points of the corresponding basic blocks. In particular, all the instructions `J(_)` are statically treated. Both are inspired from "inlining", a standard transformation that is discussed in (Cooper and Torczon, 2011, Section 8.7.1, p.458) and (Muchnick, 1997, Section 15.2 and 15.3, p.465–472). In the latter case, it is dynamical, in the former one, it is statical.

Remark 3.2.8. As suggested by the way action of multi-instructions on internal states and control flow graphs are defined, the jump instruction has no built-in "return mechanism". This means that even when the destination process finishes, it does not give the control back to the origin process. Once again, Paml is just an intermediate language.

Example 3.2.9. We explain on Figure 3.4 how the control flow graph of the process `does_it_continue` of Remark 1.1.6 is built. For the sake of readability, the identifiers `does_it_continue` and `continue` have been replaced by `p` and `q`. Moreover, when the label of a vertex has been omitted, it should be understood as the instruction `Nop`. An unlabelled arrow a is the one that should be chosen when none of the expressions carried by the arrows sharing the same source as a is satisfied. In more concrete terms, it is the default branch of a "match case" construction.



Assuming that the vertex v has a unique ingoing arrow and a unique outgoing arrow as above. We would like to shunt it.

We add an arrow from the source of the ingoing arrow to the target of the outgoing one.

We remove v and its adjacent arrows to end up with the following graph.

Figure 3.3: Shunting a vertex.

Remark 3.2.10. Control flow graphs have been defined in such a manner that any execution trace of a single process corresponds to a path on its control flow graph. However the converse is false. Indeed, considering the Hasse-Syracuse algorithm (*cf.* Figure 3.4), we note that no path going through `x:=3*x+1` twice in a row matches an execution trace. The set of paths on a control flow graph is thus an overapproximation of the set of execution traces.

From now on, we assume that the middle-end representation of every Paml program (*cf.* Definition 1.2.5) comes with a map

$$\mathrm{cfg} : \mathcal{P} \to \{\text{control flow graphs}\}$$

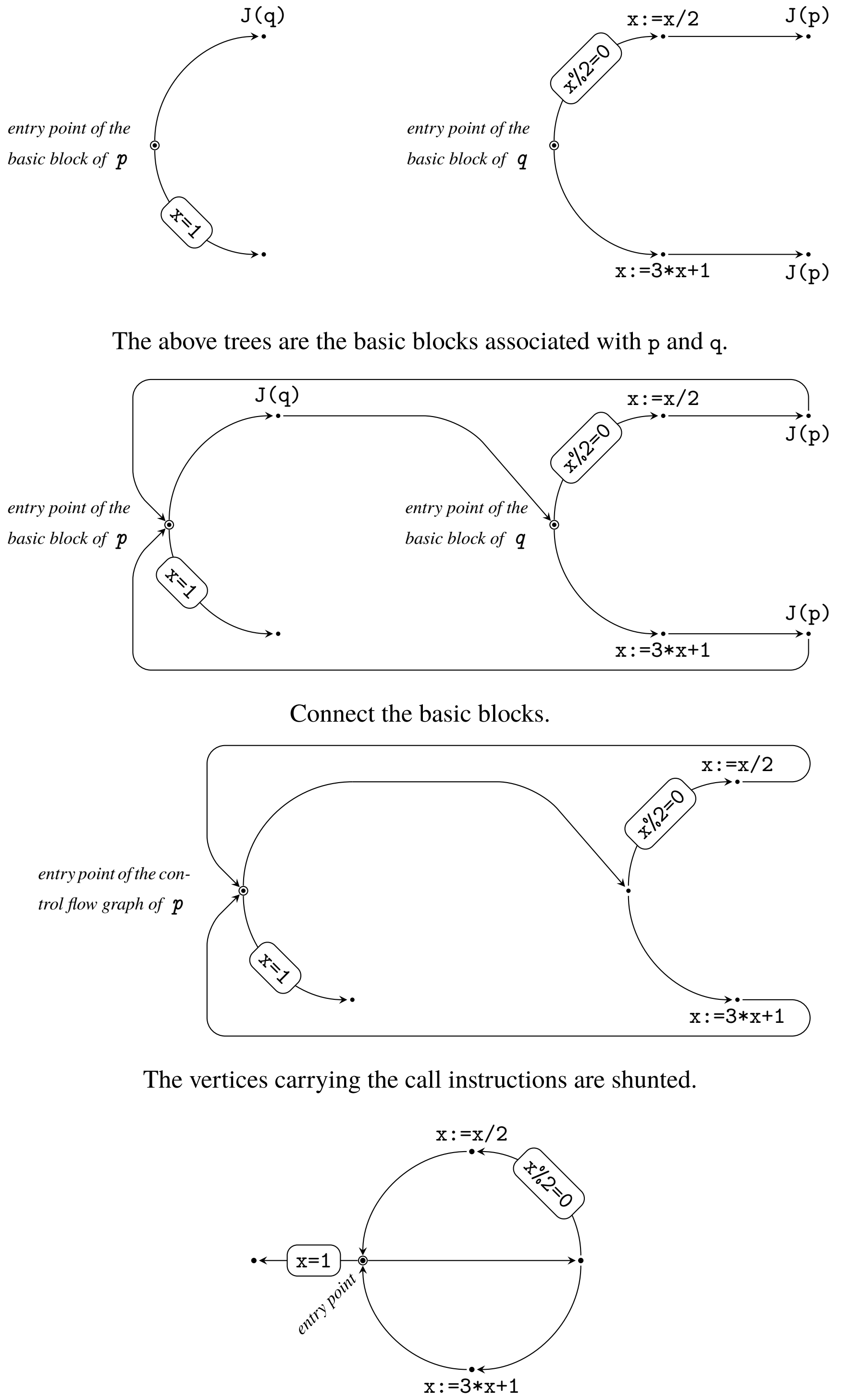
relating each element p of $\mathcal{P}$ with a control flow graph that “faithfully” represents body(p) (*cf.* Definition 1.2.4) in the sense that any execution trace on a control flow graph (*cf.* Definition 3.2.4) is actually (up to the occurrences of the `Nop` instruction) an execution trace in the sense of Definition 1.4.5. A rigorous proof of this fact would be tedious, we do not provide it here.

3.3 Another Abstract Machine

In this section, we describe an alternative abstract machine based on control flow graphs instead of stack of instructions (*cf.* Definition 1.4.3 and Section 1.6). The first step consists of providing every Paml program with a control flow structure. For technical purpose, we assume that $\mathcal{P}$ is totally ordered so it is isomorphic with the set of integers $\{1, \ldots, N\}$ for some $N \in \mathbb{N}$. Following Remark 3.2.3 we generalize the notion of point (*cf.* Definition 3.2.4) to higher dimensions in the obvious way.

Definition 3.3.1. A **point** is an N-tuple whose n^{th} component, for $n \in \{1, \ldots, N\}$, is a point of the control flow graph G_n of the n^{th} running process of the program (*cf.* Definition 3.2.4). Considering graphs as 1-dimensional precubical sets, the points are precisely the elements of the following tensor product.

$$G_1 \otimes \cdots \otimes G_n$$


The above trees are the basic blocks associated with p and q.

Connect the basic blocks.

The vertices carrying the call instructions are shunted.

A more concise view of the preceding control flow graph.

J(q)
entry point of the basic block of p
x=1
x:=x/2
J(p)
x%2=0
entry point of the basic block of q
x:=3*x+1
J(p)
J(q)
x:=x/2
J(p)
entry point of the basic block of p
x=1
entry point of the basic block of q
J(p)
x:=3*x+1
x:=x/2
x%2=0
entry point of the control flow graph of p
x=1
x:=3*x+1
x:=x/2
x%2=0
x=1
entry point
x:=3*x+1

Figure 3.4: Building the control flow graph of the Hasse/Syracuse algorithm.

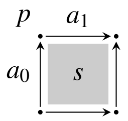


Figure 3.5: Discrete directed paths are “continuous”.

Extending the notion of path is a bit subtler.

Definition 3.3.2. A **path** on a given tuple of graphs $(G_1, \ldots, G_n)$, typically the running processes of a program, is a finite sequence of points $\overline{p} = p(0) \cdots p(K)$ such that for all $k \in \{1, \ldots, K\}$ we have

– execution: $\partial^+ p_n(k-1) = p_n(k)$ for all $n \in D_k$, or

– selection: $\partial^- p_n(k) = p_n(k-1)$ for all $n \in D_k$,

where $D_k = \big\{n \in \{1, \ldots, N\} \,\big|\, p_n(k) \neq p_n(k-1)\big\}$.

Remark 3.3.3. The constraints imposed on the notion of path might seem surprising at the first sight. They actually force paths to be “continuous” in the following sense. Assume that the picture on Figure 3.5 is the unit square $[0,1]^2$ split into points (its corners), open segments (the interior of its edges), and open unit square (its interior). A naive definition would accept any sequence of points such that $p_n(k) = p_n(k+1)$, $\partial^- p_n(k+1) = p_n(k)$, or $\partial^+ p_n(k) = p_n(k+1)$ for all $n \in \{1, \ldots, N\}$ and all $k \in \{0, \ldots, K-1\}$. In particular, it would accept the sequence (a_0, a_1) on Figure 3.5 although no continuous path on the unit square can go from a_0 to a_1 without meeting p or s. The terminology “execution” and “selection” will be explained later in this section.

Remark 3.3.4. Assuming that the set of vertices and that of arrows are disjoint and given $k \in \{0, \ldots, K\}$, if both conditions “execution” and “selection” are satisfied, then $p(k) = p(k+1)$.

We provide the points introduced in Definition 3.3.1 with a labelling.

Definition 3.3.5. The label $\lambda(p)$ of a point $p = (p_1, \ldots, p_N)$ is $(\lambda_1(p_1), \ldots, \lambda_N(p_N))$ where λ_n refers to the labelling of G_n as described in Definition 3.2.1. We thus have a labelling λ on the set of elements of $G_1 \otimes \cdots \otimes G_n$.

Definition 3.3.6. A **state** of the new abstract machine is an ordered pair made of a context (*cf.* Definition 1.3.7) and a point (*cf.* Definition 3.3.1) which respectively indicate the current state of the memory, the current distribution of resources, and the current position of the instruction pointer. Let μ be a multi-instruction. Having in mind that the elements of dom μ are the identifiers of the processes that try to simultaneously execute their current instruction, we say that μ is **admissible** in the state (σ, p) (*i.e.* σ is a context and $p = (p_1, \ldots, p_N)$ is a point) when it is admissible in the context σ (*cf.* Definition 1.3.12).

Definition 3.3.7. Each path is associated with a sequence $M_1 \cdots M_K$ of subsets of $\mathcal{P}$ which is described thereinafter for $k \in \{1, \ldots, K\}$.

$$M_k \quad = \quad \big\{n \in \{1, \ldots, N\} \,\big|\, p_n(k) = \partial^+ p_n(k-1) \text{ or } \lambda_n(p_n(k)) = W(\_)\big\}$$

For all $k \in \{1, \ldots, K\}$ we have the corresponding multi-instruction μ_k which is defined by setting $\mu_k(n) = \lambda_n(p_n(k))$ for all $n \in M_k$. This convention implements a specific behaviour: any instruction is triggered at the very moment it is reached by the instruction pointer. It also reveals that the instruction $W(\_)$ is persistent in the sense that it remains active until the instruction pointer leaves the vertex carrying it. On the contrary, the instructions $P(\_)$ and $V(\_)$ instantly alter the internal state of the machine where after its effect is over. Given σ the initial state of the program, a path is said to be **admissible** when for all $k \in \{0, \ldots, K\}$, the multi-instruction μ_k is admissible (*cf.* Definition 3.3.6) at state $\sigma \cdot \mu_0 \cdots \mu_{k-1}$, in this case we define the **action** of $\overline{p}$ on the right of σ as follows.

$$\sigma \cdot \overline{p} \quad = \quad \sigma \cdot \mu_0 \cdots \mu_K$$

This suggests that the entry point of a control flow graph be labelled with the `Nop` instruction. The terminology "execution" introduced in Definition 3.3.2 comes from that action. An admissible path $\overline{p}$ is an **execution trace** when all the conditional branchings met along the way are respected, in other words when for all $k \in \{1, \ldots, K\}$, for all $n \in \{1, \ldots, N\}$ such that $\partial^- p_n(k) = p_n(k-1)$, the arrow $p_n(k)$ is the least element of the following set (*cf.* Definition 3.2.1)

$$\left\{ \, a \text{ arrow of } G_n \;\middle|\; \partial^- a = p_n(k-1) \text{ and } [\![\lambda_n(a)]\!]_{\sigma \cdot \mu_0 \cdots \mu_{k-1}} \neq 0 \, \right\}$$

Notation $[\![\_]\!]$ refers to Definition 1.3.3, it only depends on the context of evaluation of a state. By Remark 3.3.4, if there exists $n \in \{1, \ldots, N\}$ such that $\partial^- p_n(k) = p_n(k-1)$ then there is no $n \in \{1, \ldots, N\}$ such that $\partial^+ p_n(k-1) = p_n(k)$, hence the evaluation of the expressions cannot be disturbed. The terminology "selection" introduced in Definition 3.3.2 comes from that remark. It is related to the convention that an instruction is executed at the very moment it is reached by the instruction pointer.

Remark 3.3.8. Definition 3.3.7 has been adjusted so that, provided that the last point of a path $\overline{p}$ is the first point of another path $\overline{p}'$, the sequence of multi-instructions associated with the concatenation $\overline{pp}'$ is the concatenation of the associated sequences of multi-instructions (up to occurrences of the empty ones). Moreover, it is convenient to say that an expression ε is "met along" a path $\overline{p}$ when it is the label of a point $p_n(k)$ such that $\partial^- p_n(k) = p_n(k-1)$ or when $\partial^+ p_n(k-1) = p_n(k)$ and the label of the point $p_n(k)$ is an assignment whose right-hand part is ε.

Remark 3.3.9. Any path is therefore associated with a sequence of executions and selections. During an execution step, some processes simultaneously execute their current instruction. One has to check at runtime that the corresponding multi-instruction is admissible. During a selection step, some processes decide which instruction will be executed next. If that path is actually an execution trace, then the choice is in accordance with the conditions carried by the arrows.

Remark 3.3.10. The notion of execution trace from Definition 1.4.5 is not "equivalent" to that of Definition 3.3.7. This is mainly due to the fact that branching instructions are atomic in the latter, not in the former. More precisely there might be several execution step between the "moment" that a branching point is reached, and the moment that the expression carried by the outgoing arrows are evaluated. This can be fixed up imposing a selection step after every execution step together with some extra constraints on the admissible multi-instructions that reflect the ones imposed to them by Definition 1.4.5. More precisely, an execution trace in the sense of Definition 3.3.7 along with which every process that executed an instruction at a given step selects an outgoing arrow at the next step, is actually an execution trace in the sense of Definition 1.4.5. The distinction vanishes when the program under consideration has a single running process.

3.4 Discrete Models of Conservative Programs

Until now, admissibility (*cf.* Definitions 1.3.12 and 3.3.6) had to be checked at runtime. The purpose of this section is to tackle this problem. Suppose for example that we want to execute the multi-instruction $(P(a), P(a))$. Clearly, we should be able to execute it concurrently iff (at least) two occurrences of the semaphore x are available. However the availability of a semaphore is a matter of internal state of the abstract machine. In order to get rid of this runtime dependency we would like to encode the resource availability in the precubical control flow of the program, which cannot be achieved without confining the range of "accepted" Paml programs. Informally speaking, the engaged amount of resources should only depend on the position of the instruction pointer.

From a physicist point of view, the situation suggests to compare the action of a directed path on the semaphores sharing with the work of a force along a curve. As a physicist saying that a force is conservative when this work only depends on the starting and the ending of the curve, we say that a process is conservative when the action (on the abstract machine states) of a directed path (over the control flow graph) only depends on its source and its target.

From computer science considerations, the similar notion of *wellbehaved processes* was independently introduced in Fahrenberg (2002) without reference to the analogy with Physics. More generally, the notion is related to the idea that values of variables can be encoded in the states of a system.

Definition 3.4.1. Given a control flow graph G, the **neutralized** control flow graph $N(G)$ associated with G is obtained by replacing every instruction but `P(_)` and `V(_)` by `Nop`. In particular both G and $N(G)$ have the same entry point and the semaphores appearing in $N(G)$ are the ones appearing in G. Assume that the arities of all the semaphores appearing in G (and therefore in $N(G)$) are infinite. In particular every path on $N(G)$ is admissible and no crash can occur. The control flow graph G is said to be **conservative** when for all initial states σ and for all paths $\overline{p}$ on $N(G)$ starting at the initial point, the restriction of $\sigma \cdot \overline{p}$ to $\mathcal{S}$ only depends on the endpoint of $\overline{p}$. A middle-end representation is **conservative** when it associates every element of $\mathcal{P}$ with a conservative control flow graph, in other words when the mapping introduced at the end of Section 3.2 actually has the following type.

$$\text{cfg} : \mathcal{P} \to \{\text{conservative control flow graphs}\}$$

Remark 3.4.2. We could have parametrized the notion of conservative control flow graph with the map assigning each semaphore to its arity. Following this approach, the control flow graph depicted on the right-hand part of Figure 3.6 is conservative iff arity `s` $\leqslant 1$. Indeed, for greater arity, the path winding around the circle is admissible. A control flow graph is conservative in the sense of Definition 3.4.1 when it is conservative with respect to any arity. For similar reasons, we have "neutralized" assignments so that no crash (*i.e.* zero division in our restricted language) can occur. In particular, the values initially assigned by σ to the variables appearing in G do not play any role in the conservativity of G. Hence we only have to test the criterion given in Definition 3.4.1 for one initial state σ.

Remark 3.4.3. Claiming that a program is conservative (as suggested by the title of this section) is actually an abuse of language that deserves clarification. Being conservative is indeed a property that applies to control flow graphs. Whether a

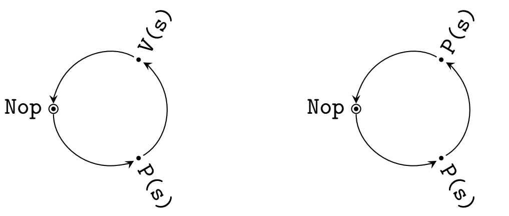


Figure 3.6: Conservative *vs* nonconservative loops.

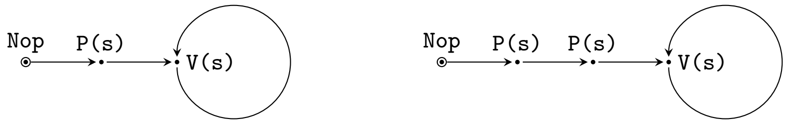


Figure 3.7: Conservative *vs* nonconservative lollipops.

program is conservative or not thus heavily depends on the way the control flow graphs of its running processes are defined. For example the Paml process

```
p = P(s);V(s);J(p)
```

is conservative – see the left-hand control flow graph on Figure 3.6 while

```
p = P(s);J(p)
```

is not – see Figure 3.8, since infinite control flow graphs are obviously prohibitive in practice. The V(_) instruction is non-blocking and according to its semantics, the number of semaphore tokens held by a process is non-negative. Since the amount of tokens held by the processes is assumed to be zero at the beginning of a program execution, the left-hand control flow graph on Figure 3.7 is conservative while the right-hand one is not. As a matter of branchings, the Paml process

```
P(a);(x:=0 + x:=1);V(a)
```

is conservative – see Figure 3.9 while

```
P(a);(x:=0 + (x:=1;V(a)))
```

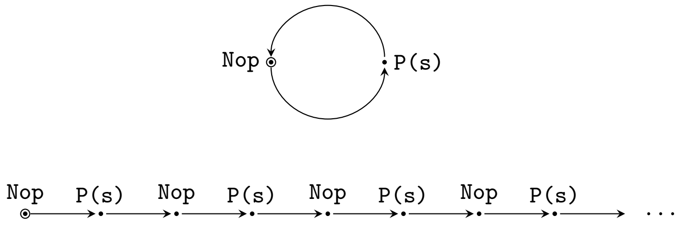


Figure 3.8: A nonconservative loop and its conservative (but infinite) unfolding.

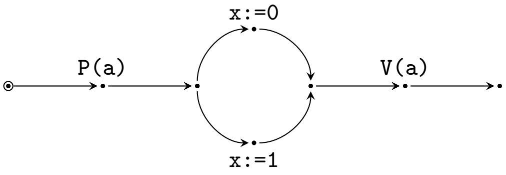


Figure 3.9: A conservative branching.

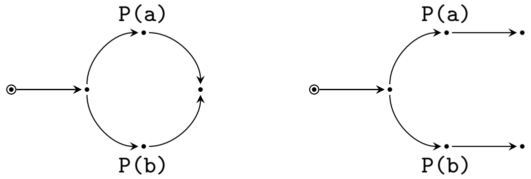


Figure 3.10: Conservative process may be obtained by duplicating vertex.

is worth a closer examination – see Figure 3.10. The latter suggests that certain nonconservative control flow graphs can be turned into conservative ones by a mild transformation, the overall idea being that we try to encode the amount of semaphore occurrences held into a finite control flow graph. As shown by Figure 3.8 there are situations where no such transformation is possible. For practical reasons, it is always convenient to merge the output vertices of a branching. Nevertheless, for the sake of simplicity of control flow graph description, we did not include this optimization in Section 3.2.

Remark 3.4.4. From a theoretical point of view, it may yet be convenient to encode the system states as vertices of the control flow graph, which sometimes amounts to resorting to infinite (conservative) control flow graphs. The question is related to the model described in Section 3.1 as well as to the notion of **unfolding** of a directed graph, and by extension to the directed universal covering problem – see Section 10.4.

Remark 3.4.5. In physics a force is traditionally defined as a vector field over a (simply connected) smooth manifold, and it is said to be conservative when its work along a curve only depends on its extremities. In a very loose analogy control flow graphs can be seen as vector fields[2]. Pushing on with the analogy, the control flow graphs of (the running processes of) a Paml program form a parallelization of the tensor product $G_1 \otimes \cdots \otimes G_d$ (*cf.* the brief note about parallelizable manifolds at the end of Section 4.5).

Remark 3.4.6. Conservativity is only concerned with resources. Following the physicist metaphor, we restrict to programs in which the amount of available resources provided by the system can be seen as a potential; in other words the ones whose resource consumption depend neither on branchings nor loop iterations.

The property of being conservative can be checked by a breadth-first traversal algorithm which is similar to the one found in Fahrenberg (2002) to detect well-behaved processes. Let us denote the commutative monoid of multisets over $\mathcal{S}$ by $\mathbb{N}^{\mathcal{S}}$. By a

[2] This way of thinking arises from a discussion with Samuel Mimram and his will to explore relations between computer science and physics.

slight abuse of notation we write 0 for empty multiset (*i.e.* the map sending all $s \in \mathcal{S}$ to 0). The instructions `P(_)` and `V(_)` acts on the right of multisets over $\mathcal{S}$ in the obvious way: $\pi \cdot \texttt{P}(s) = \pi + \{s\}$ and $\pi \cdot \texttt{V}(s) = \pi - \{s\}$. We inductively define a sequence of partial function of type {points} $\longrightarrow \mathbb{N}^{\mathcal{S}}$. The first term π_0 is only defined at the origin of the graph and $\pi_0(\text{origin}) = 0$. Assuming that π_n is defined, for each ordered pair of points (p, p') such that:

- $\pi_n(p)$ is defined but not $\pi_n(p')$, and
- $\partial^- p' = p$ or $p' = \partial^+ p$,

we define a strict extension of π_n, by assigning to p' the following element on $\mathbb{N}$.

$$p' \mapsto \begin{cases} \pi_n(p) & \text{if } \partial^- p' = p \\ \pi_n(p) \cdot \lambda(p') & \text{if } p' = \partial^+ p \end{cases}$$

If all these extensions are compatible, then π_{n+1} is their union. Otherwise the induction stops and the graph is not conservative. In case the induction does not stop we have, due to the finiteness of the set of points, a stationary chain of extensions whose limit is denoted by π.

$$\pi_0 \subseteq \cdots \subseteq \pi_n \subseteq \pi_{n+1} \subseteq \cdots$$

For every point of a control flow graph can be reached from its origin (*cf.* Definition 3.2.1) the mapping π is defined everywhere. If the condition

$$\pi(p') \quad = \quad \begin{cases} \pi(p) & \text{if } \partial^- p' = p \\ \pi(p) \cdot \lambda(p') & \text{if } p' = \partial^+ p \end{cases}$$

holds for all ordered pairs of points (p, p') such that $\partial^- p' = p$ or $p' = \partial^+ p$, then G is conservative, otherwise it is not.

Example 3.4.7. The upper left part of Figure 3.11 is a control flow graph. The upper right one displays its neutralized form, the highlighted part is the domain of definition of π_5. We have $\pi_5(s, p) = 0$ for all points p of this domain. The partial map π_9 is shown on the lower left part. The points p in the thick highlighted part satisfy $\pi_9(s, p) = 1$. The total map π_{15} is depicted on the last part of Figure 3.11. For all points p, we have $\pi_{15}(s, p) = 1$ if p belongs to the thick highlighted part, else $\pi_{15}(s, p) = 0$. The control flow graph is thus conservative. In Figure 3.12, the label `V(s)` has been replaced by `x++`. The resulting control flow graph is not conservative. Applying the algorithm to the left hand control flow graph on Figure 3.10 results in a sequence $\pi_0, \ldots, \pi_5$, the map π_6 being undefined. On the contrary, if the algorithm is applied to the right hand control flow graph on Figure 3.6 one obtains a limit $\pi = \pi_5$, however it is not a potential function (as in the case of Figure 3.12).

Definition 3.4.8. Pushing the analogy with conservative forces in physics, the control flow graph G is conservative if and only if it comes with a **potential** funtion, in other words a map F : {points of G} $\times\, \mathcal{S} \longrightarrow \mathbb{N}$ such that for all paths $\overline{p}$ on G starting at the initial point of G, one has the following equality with p and σ being the endpoint of $\overline{p}$ and an initial state.

$$(\sigma \cdot \overline{p})(s) \quad = \quad F(p, s)$$

The potential function of a conservative control flow graph is actually the limit of the sequence $(\pi_n)_{n \in \mathbb{N}}$ described above. The **potential** function F : {points of the program} $\times$ $\mathcal{S} \longrightarrow \mathbb{N}$ of a conservative middle-end representation is thus defined as the sum of the

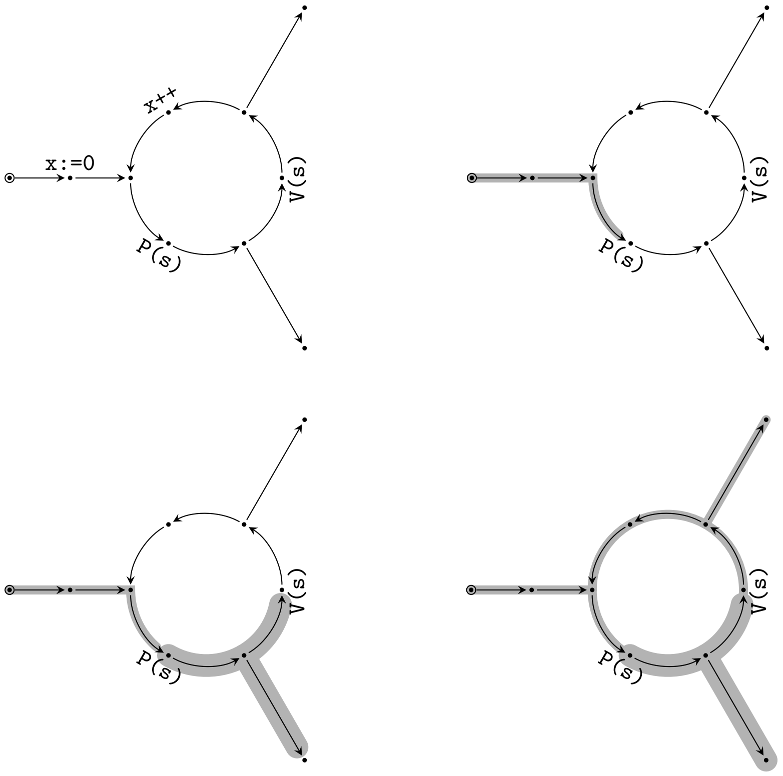


Figure 3.11: Conservativity algorithm applied to a conservative control flow graph

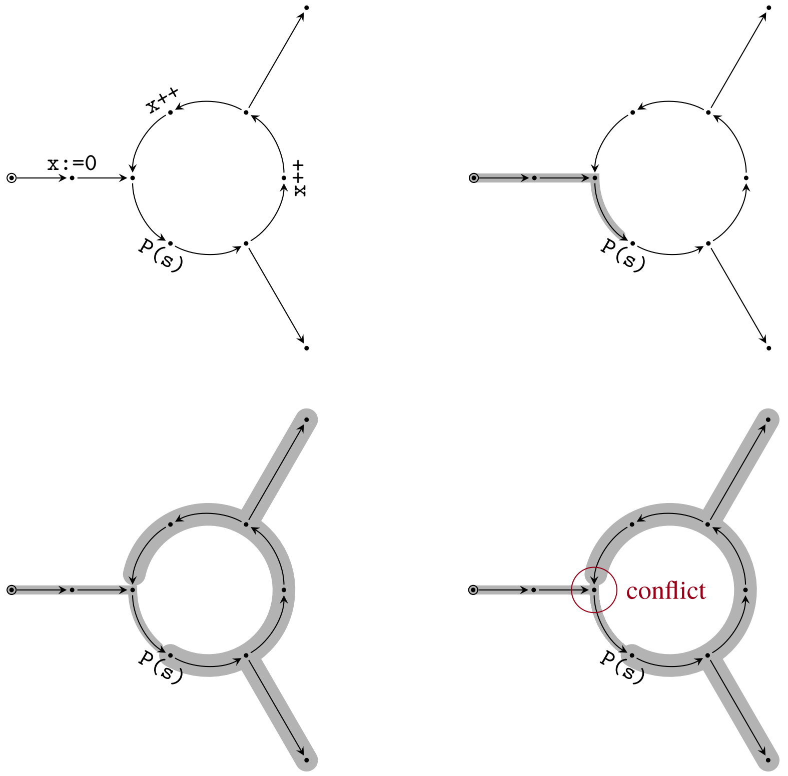


Figure 3.12: Conservativity algorithm applied to a non-conservative control flow graph

potential functions of the control flow graphs $G_1, \ldots G_n$ of the running processes. Keeping in mind that a point of the program is a tuple $(p_1, \ldots, p_n)$ where each p_i is a point of G_i, the function F is given by the formula below. By analogy with multilinear algebra, it is the product of a tuple of linear forms and a tuple of vectors.

$$F(p_1, \ldots, p_n, s) \quad = \quad \sum_{i=1}^{n} F_i(p_i, s)$$

Remark 3.4.9. It is worth mentioning that the complexity of the algorithm that determines whether a control flow graph is conservative or not is linear with respect to its number of arrows. Compared to the complexity of the algorithms used to build the geometric model of a program (*cf.* Definitions 3.4.10 and 7.1.2), the overhead can be neglected.

We introduce the discrete models of conservative middle-end representations and give a technical result that will be used to prove Theorem 7.2.4 which is one of the motivation for introducing topological methods in the study on concurrency. We denote the potential function of that representation by F (*cf.* Definition 3.4.8).

Definition 3.4.10. A point $p = (p_1, \ldots, p_N)$ of some conservative middle-end representation is said to be:

- **conflicting** when $\lambda_n(p_n)$ and $\lambda_m(p_m)$ conflict for some $n, m \in \{1, \ldots, N\}$ such that $n \neq m$ (*cf.* Definition 1.3.9),
- **exhausting** when there exists a semaphore $s \in \mathcal{S}$ such that

$$F(p_1, \ldots, p_N, s) \quad > \quad \text{arity}(s)\,,$$

- **synchronising** when there is some synchronisation barrier $b \in \mathcal{B}$ such that

$$0 \quad < \quad \text{card}\big\{n \in \{1, \ldots, N\} \,\big|\, \lambda_n(p_n) = W(b)\big\} \quad \leqslant \quad \text{arity}(b)\,,$$

- **terminal** when the sequence of multi-instructions associated with any directed path starting at p only contains trivial multi-instructions (*cf.* Definitions 1.3.8 and 3.3.2).

The **forbidden** set of the program P gathers all the conflicting, exhausting , and synchronising points.

$$\{\text{fobidden}\} \quad = \quad \{\text{conflicting}\} \cup \{\text{exhausting}\} \cup \{\text{synchronising}\}$$

The **discrete model** of the program is the complement of its forbidden set.

$$\{\text{points of the program}\} \setminus \{\text{forbidden points}\}$$

A **deadlock** is a non-terminal point p such that any directed paths on the model starting at p is associated with a sequence of trivial multi-instructions. Equivalently, a deadlock is a point p such that any directed path which starts at p and triggers some non trivial multi-instruction actually meets a forbidden point. The latter formulation exactly states that the forbidden region hinders the expected functioning of the program.

The following result is one of the main motivations for introducing conservative processes and discrete models. From a practical point of view, it states that for conservative programs, admissibility no longer need to be checked at runtime because it is statically encoded in the discrete model.

Theorem 3.4.11. A directed path which does not meet any forbidden point is admissible. Conversely, for each admissible path which meets a forbidden point there exists a directed path which avoids them and such that both paths induce the same sequence of multi-instructions up to empty multi-instructions.

Proof. Let $\overline{p} = p(0)\cdots p(K)$ be a path that is not admissible and let k be such that the multi-instruction μ_k defined in Definition 3.3.7 is not admissible at state $\sigma \cdot \mu_0 \cdots \mu_{k-1}$. If μ_k contains a conflict then so does $\lambda(p(k))$, the latter being an extension of the former (*cf.* Definition 3.3.7). By definition of μ_k we have for all barriers $b \in \mathcal{B}$

$$\{n \mid \mu_k(n) = W(b)\} \quad = \quad \{n \mid \lambda(p(k))(n) = W(b)\}$$

hence if μ_k forces a barrier (*i.e.* the left hand set above is not empty and its cardinal is less or equal than arity(b)) then the point $p(k)$ is forbidden in the sense of Definition 3.4.10. Since the program is conservative, the following holds for all semaphores $s \in \mathcal{S}$

$$\sigma \cdot \mu_0 \cdots \mu_k \quad = \quad F(p(k), s)$$

so if the left hand term of the preceding equality is strictly greater that arity(s), then the point $p(k)$ is forbidden. Conversely suppose that the path $\overline{p}$ is admissible and that $p(k)$ is forbidden, also suppose that k is minimum. By similar arguments to those exposed above, we conclude that there exist $n, m \in \{1, \ldots, N\}$ such that $\lambda_n(p_n(k))$ and $\lambda_m(p_m(k))$ conflict in the sense of Definition 3.4.10. Because $\overline{p}$ is admissible, there must be some $k' < k$ such that the following holds for all $k''' \in \{k', \ldots, k\}$.

$$\lambda_m(p_m(k''')) \quad = \quad \lambda_m(p_m(k))$$

In less formal words, the m^{th} process is stalled on an assignment in conflict with the instruction $\lambda_n(\gamma_n(k))$. The interpretation of arrows as interlude between instructions plays a role here. Denote by k'' the first index such that the m^{th} coordinate of $p(k'')$ is the unique arrow α outgoing from $p_m(k')$. If no such index exists, then $k'' = \infty$. To obtain the expected directed path, it suffices to change the m^{th} coordinate of points $p(k''')$ into α for all $k' < k''' < k''$, which amounts to set the instruction pointer of the m^{th} process in an intermediate position. One readily deduces from Definition 2.1.7 that the altered directed path induces the same sequence of multi-instructions than the original one. Of course there might be another $m' \in \{1, \ldots, N\}$ such that $\lambda_n(p_n(k))$ and $\lambda_{m'}(p_{m'}(k))$ conflict, but then it suffices to iterate the preceding construction until all such indices have been treated. Finally we obtain a directed path $\overline{p'}$ whose initial segment $p'(0), \ldots, p'(k)$ does not meet any forbidden point and induces the same sequence of multi-instructions than $p(0), \ldots, p(k)$. We conclude by a straightforward induction over k. □

Remark 3.4.12. There might be admissible paths that meet forbidden points. However, by Theorem 3.4.11, it is always possible to "replace" such a path by an "equivalent" one which avoids forbidden points as illustrated by Figure 3.13. Therefore restricting the class of admissible paths to those which do not meet any forbidden point does not result

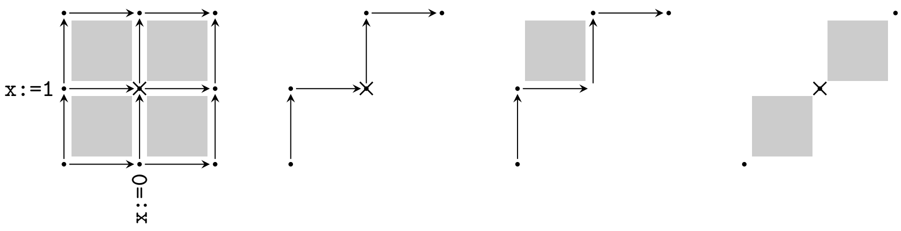


Figure 3.13: A discrete model, an admissible path on it that meets a forbidden point, a possible replacement, and a nonadmissible path.

in a significant loss. Moreover it allows to turn the instructions $P(\_)$, $V(\_)$, and $W(\_)$ into directives, in other words to treat them statically rather than dynamically. Indeed they mould the discrete model of the program at compile time after which they do not interfer on the runtime execution of the program because the language does not offer any feature to access the amount of occurrences held at a given moment.

Remark 3.4.13. Theorem 3.4.11 demands to sacrifice pointer arithmetics (which should be seen as a good thing from the static analysis point of view) as well as to have a notion of conflicting instructions that can be decided statically. For example, according to Definition 1.3.9, every assignment conflicts with itself. Our credo is to statically treat concurrency as far as possible, by encoding the constraints imposed on executions in the models of the programs.

We provide a technical result that will be used in the proof of Theorem 7.2.4. Indeed, writing a multi-instruction as a union of smaller ones can be seen as a discrete version of the notion of weakly dihomotopic directed paths which will be introduced in Definition 7.2.1. Corollary 3.4.14 is illustrated on Figure 3.14 where every horizontal axis is to be understood as the "timeline" of a running process of a program. Each "dot" is the time at which some instruction is executed, hence vertically aligned dots represent a multi-instruction.

Corollary 3.4.14. Let $\overline{p}$ and $\overline{p}'$ be two paths (*cf.* Definition 3.3.2) that do not meet any forbidden points (*cf.* Definition 3.4.10). Suppose that the sequence of multi-instructions associated with $\overline{p}$ is $\mu_1,\dots,\mu_K$ while the one associated with $\overline{p}'$ is the concatenation of $S_1,\dots,S_K$ where each S_k is a sequence of multi-instructions whose union is equal to μ_k. Let σ be the initial state of the program. For all initial segments $\mu'_1,\dots,\mu'_{K'}$ of the concatenation $S_1\cdots S_K$ there exists $k \leqslant K$ and K'' such that $\mu'_{K''},\dots,\mu'_{K'}$ is a strict initial segment of S_{k+1} and $\mu'_1,\dots,\mu'_{K'} = S_1\cdots S_k\cdot(\mu'_{K''},\dots,\mu'_{K'})$. Let σ be the initial context of the program. Then for all expressions ε met along the path $p'_{K''-1}\cdots p'_{K'}$ (*cf.* Remark 3.3.8) the evaluations of ε (*cf.* Definition 1.3.3) in the contexts $\sigma\cdot\mu'_1\cdots\mu'_{K'}$ and $\sigma\cdot\mu_1\cdots\mu_k$ are the same:

$$[\![\varepsilon]\!]_{\sigma\cdot\mu'_1\cdots\mu'_{K'}} \quad = \quad [\![\varepsilon]\!]_{\sigma\cdot\mu_1\cdots\mu_k}$$

In particular p is an execution trace iff so is $\overline{p}'$.

Proof. By Theorem 3.4.11 both $\overline{p}$ and $\overline{p}'$ are admissible (*cf.* Definition 3.3.6). Because $p'(K''-1) = p(k)$ a free variable occurring in some expression ε met along the path

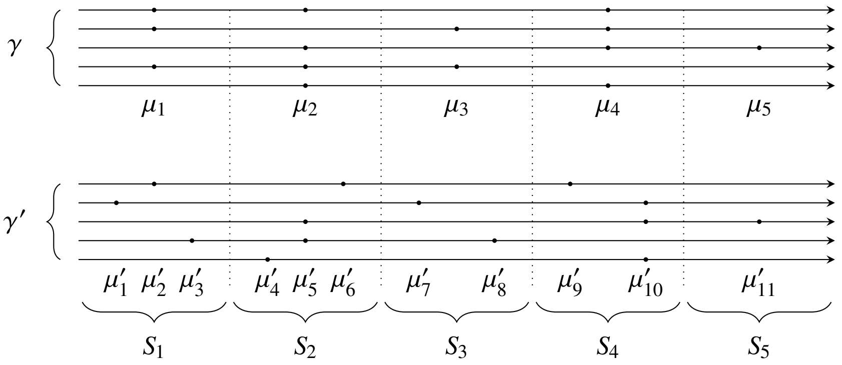


Figure 3.14: Timelines interpreting a sequence of multi-instructions.

$p'_{K''-1} \cdots p'_{K'}$ cannot be altered by μ_{k+1}. Since $\mu'_{K''} \sqcup \ldots \sqcup \mu'_{K'} \subseteq \mu_{k+1}$ we get the following equality

$$[\![\varepsilon]\!]_{\sigma\cdot\mu'_1\cdots\mu'_{K'}} \quad = \quad [\![\varepsilon]\!]_{\sigma\cdot\mu'_1\cdots\mu'_{K''-1}}$$

and we conclude by remarking that $\sigma \cdot \mu'_1 \cdots \mu'_{K''-1} = \sigma \cdot \mu_1 \cdots \mu_k$. □

4

Models of Directed Topology

The distinctive feature of true concurrency is the lack of a global clock – Dijkstra (1968). In particular any action performed by some agent may stop or start during the execution of an action performed by another agent. Such a framework is said to be **asynchronous**. Combinatorial models do not fulfill this requirement because their very nature imposes a discrete notion of time. Yet one can still argue that physical time is discrete (*cf.* Planck time) and anyway, computers are designed to guarantee atomicity. Nevertheless, combinatorial models remain unable to render the speed ratio between concurrent agents, unless one accepts a huge number of states. A natural idea to circumvent the problem is to equip the collection of states of the system with a topology. Doing so, one has to keep in mind the causality that binds the states of the system. This constraint is the consequence of a simple fact that is common to every programming language: reading any source code, the instruction pointer is always supposed to move from top down to bottom, sometimes jumping back when a loop is met. Therefore the (topological) space of states should be provided with an extra structure expressing the causality. Then, while the set of execution traces of a sequential program is overapproximated by the set of paths on its control flow graph, the set of execution traces of a parallel program is overapproximated by the set of continuous paths (on the space of states) that respect causality. What “overapproximation” exactly means will be thoroughly explained in Section 7.1. Basically, every continuous path that respects causality (such a path will be said to be directed) is associated with a sequence of instructions that might not be induced by an execution trace. But conversely every execution trace is obtained that way. The collection of continuous path which respects causality should thus be compared to the collection of paths on the control flow graph of a sequential program. In particular, models should have directed loops (*i.e.* a formalization of the causality should allow nonconstant paths that stop where they begin). This chapter explores and compares several such formalisms. The categorical properties, and more specifically completeness and cocompleteness, are systematically studied as they allow one to define the directed counterparts of the geometric realizations of cubical sets and precubical sets – see Section 2.4.

In Section 4.2, we attempt to axiomatize the minimal requirements for a category to be a framework for directed topology. In Sections 4.1 and 4.3, we describe the formalization of directed topology that came in the first place: locally ordered spaces extend pospaces as smooth manifolds extend $\mathbb{R}^n$. Their category behaves poorly with respect to abstract constructions algebraic topologists are accustomed to. However, it is rich enough to provide any reasonable program with a continuous model. Moreover, the notion of locally ordered space being fairly rigid, it enjoys somewhat nice properties. The categorical drawbacks of locally ordered spaces were addressed by Sanjeevi Krishnan introducing *streams*. Indeed he observed that cosheaves behave better than sheaves when preorders are at stake. Streams are discussed in Section 4.4 then we focus on d-spaces in Section 4.5, paying special attention to their relation to other frameworks. Each of them indeed comes with a canonical functor to the category of d-spaces. In the case of streams, it is actually an adjunction. Many examples are provided, some of them coming from vector fields.

Before embarking for the realm of directed topology, we remind the reader about some points of topology and category theory.

Definition 4.0.1. A **Moore path** (or just **path**) on a topological space X is a continuous map $\delta : [0,r] \to X$ with $r \in \mathbb{R}_+$. The parameter r is called the **shape** of the path while its source $\partial^- \delta$ and its target $\partial^+ \delta$ are defined as $\delta(0)$ and $\delta(r)$. A **subpath** of shape s of a path δ of shape r is a path of the form $\delta \circ \theta$ where $\theta : [0,s] \to [0,r]$ is nondecreasing continuous. The map θ needs not to be one-to-one nor onto. Given Moore paths δ and γ of shapes r and s such that $\partial^- \gamma = \partial^+ \delta$, we define the **concatenation** $\gamma * \delta : [0, r+s] \to X$ by

$$\gamma * \delta(t) = \begin{cases} \delta(t) & \text{if } t \in [0,r] \\ \gamma(t-r) & \text{if } t \in [r, r+s] \end{cases}$$

The concatenation is obviously associative so we define $\mathcal{M}(X)$, the **(Moore) path category** of X. Its objects are the points of X while its identities are the paths of null shape. The construction is functorial $\mathcal{M} : \mathbf{Top} \to \mathbf{Cat}$, it is the prototype of a category of dipaths (*cf.* Definition 4.2.18).

Definition 4.0.2. As a matter of notation, the left adjoint R to an inclusion functor $I : \mathcal{A} \hookrightarrow \mathcal{B}$ is called the **reflection** of $\mathcal{B}$ in $\mathcal{A}$ when it also satisfies $R \circ I = \mathrm{id}_{\mathcal{A}}$. In particular we write that $\mathcal{A}$ is a **reflective** subcategory of $\mathcal{B}$ – see (Borceux, 1994a, p.118, Section 3.5). By extension, we define the reflect in $\mathcal{A}$ of an object or a morphism of $\mathcal{B}$.

The notion of reflection offers a standard method to prove that a category is complete or cocomplete. Indeed we have

Lemma 4.0.3. Any reflective subcategory of a complete (resp. cocomplete) category is complete (resp. cocomplete).

Proof. See (Borceux, 1994a, p.118-119, Proposition 3.5.3 and Proposition 3.5.4). □

The notion of **coreflection** is defined the same way from the right adjoint. By duality, Lemma 4.0.3 holds for coreflective subcategories.

Definition 4.0.4. An **embedding** is a faithful functor that is also injective on objects.

4.1 Partially Ordered Spaces

The first idea one may have consists of providing topological spaces with partial orders. We will see however that this notion is too restrictive because it forbids spaces with directed loops (*i.e.* nonconstant paths whose starting point and endpoint are equal). It turns out that this notion was first investigated in the context of functional analysis. Leopoldo Nachbin (1948a,c,b) has dedicated a series of papers to their study and gathered his results in a document that remains, as far as I know, the only book entirely dedicated to pospaces – Nachbin (1965).

Definition 4.1.1. A **partially ordered space** (or **pospace**) is a topological space X together with a partial order $\sqsubseteq$ on (the underlying set of) X whose graph

$$\{(a, b) \in X \times X \mid a \sqsubseteq b\}$$

is a closed subset of $X \times X$. A pospace morphism is an order-preserving continuous map. Pospaces and their morphisms form the category **PoTop**.

Remark 4.1.2. The underlying space of a pospace is Hausdorff since its diagonal is closed as the intersection of the graph of the partial order and the graph of its opposite – (Nachbin, 1965, p.27, Proposition 2).

Example 4.1.3.

Any poset with the discrete topology. In particular the forgetful functor **PoTop** → **PoSet** has a left adjoint.

Any Hausdorff space with the discrete order. In particular the forgetful functor **PoTop** → **Haus** has a left adjoint.

The directed real line (*i.e.* $\mathbb{R}$ with its standard topology and order).

The sub-pospaces of a pospace (*i.e.* its subsets with the induced topology and order).

The **directed intervals** (*i.e.* the connected sub-pospaces of the directed line) and especially the directed compact unit segment $[0, 1]$, denoted by $\vec{I}$, which will be of a great importance in the sequel.

A more intricate example is given by the nonempty closed subspaces of a metric space, ordered by inclusion and equipped with the topology induced by the Hausdorff distance – (Aliprantis and Border, 2006, pp.109-113) or (Beer, 1993, p.85) or (Rockafellar and Wets, 2004, p.117).

$$d(K_1, K_2) = \max\{d(x_1, K_2), d(x_2, K_1) \mid x_1 \in K_1; x_2 \in K_2\}$$

$$\text{with} \quad d(x, K) = \min\{d(x, k) \mid k \in K\}$$

Definition 4.1.4. A **dipath** on a pospace X is a pospace morphism from some nonempty directed compact interval $[0, r]$ to X where $r \in \mathbb{R}_+$ (r is called the domain of γ). If the domain of a dipath γ is r then its source and its target are respectively $\gamma(0)$ and $\gamma(r)$. We should take care that γ is in particular a pospace morphism and as such, its source and its target also refer to its domain and its codomain. To avoid confusion we sometimes say that $\gamma(0)$ and $\gamma(r)$ are the starting point and the endpoint of γ. A pospace X is said to be **directed by the dipaths** when one has $x \sqsubseteq_X y$ iff there exists a dipath on X from x to y. These pospaces form the full subcategory $\mathbf{PoTop}_\mathrm{d} \subseteq$ **PoTop**.

Example 4.1.5. The set of rational numbers $\mathbb{Q}$ inherits a pospace structure from the directed real line. However the resulting pospace is not directed by dipaths though $\mathbb{R}$ is so.

Theorem 4.1.6. The image of a non-constant dipath is isomorphic to $\vec{I}$.

Proof. The proof heavily relies on the tight relation between the standard topology of $[0, 1]$ and its total order, and more precisely on the following two facts. A topological space that is not reduced to a singleton is said to be **separable** when it contains a countable dense subset. A **continuum** is a compact connected Hausdorff space – Nadler Jr. (1992). A point x of a connected space X is said to be **nonseparating** when

$X - \{x\}$ is still connected. An **arc** is a continuum with exactly two non-separating points. The first fact claims that any separable arc is homeomorphic with $[0, 1]$. A pospace is said to be **linear** when its underlying order is so. The second fact claims that there are exactly two linear pospaces whose underlying space are a given arc, each of them being actually isomorphic to the opposite of the other. See (Nadler Jr., 1992, Th.6.17 p.96), and the relation to Suslin's Problem – (Jech, 2002, pp.38-39), for further details.

Admitting these assertions the proof becomes easy. Let δ be a pospace morphism from $[0, r]$ to X. Then $\mathrm{img}(\delta)$ inherits a pospace structure. Its underlying space is a continuum as the direct image of a continuum in a Hausdorff space. It is clearly separable considering the direct image of $\mathbb{Q} \cap [0, r]$ by δ. According to the first claim the underlying space of $\mathrm{img}(\delta)$ is homeomorphic to $[0, 1]$. Furthermore the order inherited by $\mathrm{img}(\delta)$ from X is linear because so is $[0, r]$. Then according to the second claim the pospace $\mathrm{img}(\delta)$ is isomorphic to $[0, 1]$. □

Theorem 4.1.6 has no obvious counterpart in general topology. A Peano curve indeed provides a continuous map from $[0, 1]$ onto $[0, 1] \times [0, 1]$. In fact the study of continuous images of the compact unit segment is a mathematical research subject on its own – Nikiel et al. (1993) and Nadler Jr. (1992).

Remark 4.1.7. Another consequence of Theorem 4.1.6 is that any pospace morphism $\gamma : \vec{I} \to X$ such that $\gamma(0) = \gamma(1)$ is constant. Therefore a pospace has no directed loop.

Before leaving pospaces we observe some of their categorical properties. We will prove that **PoTop** is cocomplete though certain coequalizers may not be preserved by the forgetful functor. In relation to Chapter 2 we will see that the underlying space of a realization in **PoTop** may not be the realization (in **Top**) of the underlying spaces.

Lemma 4.1.8. The category **PoTop** is complete.

Proof. Suppose that we are given a nonempty family of pospaces. The underlying space of the product is the product in **Top** of the underlying spaces of the elements of the family. The graph of the product order is the product set of their graphs. Since each of them is closed, so is the graph of the product order. If we are given two morphisms of pospaces with the same source and the same target, their equalizer is the partially ordered subspace of their common source on which they agree. □

Example 4.1.9. For $n \in \mathbb{N}$, the space $\mathbb{R}^n$ with the product order is a pospace.

Cocompleteness of **PoTop** is obtained by proving that it is a reflective subcategory of a cocomplete one.

Definition 4.1.10. A **preordered space** is a Hausdorff space X equipped with a closed preorder $\preccurlyeq$. A morphism of preordered spaces is a continuous map that preserves preorders. The category of preordered spaces is denoted by **PreTop**.

Remark 4.1.11. The forgetful functor U : **PreTop** $\to$ **Haus** has left and right adjoints. They are respectively given by the discrete preorder (*i.e.* $x \preccurlyeq y$ when $x = y$) and the chaotic preoreder (*i.e.* $x \preccurlyeq y$ for all x and y).

Lemma 4.1.12. The category **PreTop** is complete and cocomplete and the inclusion functor **PoTop** $\hookrightarrow$ **PreTop** preserves limits.

Proof. The construction of products and equalizers is the same as in the proof of Lemma 4.1.8 hence we have the completeness and the limits preservation. By Remark 4.1.11 the forgetful functor preserves colimits, so their underlying spaces are taken in **Haus**. It remains to describe the preorders. A coproduct of a family of closed preorders is clearly a closed preorder. The case of coequalizers is a bit more subtle. Suppose that we are given $f, g \in \textbf{PreTop}(X, Y)$. Therefore we have the coequalizer diagram in **Haus**

$$\begin{tikzcd} UX \arrow[r, shift left, "Uf"] \arrow[r, shift right, "Ug"'] & UY \arrow[r, "q"] & Z \end{tikzcd}$$

The preorder of the coequalizer in **PreTop** is the least closed preorder $\preccurlyeq_Z$ on Z such for all $y, y' \in Y$, $y \preccurlyeq_Y y' \quad \Rightarrow \quad q(y) \preccurlyeq_Z q(y')$. The latter argument would not have been valid for pospaces since the family of closed partial orders on Z satisfying the earlier property may be empty while the chaotic preorder matches the requirements. □

The cocompleteness of **PoTop** is a consequence of Lemma 4.1.12 and

Proposition 4.1.13. The full embedding **PoTop** $\hookrightarrow$ **PreTop** has a left adjoint.

Proof. We mimic the proof of (Mac Lane, 1998, p.135, Proposition 2). Following Lemma 4.1.8 and Lemma 4.1.12 we know that **PoTop** is complete and that the inclusion functor preserves limits. Given a morphism $f \in \textbf{PreTop}(X, Y)$ whose codomain is actually a pospace, we have the factorization

$$\begin{tikzcd} X \arrow[r, "q"] \arrow[rr, bend left, "f"] & \mathrm{img}(f) \arrow[r, hook] & Y \end{tikzcd}$$

with $\mathrm{img}(f)$ being a subobject of the pospace Y, therefore a pospace, and q being onto. The collection of quotients of X is thus a set so the solution set condition is satisfied and we apply the Freyd adjoint functor theorem (Mac Lane, 1998, p.121, Theorem 2) to conclude. □

Corollary 4.1.14. The category **PoTop** is cocomplete.

Proof. By Lemma 4.0.3. □

Example 4.1.15. The **directed open star** St_a^b, with $a, b \in \mathbb{N}$, is the colimit in **PoTop** of a copies of $\mathbb{R}_-$ and b copies of $\mathbb{R}_+$ over 0, with $\mathbb{R}_-$ and $\mathbb{R}_+$ being understood with the pospace structure inherited from the directed real line – see Figure 4.1. In particular the pospaces $\{0\}$, $\mathbb{R}_-$, $\mathbb{R}_+$, and $\mathbb{R}$ are isomorphic to St_0^0, St_1^0, St_0^1, and St_1^1. The colimit that defines a directed open star is preserved by the forgetful functor $U : \textbf{PoTop} \to \textbf{Haus}$. Another equivalent definition of the directed open stars will be given in Section 6.1 where we will see that they can be indifferently considered as locally ordered spaces – see Section 4.3, as streams – see Section 4.4, or as d-spaces – see Section 4.5.

Remark 4.1.16. The proof of Proposition 4.1.13 is a bit confusing as the Freyd adjoint theorem acts like a magic wand. We give a more intuitive approach. Let $(X_0, \preccurlyeq_0)$ be a preordered set. Then consider the quotient in **Haus** of X_0 by the least equivalence relation that identifies x and y when $x \preccurlyeq_0 y$ and $y \preccurlyeq_0 x$. Denote the Hausdorff quotient space as X_1 and the quotient map as $q_0 : X_0 \to X_1$. Then let $\preccurlyeq_1$ be the least closed

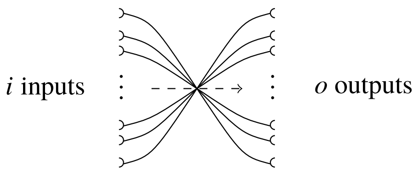


Figure 4.1: Directed open stars

preorder on X_1 so that q_0 induces a morphism of **PreTop**. The preorder $\preccurlyeq_1$ may not be antisymmetric yet we can iterate the construction

$$(X_0, \preccurlyeq_0) \xrightarrow{q_0} \cdots \xrightarrow{q_{n-1}} (X_n, \preccurlyeq_n) \xrightarrow{q_n} \cdots$$

If λ is a limit ordinal, then X_λ is the colimit in **Haus** of the preceding diagram, equipped with $\preccurlyeq_\lambda$ the least closed preorder on X_λ so that all the mapping of the colimit cone induce morphisms of **PreTop**. What really matters is that each step of the transfinite induction identifies some points of X_0. It follows that the induction stops, or more precisely there is an ordinal α beyond which the transfinite sequence is constant. Then $(X_\alpha, \preccurlyeq_\alpha)$ is the reflect of $(X_0, \preccurlyeq_0)$ in **PoTop**.

Remark 4.1.17. If the underlying space of a pospace X is exponentiable, then so is X. Given a pospace Y, the exponential space Y^X is the set **PoTop**(X, Y) equipped with the pointwise order and the compact-open topology – see the tenth point of Proposition 2.3.3.

Remark 4.1.18. Let us alter Definition 4.1.1 and require that the underlying space be compactly generated. By Remark 4.1.2 it is actually an object of **CGH**. The right adjoint k : **Haus** → **CGH** extends to a right adjoint k : **PoTop** → **PoCGH**, indeed a closed partial order on a space X is still closed in kX since the latter topology is finer than the former. Hence **PoCGH** is a coreflective subcategory of **PoTop**, so it is complete and cocomplete by the dual of Lemma 4.0.3. In particular **PoCGH** is Cartesian closed. If X and Y are two objects of **PoCGH**, then Y^X is the set **PoCGH**(X, Y) seen as a subspace of UY^{UX} (in **CGH**). Then $U(Y^X)$ is a closed subspace of UY^{UX}. Indeed a continuous map $f : X \to Y$ that is not a morphism of **PoCGH** does not preserve the orders. So there exist $x, x' \in X$ such that $x \sqsubseteq_X x'$ but $f(x) \not\sqsubseteq_Y f(x')$. Since Y is a pospace, we have an open neighborhood A of $f(x)$ and an open neighborhood B of $f(x')$ such that $a \not\sqsubseteq_Y b$ for all $a \in A$ and all $b \in B$. Then

$$\big\{ f \in \mathbf{CGH}(X, Y) \,\big|\, f(x) \in A ; f(x') \in B \big\}$$

is an open neighborhood of f that does not meet $U(Y^X)$. We conclude that $U(Y^X)$ is compactly generated by (Engelking, 1989, p.153, Theorem 3.3.25).

There is an obvious embedding of □ in **PoTop** so we can define the cubical and the precubical realizations in **PoTop**. Let us compute a colimit which turns out to be the realization of the graph with a single vertex and a single arrow (*i.e.* the 1-dimensional precubical set K defined by $K_0 = \{0\}$, $K_1 = \{1\}$, and $K_n = \emptyset$ for $n \geqslant 2$).

Example 4.1.19. Consider the following pair of parallel morphisms being intentionally vague about the order carried by the unit segment.

$$\begin{tikzcd} \{*\} \arrow[r, shift left, "0"] \arrow[r, shift right, "1"'] & {[0,1]} \end{tikzcd}$$

First suppose that it is equipped with the standard order and consider a pospace morphism f such that $f(0) = f(1)$. If $0 \leqslant t \leqslant 1$ then $f(0) \sqsubseteq_X f(t) \sqsubseteq_X f(1)$, therefore $f(t) = f(0)$ because $\sqsubseteq_X$ is antisymmetric. So the coequalizer is the singleton. Hence the forgetful functor does not preserve colimits since the coequalizer in **Top** is $\mathbb{S}^1$ as the intuition suggests. Therefore it has no right adjoint – (Borceux, 1994a, Prop.3.2.2 p.106). Now suppose that $[0,1]$ is equipped with the discrete order (*i.e.* $x \sqsubseteq y$ when $x = y$). Then for all pospaces X we have $\mathbf{PoTop}([0,1], X) = \mathbf{Top}([0,1], X)$. Hence the coequalizer is just the circle $\mathbb{S}^1$ equipped with the discrete order.

From Example 4.1.19 we conclude that pospaces are too rigid because they do not allow directed loops. In fact the forgetful functor to **Set** does not even preserve the simplest colimits. The remaining of this chapter is dedicated to exploring more supple formalizations of directed topology. We will need the following definition.

Definition 4.1.20. The full subcategory of **PoTop** generated by the n-fold products of directed intervals (*cf.* Example 4.1.3 and Lemma 4.1.8) is denoted by **Cub**.

Remark 4.1.21. The category **Cub** contains both $\emptyset$ and all the singletons $\{x\}$ with $x \in \mathbb{R}$. In addition it has all the finite products and they are preserved by the inclusion functor $\mathbf{Cub} \hookrightarrow \mathbf{PoTop}$. The category **Cub** also contains all the pushout squares of the form

$$\begin{tikzcd}
B \arrow[r, "\beta'"] & D \\
A \arrow[u, "\beta"] \arrow[r, "\gamma"'] & C \arrow[u, "\gamma'"']
\end{tikzcd}$$

with A, B, C, being directed intervals and $\mathrm{img}(\beta)$ (resp. $\mathrm{img}(\gamma)$) being a final (resp. initial) segment of B (resp. C). Such a pushout is called a **pasting of intervals**. In that case D is also a directed interval and B (resp. C) is mapped to an initial (resp. final) segment of D. Every finite product of directed intervals is exponentiable since all the directed intervals are exponentiable in **PoTop** – see Remark 4.1.17. As left adjoints, the endofunctors of the form

$$X \in \mathbf{Cub} \mapsto I_1 \times \cdots \times I_k \times X \times I_{k+1} \times \cdots \times I_n \in \mathbf{Cub}$$

with $(I_1, \ldots, I_n)$ being some n-tuple of directed intervals, preserve colimits. The image of a pasting of intervals by one of the previous endofunctors is called a **pasting of cubes**.

4.2 Framework for Directed Topology

The content of this section is mainly extracted from (Haucourt (2012)), yet the exposition has been greatly simplified. Its purpose is to provide an abstraction that encompasses all the most widely spread formalizations of directed topology, and to build the fundamental category functor from it – see Definition 5.2.10. In particular, a generic version of the van Kampen theorem holds in this context – see Theorem 5.4.1. The first step of the construction is the dipath category functor, it is done in this section – see Definition 4.2.18. The abstraction also provides enough features for a reasonable cubical nerve functor to exist (see Definition 2.2.3 and Theorem 2.2.4). The objects of the categories we have in mind are topological spaces endowed with an extra structure that

$$\begin{tikzcd}
 & & \mathbf{PoTop} \arrow[rrd, bend left, "\text{forgetful}"] & & \\
\mathbf{Cub} \arrow[r, tail, "I"] \arrow[rru, hook, bend left] & \mathcal{C} \arrow[rr, bend left, "U"] \arrow[rr, phantom, "\top"] & & \mathcal{K} \arrow[ll, bend left, "F"] \arrow[r, hook] & \mathbf{Top}
\end{tikzcd}$$

Figure 4.2: Framework for directed topology

encodes causality. Of course, their morphisms are the structure preserving continuous maps. Let $\mathcal{K}$ be a cocomplete subcategory of **Top** (*e.g.* **Top**, **CG**, **CGWH**, but preferably **Haus** or **CGH**).

Remark 4.2.1. Any object X of $\mathcal{K}$ is, in particular, a topological space. From now on, and till the end of the section, a $\mathcal{K}$-subspace of X is a topological subspace of X that actually belongs to $\mathcal{K}$. If $\mathcal{K} = \mathbf{Haus}$ then any subspace is a $\mathcal{K}$-subspace since any subspace of a Hausdorff space is again a Hausdorff space. Things are more intricate when $\mathcal{K} = \mathbf{CGH}$ since a subspace of a compactly generated space may not be compactly generated.

Definition 4.2.2. A **framework for directed topology** over $\mathcal{K}$ is given by the diagram on Figure 4.2 with the following properties.

1. The embedding I strictly preserves both the finite products and the pastings of cubes (*cf.* Remark 4.1.21).

2. For all objects (resp. morphisms) $X \in \mathbf{Cub}$, the underlying spaces (resp. maps) of X and IX are the same (*i.e.* all the paths from **Cub** to **Top** on the diagram of Figure 4.2 induce the same functor).

3. The functor U is faithful, has a left adjoint F, and $U \circ F = \mathrm{id}_{\mathcal{K}}$. Moreover U induces a bijection between the constant morphisms of $\mathcal{C}(X, Y)$ and the ones of $\mathcal{K}(UX, UY)$ for all objects X and Y.

4. The functor U reflects $\mathcal{K}$-subspaces: for all objects X of $\mathcal{C}$ and all $\mathcal{K}$-subspaces $A \subseteq UX$, the full subcategory of $\mathcal{C}/X$ of objects α that satisfy $\mathrm{img}(U(\alpha)) \subseteq A$ admits a terminal object $A \rightarrowtail X$. Its source is, by definition, the $\mathcal{C}$**-subspace** A of X (note that we use the same notation for A and its underlying space).

5. For all objects $X \in \mathcal{C}$ the subcategory $U^{-1}(U(X)) \subseteq \mathcal{C}$ is small.

If the embedding I is full then the framework is said to be **directed**. By analogy with homeomorphisms, which are isomorphisms of topological spaces, an isomorphism of $\mathcal{C}$ is also called a **dihomeomorphism**.

Remark 4.2.3. Definition 4.2.2 needs some taking apart. The first two axioms claim that $\mathcal{C}$ contains a copy of **Cub** that respects the constructions one wishes to export in $\mathcal{C}$. In particular the pastings of cubes in **Cub**, which are really pushout diagrams that compose in the strict sense, are sent to pushout diagrams in $\mathcal{C}$ which thus also compose in the strict sense. This property, which is trivially satisfied in practice, simplifies the definition of the concatenation of dipaths and as well as that of dihomotopies. Moreover, we have an embedding of $\square$ into $\mathcal{C}$ from which one can define the nerve functor (*cf.* Definition 2.2.3).

Remark 4.2.4. According to the embedding I, we write X instead of $I(X)$ for all objects X of **Cub**, and for all $r \in \mathbb{R}_+$, the notation I_r stands for $I([0,r])$. In particular we have $I_0 = I(\{0\})$.

The last three axioms are a bit subtler. They formalize the idea that any object of C is a topological space endowed with an extra structure that is preserved by the morphisms. With respect to this point of view, the meaning of the third axiom is clear. In particular it claims that any constant map actually induces a morphism of C. However it does not ensure that the notion of subspace in C behaves as one would expect. Intuitively we would like that all the $\mathcal{K}$-subspaces of UX, with X being an object of C, inherit a C-object structure from the one of X. Without further hypothesis, we might have the following unpleasant situation: $A \subseteq B \subseteq UX$ is a tower of $\mathcal{K}$-subspaces, A' and B' are the C-object structures induced by X on A and B, while A'' is the C-object structure induced on A by B', and $A' \ncong A''$. This situation is illustrated by Example 4.2.5.

Example 4.2.5. To highlight the meaning of the fourth axiom, we describe a situation in which all the five axioms but only a weak version of the fourth one are satisfied. Then we check that the resulting notion of subspace is ill-behaved. Suppose that we have associated each set S with a partition ϕS. Let C be the category of pairs (X,P) with $X \in$ **Top** and P being any partition on S, the underlying set of X, that is finer than ϕS. The morphisms of C from (X,P) to (X',P') are the continuous maps such that for all $p \in P$ there exists $p' \in P'$ such that $f(p) \subseteq p'$. The partition of X into singletons is therefore always allowed so the forgetful functor $U : C \to$ **Top** admits a left adjoint F. The third axiom is satisfied and $F \circ (\mathbf{Cub} \hookrightarrow \mathbf{Top})$ provides an embedding of **Cub** in C that satisfies the first two axioms. The fifth axiom is satisfied since the collection of partitions over a set is a set. Hence all the axioms but the fourth one are satisfied without making any assumption on the choice function ϕ. Suppose that $\phi\mathbb{R} = \{\mathbb{R}\}$, $\phi[-2,2] = \{[-2,0[,[0,2]\}$, and $\phi[-1,1] = \{[-1,1]\}$. Denote by C the object $(\mathbb{R},\phi\mathbb{R})$, and by B the subspace of C whose underlying space is $[-2,2]$. The partition associated with B is the coarsest one P such that

- P is finer that $\phi[-2,2]$ and
- the inclusion $[-2,2] \hookrightarrow \mathbb{R}$ induces a morphism of C (*i.e.* P is finer that $\{[-2,2]\}$),

in other words it is precisely the partition $\phi[-2,2]$. Now let A and A' be the subspaces of B and C whose underlying spaces are $[-1,1]$. Their associated partitions are respectively $\{[-1,0[,[0,1]\}$ and $\phi[-1,1]$, which proves that A is a subspace of B, B is a subspace of C, but $A \ncong A'$ therefore A is not a subspace of C. Intuitively the C-object structure induced by $X \in C$ on $A \subseteq UX$ might be changed by any alteration of X occurring outside of A. Our counter-example is not very convincing for the choice function ϕ has been deliberately settled to thwart the axiom, we will see in Section 8.6 that **PoTop** supplies a natural counter-example.

The fourth axiom prevents this from happening. In the statement of Lemma 4.2.6 and Lemma 4.2.7 it is assumed that $U : C \to \mathcal{K}$ is a functor that satisfy the third and the fourth axioms.

Lemma 4.2.6. With the previously introduced notation we have $A' \cong A''$.

Proof. Let $\alpha : A' \rightarrowtail X$, $\beta : B' \rightarrowtail X$, and $\alpha' : A'' \rightarrowtail B'$ be the terminal objects given by the fourth axiom. We have a unique morphism $\gamma : A' \to B'$ such that $\beta \circ \gamma = \alpha$. In particular we have $\mathrm{img}(U(\gamma)) = A \subseteq B$ hence there exists a unique $\gamma' : A' \to A''$

such that $\gamma = \alpha' \circ \gamma'$. We also have $\mathrm{img}(U(\beta \circ \alpha')) = A$ hence there exists a unique $\gamma'' : A'' \to A'$ such that $\alpha \circ \gamma'' = \beta \circ \alpha'$. It follows that γ' and γ'' are inverses of each other. □

Lemma 4.2.7. Given a morphism $f \in \mathcal{C}(X,Y)$, if $\mathrm{img}(f)$ belongs to $\mathcal{K}$, then f factorizes in a unique way through $\mathrm{img}(f)$ (*i.e.* the subspace of Y over $\mathrm{img}(Uf)$) as $i_f \circ q_f$, with $U(i_f) = \mathrm{img}(Uf) \hookrightarrow Y$ and $U(q_f)$ being onto.

Proof. The statement readily follows from the fourth axiom. □

We actually cast an envious glance at the notion of topological functor – see (Borceux, 1994b, p.366-371, Section 7.3). Such a functor has extremely strong properties: it is faithful, it admits both a left adjoint and a right adjoint, which are both full and faithful, it creates limits and colimits, and actually it is also cotopological! The last three axioms thus force properties that would be satisfied if U were indeed topological. For example, we will see that the functor that forgets the direction of a d-space is topological – see Section 4.5. On the contrary, the forgetful functor U : **PoTop** $\to$ **Top** is not (for it has no right adjoint).

Example 4.2.8. In the case that the embedding I is given by Remark 4.1.21, the frameworks **PoTop** and **PoCGH** are directed. Note that one can also consider the embedding that sends an object X of **Cub** to FUX (*i.e.* the pospace obtained by replacing the partial order of X by the discrete one). The resulting framework is not directed anymore. The frameworks **Top**, **Haus**, and their convenient subcategories, are not directed. So frameworks for directed topology might not be directed!

Now we build the dipath category functor. The singleton $\{0\}$ is a representative of the terminal object of **Cub**, hence (by the first axiom of Definition 4.2.2) I_0 is a representative of the terminal object of $\mathcal{C}$. Let X be an object of $\mathcal{C}$.

Lemma 4.2.9. In any framework for directed topology we have $F(\{0\}) \cong I_0$.

Proof. The terminal object is the product of the empty diagram. Hence the object I_0 is terminal in $\mathcal{C}$ by the first axiom of Definition 4.2.2. From the second axiom and the third axioms we deduce that the underlying spaces of I_0 and $F(\{0\})$ are $\{0\}$. Then $\mathrm{id}_{\{0\}}$ is a constant morphism hence $\mathcal{C}(I_0, F(\{0\}))$ is not empty by the third axiom. Since I_0 is terminal, none of the homsets $\mathcal{C}(X, F(\{0\}))$ is empty, then all the homsets $\mathcal{C}(X, F(\{0\}))$ are singletons because U is faithful and $U \circ F = \mathrm{id}_{\mathcal{K}}$ by the third axiom. □

Definition 4.2.10. The **points** of X are the elements of the set $\mathcal{C}(I_0, X)$.

Remark 4.2.11. By Lemma 4.2.9 we can suppose that $F(\{0\}) = I_0$. Then as a consequence of the adjunction $F \dashv U$, we have the following bijection and Definition 4.2.10 matches the intuition.

$$\mathcal{C}(I_0, X) \cong \mathcal{K}(\{0\}, UX)$$

Given $t \in [0, r]$ be, there is a unique element of **Cub**$(\{0\}, [0, r])$ that sends 0 to t. The image of this morphism by I is still denoted by t.

Definition 4.2.12. The **dipaths** of X are the elements of the set

$$\bigcup_{r \in \mathbb{R}_+} \mathcal{C}(I_r, X)$$

Given $\delta \in \mathcal{C}(I_r, X)$, the **domain** of δ is r, its source (also called starting point) $\partial^{-}\delta$ and its target (also called end point) $\partial^{+}\delta$ are respectively $\delta \circ 0$ and $\delta \circ r$.

$$\begin{tikzcd}
 & X & \\
 & I_{r+s} \arrow[u, "\gamma\cdot\delta"] & \\
I_r \arrow[ur, hook] \arrow[uur, bend left=60, "\delta"] & & I_s \arrow[ul, "+r"'] \arrow[uul, bend right=60, "\gamma"'] \\
 & I_0 \arrow[ul, "+r"] \arrow[ur, hook] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & I_{r+s+t} & & \\
 & I_{r+s} \arrow[ur, hook] & & I_{s+t} \arrow[ul, "+r"'] & \\
I_r \arrow[ur, hook] & & I_s \arrow[ul, "+r"'] \arrow[ur, hook] & & I_t \arrow[ul, "+s"'] \\
 & I_0 \arrow[ul, "+r"] \arrow[ur, hook] & & I_0 \arrow[ul, "+s"] \arrow[ur, hook] &
\end{tikzcd}$$

Figure 4.3: Concatenation of dipaths in a framework

Lemma 4.2.13. Any subpath (*cf.* Definition 4.0.1) of a dipath is a dipath.

Proof. Let δ be a dipath. With the notation of Definition 4.0.1 the mapping θ is a morphism of **Cub** – see (*cf.* Remark 4.1.21). Hence $\gamma \circ I\theta$ is a morphism C that is a dipath. □

Remark 4.2.14. In the case that we consider the directed framework **PoTop**, Definition 4.1.4 and Definition 4.2.12 coincide. On the contrary if we consider the undirected version of **PoTop** (*cf.* Example 4.2.8), then the dipaths in the sense of Definition 4.2.12 are merely the (continuous) paths.

Let δ and γ be dipaths on X of domains r and s such that $\partial^{-}\gamma = \partial^{+}\delta$. Due to the way we have defined the directed paths and the fact that I strictly preserves the pasting of cubes, the outer shape of the left diagram on Figure 4.3 is strictly commutative (*cf.* Remark 4.2.3). From the first axiom of Definition 4.2.2, we have a unique dipath $\gamma \cdot \delta$ on X that makes the left diagram on Figure 4.3 commute. That diagram is thus a pushout square.

Definition 4.2.15. The dipath $\gamma \cdot \delta$ is called the **concatenation** of δ followed by γ.

Lemma 4.2.16. The concatenation of dipaths is associative.

Proof. The three squares of the right diagram on Figure 4.3 are pushout squares. They compose in the strict sense by Remark 4.2.3. □

Remark 4.2.17. With the notation of Definition 4.2.12, we have $\partial^{+}\delta \cdot \delta = \delta = \delta \cdot \partial^{-}\delta$ for all dipaths δ on X – see Figure 4.4.

Definition 4.2.18. Denote by $\mathcal{P}X$ the **category of dipaths** of X. Its objects and morphisms are the points and the dipaths of X (Definition 4.2.10 and Definition 4.2.12), the composition is the concatenation of dipaths (Definition 4.2.15) while the identities are given by Remark 4.2.17. The construction easily extends to the **dipath category functor** $\mathcal{P} : \mathcal{C} \to$ **Cat**, a routine verification indeed proves that for all morphisms $f \in \mathcal{C}(X,Y)$ the post-composition map $f \circ \_$ induces a functor from $\mathcal{P}X$ to $\mathcal{P}Y$.

Given a object X of $\mathcal{C}$, an open subspace of X is a subspace A of X (in the sense of Definition 4.2.2) such that UA is an open subset of UX. An open covering of X is a collection $\mathcal{U}$ of open subspaces of X such that $\{UA \mid A \in \mathcal{U}\}$ covers UX. An element $\alpha \in \mathcal{C}(X,Y)$ is called an **inclusion** when X is a subspace of Y and $U(\alpha)$ is the inclusion

$$\begin{tikzcd}
 & X & \\
 & I_r \arrow[u, "\partial^+\delta\cdot\delta=\delta"] & \\
I_r \arrow[uur, bend left, "\delta"] \arrow[ur, "\mathrm{id}"] & & I_0 \arrow[ul, "+r"'] \arrow[uul, bend right, "\partial^+\delta"'] \\
 & I_0 \arrow[ul, "+r"] \arrow[ur, "\mathrm{id}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & X & \\
 & I_r \arrow[u, "\gamma\cdot\partial^-\gamma=\gamma"] & \\
I_0 \arrow[uur, bend left, "\partial^-\gamma"] \arrow[ur, "+r"] & & I_r \arrow[ul, "\mathrm{id}"'] \arrow[uul, bend right, "\gamma"'] \\
 & I_0 \arrow[ul, "\mathrm{id}"] \arrow[ur, "+r"'] &
\end{tikzcd}$$

Figure 4.4: Identities of the category of dipaths of X

$$\begin{tikzcd}
X_0 \arrow[r] \arrow[d] & X_1 \arrow[d] \\
X_2 \arrow[r] & X
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{P}X_0 \arrow[r] \arrow[d] & \mathcal{P}X_1 \arrow[d] \\
\mathcal{P}X_2 \arrow[r] & \mathcal{P}X
\end{tikzcd}
\qquad
\begin{tikzcd}
{[0,1]} \arrow[r] & X' \\
\emptyset \arrow[r] \arrow[u] & {[2,3]} \arrow[u]
\end{tikzcd}$$

Figure 4.5: Squares of inclusions

of UX into UY. In this case the notation $\mathrm{int}(UX)$ stands for the topological interior of $U(X)$ seen as a subset of $U(Y)$ – see Definition 2.1.3. Then we have

Theorem 4.2.19 (Seifert - van Kampen for the dipath categories)**.**
Let X be an object of C and X_1, X_2 be subspaces of X such that $\{\mathrm{int}(UX_1), \mathrm{int}(UX_2)\}$ is a covering of UX. Denote by X_0 the subspace of X on $UX_1 \cap UX_2$. The middle square on Figure 4.5 is a pushout square of **Cat**.

Proof. The inclusions of subspaces provide the commutative square on the left hand side of Figure 4.5, the middle one is obtained as its image under $\mathcal{P}$. Let $f_i : \mathcal{P}X_i \to C$, for $i \in \{1, 2\}$, be functors to a (small) category C. Any point $x \in X$ belongs to $UX_1 \cup UX_2$ so we define $g(x)$ as $f_1(x)$ or $f_2(x)$ accordingly: both definitions match on $UX_1 \cap UX_2$. Given a dipath $\gamma : [0, r] \to X$, the subsets $\gamma^{-1}(\mathrm{int}(UX_i))$, for $i \in \{1, 2\}$, form an open covering of the compact space $[0, r]$. Let ε be a Lebesgue number of the covering (*i.e.* any compact interval of length at most ε is contained in some element of the covering). Then let $0 = a_0 < \cdots < a_n = r$ be such that $a_k - a_{k-1} < \varepsilon$. The restriction of γ to $[a_{k-1}, a_k]$ is a dipath of X_i for some $i \in \{1, 2\}$. So we define $g(\gamma)$ as the composite $f_{i_n} \circ \gamma|_{[a_{n-1},a_n]} \cdots f_{i_1} \circ \gamma|_{[a_0,a_1]}$ with $i_k \in \{1, 2\}$ depending on whether $f([a_{k-1}, a_k])$ is included in $\mathrm{int}(UX_1)$ or $\mathrm{int}(UX_2)$. It might happen that $\gamma|_{[a_{k-1},a_k]}$ is both a dipath on X_1 and a dipath on X_2. In that case it is a dipath on X_0 and both definitions coincide (we implicitly refer to the definition of subspace given by the fourth axiom of Definition 4.2.2). The other issue to address is that our construction seems to depend on the sequence $0 = a_0 < \cdots < a_n = r$. If $0 = b_0 < \cdots < b_m = r$ is another such sequence then consider $0 = c_0 < \cdots < c_p = r$ the strictly increasing enumeration of the union $\{a_0 < \cdots < a_n\} \cup \{b_0 < \cdots < b_m\}$. By associativity we have

$$f_{i_n} \circ \gamma|_{[c_{p-1},c_p]} \cdots f_{i_1} \circ \gamma|_{[c_0,c_1]} = \left\{ \begin{array}{l} f_{i_n} \circ \gamma|_{[a_n,a_{n-1}]} \cdots f_{i_1} \circ \gamma|_{[a_1,a_0]} \\ f_{i_n} \circ \gamma|_{[b_m,b_{m-1}]} \cdots f_{i_1} \circ \gamma|_{[b_1,b_0]} \end{array} \right.$$

□

$$D\left(\begin{tikzcd} & X & \\ & I_{r+s} \arrow[u, "\gamma\cdot\delta"] & \\ I_r \arrow[uur, bend left, "\delta"] \arrow[ur, "i^{r+s}_{0,r}"] & & I_s \arrow[uul, bend right, "\gamma"'] \arrow[ul, "i^{r+s}_{r,s}"'] \\ & I_0 \arrow[ul, "r"] \arrow[ur, "0"'] & \end{tikzcd}\right) = \begin{tikzcd} & D(X) & \\ & I'_{r+s} \arrow[u, "D(\gamma\cdot\delta)"'] & \\ I'_r \arrow[uur, bend left, "D(\delta)"] \arrow[ur, "i'^{r+s}_{0,r}"] & & I'_s \arrow[uul, bend right, "D(\gamma)"'] \arrow[ul, "i'^{r+s}_{r,s}"'] \\ & I'_0 \arrow[ul, "r"] \arrow[ur, "0"'] & \end{tikzcd}$$

Figure 4.6: Preserving concatenation

Remark 4.2.20. Let X be the sub-pospace of the directed line over $[0,1] \cup [2,3]$. The pospace X' given by the pushout square on the right hand side of Figure 4.5 (which is actually a coproduct since it is taken over the initial object of **PoTop**) is not isomorphic to X. Indeed one has $U(X) = U(X') = [0,1] \cup [2,3]$ yet $1 \sqsubseteq_X 2$ while $1 \not\sqsubseteq_{X'} 2$.

Remark 4.2.21. Given a point x of a pospace X, we have $\mathcal{P}X(x,x) \cong (\mathbb{R}_+, +, 0)$ since all the elements of $\mathcal{P}X(x,x)$ are constant dipaths (*cf.* Remark 4.1.7) and concatenation just stretches out the domain of definition of the mappings.

Definition 4.2.22. A morphism of frameworks for directed topology from C to C' (over $\mathcal{K}$) is a functor $D : C \to C'$ satisfying $U = U' \circ D$, $F' = D \circ F$, and $I' = D \circ I$.

Definition 4.2.22 will be proven useful in Section 5.3, as an appetizer consider $x, r, s \in \mathbb{R}^+$ such that $x + r \leqslant s$. The mapping $t \in [0,r] \mapsto x + t \in [0,s]$ induces a morphism of **Cub** whose images by I and I' are denoted by $i^s_{x,r}$ and $i'^s_{x,r}$. Since D is a morphism of frameworks, we have $D(i^s_{x,r}) = i'^s_{x,r}$ hence we have the diagrammatic equality pictured on Figure 4.6. From Definition 4.2.15 it follows that $D(\gamma * \delta) = D(\gamma) * D(\delta)$. Hence we have defined a functor α_X from $\mathcal{P}X_C$ to $\mathcal{P}DX_{C'}$ whose object part is an identity and which sends a path δ on X to the path $D(\delta)$ on DX.

Lemma 4.2.23. The collection of functors α_X for X running through the collection of objects of C forms a natural transformation from $\mathcal{P}_C$ to $\mathcal{P}_{C'}$. Moreover the functor α_X is an identity if and only if for all $r \in \mathbb{R}_+$ we have $C(I_r, X) = C'(I'_r, DX)$.

Proof. The statement readily follows from Definition 4.2.18. □

The remainder of the chapter provides several examples of frameworks for directed topology and compares them both from the mathematical and computer science points of view.

4.3 Locally Ordered Spaces

We have noticed that the directed circle cannot be modeled as a pospace. In order to take the semantics of such programs with loops into account, one has to overcome this limitation. A natural idea is to imitate the general philosophy of manifolds and define the **locally ordered spaces** as "patchworks" of pospaces. This approach has been originally proposed by Fajstrup et al. (2006). The mathematical gadgets that naturally crosses one's mind in this kind of situation are the sheaves – see (Pedicchio et al., 2003, Chap.VII) for an introduction. However we have to deal with the hindrance that "being partially ordered in a neighbourhood of a given point" is not a local notion: one may have no dipath from x to y in a given open set but having one such dipath in a larger open subset (*e.g.* consider an open proper arc of the circle and the anticlockwise paths on it). We will need to enforce this property. The definition of local pospaces given in this section slightly differs from the one found in Fajstrup et al. (2006). In fact they are not logically equivalent. In analogy with manifolds the local pospaces are defined as equivalence classes of ordered atlases, so we will have to check that our notion of ordered atlas is well-behaved in this respect.

Definition 4.3.1. Let X be a Hausdorff space. An **(ordered) chart** on X is a pospace U whose underlying space S_U is an open subset of X. An **(ordered) atlas** on X is a collection of ordered charts $\mathcal{U}$ on X whose underlying spaces form a basis of the topology of X and such that for all $U, V \in \mathcal{U}$ for all $x \in U \cap V$ there exists $W \in \mathcal{U}$ such that $x \in W \subseteq U \cap V$ and

$$\sqsubseteq_{U|_W} \;=\; \sqsubseteq_W \;=\; \sqsubseteq_{V|_W}$$

with $\sqsubseteq_{U|_W}$ standing for the relation whose graph is $(W \times W) \cap \sqsubseteq_U$.

Since there is no possible confusion with the usual notion of atlas on a differential manifold, we will just write "atlas" instead of "ordered atlas". In other words Definition 4.3.1 only requires that for all points x, two charts of a given atlas that both contain x match over a smaller chart containing x.

Remark 4.3.2. Let $\mathcal{U}$ be a collection of charts whose underlying spaces form an open cover of X and such that for all $x \in X$, all $U_0, U_1 \in C$ both containing x, there exists an open neighborhood W of x in $U_0 \cap U_1$ such that both U_0 and U_1 induce the same pospace structure on W. Then the collection of all open subpospaces of all the charts in $\mathcal{U}$ forms an atlas in the sense of Definition 4.3.1. This should be compared to Lemma 4.3.9.

As we shall see, an atlas might contain two globally incompatible ordered charts on the same open subset. In particular writing $U \subseteq U'$ for $U, U' \in \mathcal{U}$ we mean that the underlying space of U is included in the one of U' without taking the compatibility of their partial orders into account. Therefore $U \cap U'$ refers to the intersection of the underlying subspaces while $U \wedge U'$ refers to $U \cap U'$ equipped with the partial order $\sqsubseteq_U \cap \sqsubseteq_{U'}$ and thus forms a pospace.

Lemma 4.3.3. Let $U_1, \ldots, U_n$ be charts of an atlas $\mathcal{U}$ and $x \in U_1 \cap \cdots \cap U_n$. There exists $W \in \mathcal{U}$ such that $x \in W \subseteq U_1 \cap \cdots \cap U_n$ and $\sqsubseteq_{U_k|_W} = \sqsubseteq_W$ for all $k \in \{1, \ldots, n\}$.

Proof. By induction on $n \geqslant 1$. □

Lemma 4.3.4. If $\mathcal{U}$ is an atlas then so is the following collection of pospaces.

$$\{U_0 \wedge \cdots \wedge U_n \mid n \in \mathbb{N}; U_k \in \mathcal{U} \text{ for all } k = 1, \ldots, n\}$$

Proof. Given $U_0 \wedge \cdots \wedge U_p$ and $V_0 \wedge \cdots \wedge V_q$ we have, by Lemma 4.3.3, U' and V' in $\mathcal{U}$ such that for all i and j

$$\sqsubseteq_{U'} = \sqsubseteq_{U_i|_{U'}} \quad \text{and} \quad \sqsubseteq_{V'} = \sqsubseteq_{V_j|_{V'}}$$

and then we have $W \in \mathcal{U}$ included in $U' \cap V'$ such that

$$\sqsubseteq_W = \sqsubseteq_{U'|_W} = \sqsubseteq_{V'|_W}$$

□

Lemma 4.3.5. For all points x of a chart U of an atlas $\mathcal{U}$, and all neighborhoods V of x, there exists $U' \in \mathcal{U}$ such that $x \in U' \subseteq V$ and $\sqsubseteq_{U'} = \sqsubseteq_{U|_{U'}}$.

Proof. Since the underlying spaces of $\mathcal{U}$ form a basis of topology, we have some $U'' \in \mathcal{U}$ such that $U'' \subseteq V \cap U$. The expected U' is then given by Definition 4.3.1. □

Definition 4.3.6. Two atlases on the same space are **compatible** when their union is still an atlas.

Lemma 4.3.7. The atlases $\mathcal{U}$ and $\mathcal{V}$ are compatible iff for all $U \in \mathcal{U}$ for all $x \in U$ there exists $V \in \mathcal{V}$ such that $x \in V \subseteq U$ and $\sqsubseteq_V = \sqsubseteq_{U|_V}$.

Proof. Suppose that $\mathcal{U}$ and $\mathcal{V}$ are compatible. Since $\mathcal{V}$ induces a basis of topology there is $V' \in \mathcal{V}$ such that $x \in V' \subseteq U$. By hypothesis of compatibility we have $W \in \mathcal{U} \cup \mathcal{V}$ such that $x \in W \subseteq V' \cap U$ and

$$\sqsubseteq_W = \sqsubseteq_{V'|_W} = \sqsubseteq_{U|_W}$$

If $W \in \mathcal{V}$ then take $V = W$ otherwise apply Lemma 4.3.5 to obtain $V'' \in \mathcal{V}$ such that $x \in V'' \subseteq W$ and $\sqsubseteq_{V''} = \sqsubseteq_{V'|_{V''}}$. Then take $V = V''$ since

$$\sqsubseteq_{V'|_{V''}} = \sqsubseteq_{V'|_{W \cap V''}} = \sqsubseteq_{U|_{W \cap V''}} = \sqsubseteq_{U|_{V''}}$$

Conversely, given $U \in \mathcal{U}$, $V \in \mathcal{V}$, and $x \in U \cap V$ there is $V' \in \mathcal{V}$ such that $x \in V' \subseteq U$ and $\sqsubseteq_{V'} = \sqsubseteq_{U|_{V'}}$. Moreover there exists $V'' \in \mathcal{V}$ such that $x \in V'' \subseteq V \cap V'$ and $\sqsubseteq_{V|_{V''}} = \sqsubseteq_{V''} = \sqsubseteq_{V'|_{V''}} = \sqsubseteq_{U|_{V' \cap V''}} = \sqsubseteq_{U|_{V''}}$ so $\mathcal{U}$ and $\mathcal{V}$ are equivalent. □

Lemma 4.3.8. The notion of compatibility induces an equivalence relation over the collection of atlases sharing the same underlying space.

Proof. Symmetry and reflexivity are obvious, transitivity derives from Lemma 4.3.7. □

Lemma 4.3.9. The **downward closure** $\downarrow \mathcal{U}$ of an atlas $\mathcal{U}$ is an atlas.

$$\downarrow \mathcal{U} = \{(\ V\ ,\ \sqsubseteq_{U|_V}\ ) \mid V \text{ open }; \ U \in \mathcal{U}\ ; \ V \subseteq U\}$$

Proof. Let x be in the intersection of two ordered charts V and V' of the downward closure and suppose that their orders are induced by the charts U and U' of $\mathcal{U}$. Since the underlying spaces of the elements of $\mathcal{U}$ form a basis of the topology, we have $U'' \in \mathcal{U}$ such that $x \in U'' \subseteq V \cap V'$. From Lemma 4.3.3 we obtain $W \in \mathcal{U}$ such that $x \in W \subseteq U \cap U' \cap U''$ and $\sqsubseteq_W = \sqsubseteq_{U|_W} = \sqsubseteq_{U|_{V \cap W}} = \sqsubseteq_{V|_W}$. We conclude by remarking that the same holds for U' and V' instead of U and V. □

Lemma 4.3.10. The union of all the atlases of a compatibility class is an atlas.

Proof. Let $\mathcal{U}$ be the union. Given $V_1, V_2 \in \mathcal{U}$ and $x \in V_1 \cap V_2$ there are compatible atlases $\mathcal{V}_1$ and $\mathcal{V}_2$ such that $V_1 \in \mathcal{V}_1$ and $V_2 \in \mathcal{V}_2$. Therefore we have $W \in \mathcal{V}_1 \cup \mathcal{V}_2 \subseteq \mathcal{U}$ such that $x \in W \subseteq V_1 \cap V_2$ and $\sqsubseteq_W = \sqsubseteq_{V_1|_W} = \sqsubseteq_{V_2|_W}$. So $\mathcal{U}$ is an atlas. □

Remark 4.3.11. The maximum atlas $\mathcal{U}$ is stable under $\wedge$ (Lemma 4.3.4) and equal to its downward closure (Lemma 4.3.9).

Definition 4.3.12. An **atlas morphism** from $\mathcal{U}$ to $\mathcal{V}$ is a map f (between the underlying sets of $\mathcal{U}$ and $\mathcal{V}$) such that for all $x \in X$ and all $V \in \mathcal{V}$ containing $f(x)$, there exists $U \in \mathcal{U}$ containing x such that f induces a pospace morphism from U to V.

Remark 4.3.13. By Definition 4.3.12, a morphism of atlases is necessarily continuous because the underlying spaces of the charts of $\mathcal{U}$ (resp. $\mathcal{V}$) induces a basis of the underlying topology of $\mathcal{U}$ (resp. $\mathcal{V}$).

Definition 4.3.12 can be rephrased in a seemingly weaker form.

Lemma 4.3.14. A mapping is an atlas morphism iff for all $x \in \operatorname{dom} f$ there exists a directed chart $U \in \mathcal{U}$ and a directed chart $V \in \mathcal{V}$ such that $x \in U$ and f induces a pospace morphism from U to V (implicitly $f(U) \subseteq V$).

Proof. Let $V' \in \mathcal{V}$ containing $f(x)$, and $U \in \mathcal{U}$, $V \in \mathcal{V}$ such that $x \in U$ and f induces a pospace morphism from U to V. Since $\mathcal{V}$ is an atlas it contains an ordered chart $V'' \subseteq V \cap V'$ whose partial order both coincide with the partial orders of V and V'. Applying Lemma 4.3.5 it comes $U' \in \mathcal{U}$ that is included in $U \cap f^{-1}(V'')$ and whose partial order coincide with the one of U. Therefore f induces a pospace morphism from U' to V'. □

Lemma 4.3.15. Let f be a continuous map. Let $\mathcal{U}$ and $\mathcal{U}'$ (resp. $\mathcal{V}$ and $\mathcal{V}'$) be equivalent atlases on $\operatorname{dom} f$ (resp. $\operatorname{codom} f$). Then $f : \mathcal{U} \to \mathcal{V}$ is an atlas morphism iff $f : \mathcal{U}' \to \mathcal{V}'$ is so.

Proof. By Lemma 4.3.7. □

Denote by S_U the underlying space of an ordered chart $U \in \mathcal{U}$ (*cf.* Definition 4.3.1) and let $\sqsubseteq$ be the intersection of all the partial orders associated with some element of $\mathcal{U}$ whose underlying space is S_U. The space S_U equipped with $\sqsubseteq$ forms a pospace denoted by Core(U). Then define Core($\mathcal{U}$) as the collection of pospaces $\{\mathrm{Core}(U) \mid U \in \mathcal{U}\}$.

Open question 4.3.16. Is Core($\mathcal{U}$) an atlas? What if $\mathcal{U}$ is the maximal atlas?

These preliminaries led us to the central notion of this section.

Definition 4.3.17. A **locally ordered space** (or **local pospace**) is a Hausdorff space together with an equivalence class of directed atlases. A **local pospace morphism** is an atlas morphism (this definition is made sound by Lemma 4.3.15). These data define the category **LpoTop** of local pospaces.

Remark 4.3.18. Given an atlas $\mathcal{U}$ on a space X and a subspace Y of X, the collection of sub-pospaces of the form $U \cap Y$ with $U \in \mathcal{U}$, is an atlas on Y. The inclusion map $Y \hookrightarrow X$ thus becomes a local pospace morphism and the forgetful functor U satisfies the fourth axiom of Definition 4.2.2. In this context, Y is a sub local pospace of X.

Remark 4.3.19. By Definition 4.3.12, being a local pospace morphism is a local property. As a consequence, for all local pospaces X and Y, the functor **LpoTop**(_, Y) defined over the locale of open subsets of X is a **sheaf** (*cf.* (Mac Lane and Moerdijk, 1994, p.65) or (Pedicchio et al., 2003, p.316)). In other words it satisfies the **gluing condition**: the following diagram is an equalizer in **Set**, where U is an open subset of (the underlying space of) X, and V and W range through a given open cover C of U.

$$\mathbf{LpoTop}(U,Y) \to \prod_{V\in C} \mathbf{LpoTop}(V,Y) \quad \rightrightarrows \quad \prod_{(V,W)\in C\times C} \mathbf{LpoTop}(V\cap W,Y)$$

The arrows on the preceding diagram should be understood as follows. Given $V \subseteq U$, the leftmost arrow is the precomposition by the inclusion map $V \hookrightarrow U$. Such a map is called a **restriction**. Moreover, given an ordered pair (V, W) of open subsets of U, we have two restrictions according to V and W (*i.e.* the first and the second components of the ordered pair).

$$\mathbf{LpoTop}(V, Y) \to \mathbf{LpoTop}(V \cap W, Y) \leftarrow \mathbf{LpoTop}(W, Y)$$

The product of all the first (resp. second) components provide the first (resp. second) arrow. In more concrete terms, it says that one obtains a local pospace morphism over U from any family of local pospace morphisms defined over the elements of an open cover of U, provided that any two of them agree on the intersection of their domains of definition.

Remark 4.3.20. Given a Hausdorff space X, the collection of the discrete pospaces on the open subspaces of X is an atlas. The resulting local pospace is said to be discrete. In particular the forgetful functor $U : \mathbf{LpoTop} \to \mathbf{Haus}$ admits a left adjoint and satisfies the third axiom of Definition 4.2.2.

Remark 4.3.21. The collection of ordered charts induced by a pospace on its open subsets is an atlas, and any pospace morphism gives rise to a morphism between the corresponding atlases. These data induce a functor

$$A : \mathbf{PoTop} \to \mathbf{LpoTop}$$

which is neither full nor injective on objects. Indeed let X be the subpospace of $\mathbb{R}$ over $[0, 1] \cup [2, 3]$. The homeomorphism from $[0, 1] \cup [2, 3]$ to itself that swaps the connected components induces an endomorphism of X in **LpoTop**, but not in **PoTop**. Moreover if we let X' be the coproduct in **PoTop** of $[0, 1]$ and $[2, 3]$, then X and X' have the same image under the inclusion functor yet they are not isomorphic in **PoTop**. As a matter of notation A stands for “atlas”.

Definition 4.3.22. The local pospace induced by a pospace X is, by definition, its image under A.

As an immediate consequence of Lemma 4.3.7 we have

Proposition 4.3.23. Two pospaces sharing their underlying space X induce the same local pospace iff every point of X admits an open neighborhood on which both partial orders coincide. As a consequence, a local pospace lies in the image of the inclusion functor iff its greatest atlas contains an ordered chart supported by its underlying space.

Example 4.3.24. The directed real line in **LpoTop** is the local pospace induced by the directed real line in **PoTop** (*cf.* Example 4.1.3 and Definition 4.3.22). The following collections of ordered charts are equivalent atlases over $\mathbb{R}$.

1. $\{(I, \leqslant) \mid I$ open interval of $\mathbb{R}\}$,
2. $\{(U, \leqslant) \mid U$ open subset of $\mathbb{R}\}$,
3. $\{(U, \sqsubseteq_U) \mid U$ open subset of $\mathbb{R}\}$ where $x \sqsubseteq_U y$ stands for $x \leqslant y$ and $[x, y] \subseteq U$,
4. $\{(U, \sqsubseteq'_U) \mid U$ open subset of $\mathbb{R}\}$ where $x \sqsubseteq'_U y$ is any extension of $\sqsubseteq_U$.

The first one naturally comes to mind. The second one is induced by the pospace $\mathbb{R}$. In the same fashion, the third one is provided by coproducts (in **PoTop**) of directed open intervals. The last atlas is the greatest one (*cf.* Lemma 4.3.10), note that it contains charts which have the same underlying space and which are yet not globally compatible.

Proposition 4.3.25. If X is a pospace and δ is a dipath on AX starting at x and ending at x', then $x \sqsubseteq_X x'$.

Proof. Suppose that the domain of δ is r. The collection $\mathcal{V}$ of open sub pospaces of X is an atlas of AX. In particular we have $\sqsubseteq_{X|_V} = \sqsubseteq_V$ for all $V \in \mathcal{V}$. By Definition 4.3.12 we have a covering $\mathcal{U}$ of $[0, r]$ by charts of I_r such that for all $U \in \mathcal{U}$ there exists $V \in \mathcal{V}$ such that δ induces a pospace morphism from U to V. Given a sequence

$$0 = t_0 < \cdots < t_n = r$$

such that $t_k - t_{k-1}$ is strictly less than the Lebesgue number of the covering $\mathcal{U}$, we have for all $k \in \{1, \ldots, n\}$, a chart $V_k \in \mathcal{V}$ such that $\delta(t_{k-1}) \sqsubseteq_{V_k} \delta(t_k)$. Since we have $\sqsubseteq_{X|_{V_k}} = \sqsubseteq_{V_k}$ for all $k \in \{1, \ldots, n\}$ we actually have $x = \delta(0) \sqsubseteq_X \delta(r) = x'$. □

Corollary 4.3.26. Let X be a pospace such that $x \sqsubseteq_X x'$ implies the existence of a dipath on X (*cf.* Definition 4.1.4) from x to x'. Then for all pospaces Y we have

$$\mathbf{LpoTop}(AX, AY) = \mathbf{PoTop}(X, Y)$$

Proof. Let f be a local pospace morphism from AX to AY and suppose that we have $x \sqsubseteq_X x'$. If γ be a dipath on X from x to x' then $f \circ A(\gamma)$ is a dipath on AY from $f(x)$ to $f(x')$. From Proposition 4.3.25 it comes $f(x) \sqsubseteq_Y f(x')$ so f is actually a pospace morphism. □

As an immediate consequence of Corollary 4.3.26 we have

Lemma 4.3.27. A dipath δ on a local pospace X is constant iff its extremities are equal and there exists an ordered chart of some atlas of X that contains the image of δ.

Proof. If U is a chart as in the statement of the lemma, then δ is a dipath on U hence a local pospace morphism. By Corollary 4.3.26 it is also a pospace morphism and therefore it is constant. □

Lemma 4.3.28. The category **LpoTop** is a directed framework for directed topology.

Proof. The third and fourth axioms of Definition 4.2.2 are given by Remark 4.3.20 and Remark 4.3.18. The fifth one is obvious and the embedding of **Cub** in **LpoTop** is provided by the composite of **Cub** $\hookrightarrow$ **PoTop** (*cf.* Remark 4.1.21) followed by the functor A (*cf.* Remark 4.3.21). The resulting functor is full by Corollary 4.3.26. □

Example 4.3.29. The example that motivates the notion of local pospace is the directed circle which we now describe by exporting the local pospace structure of the directed line to $\mathbb{S}^1$ through the exponential map $t \in \mathbb{R} \mapsto e^{it} \in \mathbb{S}^1$. Indeed all the statements given in this example are mainly due to the fact that the exponential map is the universal covering of $\mathbb{S}^1$ (Hatcher, 2002, Chap.1) (*i.e.* any continuous map from an interval of $\mathbb{R}$ to $\mathbb{S}^1$ can be factored through the exponential map in a unique way up to a translation by 2π). An arc is a connected proper subspace of $\mathbb{S}^1$. Any arc is the image of some interval of $\mathbb{R}$ under the exponential map. Then we have the following collection of compatible atlases on $\mathbb{S}^1$, the last of which being the greatest one.

1. $\{(A, \leqslant) \mid A \text{ open arc}\}$ where $\leqslant$ is the order induced by $\mathbb{R}$ and the restriction of the exponential map to an open subinterval of $\{t \in \mathbb{R} \mid \mathrm{e}^{\mathrm{i}t} \in A\}$ of length at most 2π,

2. $\{(U, \sqsubseteq_U) \mid U$ proper open subset of $\mathbb{S}^1\}$ where $x \sqsubseteq_U y$ means that the anticlockwise compact arc from x to y is included in U,

3. $\{(U, \sqsubseteq'_U) \mid U$ proper open subset of $\mathbb{S}^1\}$ where $\sqsubseteq'_U$ is any extension of the partial order $\sqsubseteq_U$.

We have thus defined a local pospace structure on $\mathbb{S}^1$ which is called the directed circle. Its directed paths are indeed the paths δ on $\mathbb{S}^1$ that can be written as $\delta = \mathrm{e}^{i\gamma}$ where γ is a dipath on $\mathbb{R}$. In fact it can be obtained as the following coequalizer in **LpoTop** thus addressing the issue we have met in **PoTop** – see Remark 4.1.7.

$$\begin{tikzcd} \{*\} \arrow[r, shift left, "0"] \arrow[r, shift right, "1"'] & {[0,1]} \end{tikzcd}$$

It is worth noticing that no ordered chart of the greatest atlas is supported by $\mathbb{S}^1$ hence the directed circle does not arise from a pospace (*cf.* Proposition 4.3.23). However the **undirected circle** (*i.e.* the discrete pospace over $\mathbb{S}^1$) induces a local pospace over the circle. We insist on this dummy example to emphasize the difference between the loops in algebraic topology and the diloops.

Remark 4.3.30. Example 4.3.29 offers an opportunity to compare our notion of local pospace to the original one. An *original* atlas is a Hausdorff space X together with a collection of pospaces $\mathcal{U}$ whose underlying spaces form an open cover of X and such that every point $x \in X$ comes with a pospace W_x such that

- the underlying space of W_x is a neighborhood of x, and
- given any $U \in \mathcal{U}$ containing x, the pospaces induced on $U \cap W_x$ by U and W_x coincide.

An *original* local pospace is an equivalence class of original atlases, both of them being equivalent when their union is again an original atlas. As so often in Mathematics, the problem arises from quantifier ordering. Given a point $x \in X$, the pospace W_x only depends on x. We claim that it should not be so. For example, the second atlas described on the above list is not an atlas in the original sense. Indeed it contains open dense subsets of arbitrary low nonzero measure (*i.e.* the accumulated length of its connected components). Let x be any point of $\mathbb{S}^1$, its associated pospace W_x contains a neighborhood A of x which is a closed arc. Then there exists an open dense subset D of $\mathbb{S}^1$ which contains x and whose measure is strictly less than the length of A. Let a_0 and a_1 be the extremities of A with a_0 coming before a_1 in the partial order on W_x. Then a_0 and a_1 are not comparable in D because $A \not\subseteq D$ for measure consideration.

Starting from that, one may ask how annoying is that situation. We might as well consider such an atlas as irrelevant. Then let us observe that the identity map on $\mathbb{S}^1$ induces an isomorphism between the first and the second atlases on the list of Example 4.3.29. In fact, given an atlas in the original sense, we obtain an atlas in the sense of Definition 4.3.1 by gathering, for all $x \in X$, all the open sub-pospaces of W_x containing x (*cf.* Remark 4.3.2). In categorical terms the category of original atlases is a full subcategory of the category of atlases in the sense of Definition 4.3.1. The issue of whether the full inclusion is an equivalence, which amounts to have an original atlas in each equivalence classes (*cf.* Lemma 4.3.8), is open.

We advocate shifting to the new formalism for, at least, the following three reasons. First, the original notion of atlas requires a mapping, namely the one that associates each point with a pospace, besides the collection of charts, whereas the latter should be, in the author point of view, intrinsic. Second, the new definition is better suited to the analogy with manifolds. Indeed, a collection of charts forms a manifold when the charts are compatible two-by-two. Finally, it is also better suited to local orders arising from a given class of paths of a space, that is to say when for all charts U of the atlas $\mathcal{U}$, we write $x \sqsubseteq_U y$ when there exists a path from x to y whose image is wholly contained in U, and which belongs to the distinguished class. Such local pospaces can be met when one tries to realize a precubical. They also naturally arise when the class of distinguished paths consists of the smooth curves γ on a manifold $\mathbb{M}$ with vector fields $f_1, \ldots, f_n$ such that for all $t \in \operatorname{dom}\gamma$ the vector $\dot{\gamma}(t)$ is a linear combination of vectors $f_1(t), \ldots, f_n(t)$ with non-negative coefficients. Note that the directed circle falls into this range of examples.

Corollary 4.3.26 and the definition of the category $\mathbf{PoTop}_{\text{d}}$ (*cf.* Definition 4.1.4) suggest to introduce the following class of local pospaces.

Definition 4.3.31. A local pospace X is said to by **directed by the dipaths** when it has an atlas whose charts are directed by the dipaths. These local pospaces form the full subcategory $\mathbf{LpoTop}_{\text{d}} \subseteq \mathbf{LpoTop}$.

Remark 4.3.32. As a consequence of Lemma 4.3.9, an ordered chart U of the maximal atlas $\mathcal{U}$ satisfies

$$\forall U' \in \mathcal{U} \;\; U \subseteq U' \;\; \Rightarrow \;\; \sqsubseteq_U = \sqsubseteq_{U'}|_U$$

iff U is the only element of $\mathcal{U}$ whose underlying space is S_U. In particular, if the subcollection of charts of $\mathcal{U}$ satisfying this property induces a basis of the topology, then it is an atlas. It is the case in Example 4.3.24 and Example 4.3.29 considering the ordered charts with connected underlying spaces. If the underlying space is locally connected (*i.e.* its connected components are open), then given an ordered chart U of some atlas $\mathcal{U}$ we write $x \sqsubseteq' y$ when $x \sqsubseteq_U y$ and x, y belong to the same connected component. Then S_U equipped with $\sqsubseteq'$ is a pospace that can be added to $\mathcal{U}$ so as to obtain a larger atlas. One can even go further considering the map q identifying points that belong to the same connected component. Then any pospace structure on the set of connected components of U (*i.e.* the codomain of q) can be lifted to an alternative closed partial order on S_U. The resulting pospace can also be added to $\mathcal{U}$ so as to obtain a larger atlas. As a consequence, when the underlying space of a local pospace is locally connected, if U is the only ordered chart over S_U in the maximal atlas, then S_U is connected.

We leave the topological aspects of local pospaces and now focus on the categorical properties of **LpoTop**.

Lemma 4.3.33. The category **LpoTop** is finitely complete and has all coproducts.

Proof. The equalizer of two parallel local pospace morphisms f and g is

$$\{x \in \operatorname{dom} f \mid f(g) = g(x)\}$$

equipped with the induced local pospace structure. The product of two local pospaces is carried by the product of the underlying topological spaces equipped with the following atlas

$$\{U \times V \mid U \in \mathcal{U}; V \in \mathcal{V}\}$$

where $\mathcal{U}$ and $\mathcal{V}$ are atlases and $U \times V$ is the product in **PoTop**. The coproduct is given by the coproducts of the underlying spaces together with the union of the atlases. □

Proposition 4.3.34. Let p be a point of the local pospace X and γ be a dipath on X. Then $\gamma^{-1}(\{p\})$ is a finite union of disjoint compact intervals.

Proof. From basic topological arguments the connected components of $\gamma^{-1}(\{p\})$ form a family of disjoint compact intervals. Define A as the collection of their middle points. Let $t, t' \in A$ be satisfying $t \leqslant t'$. Since γ is a local pospace morphism there is an ordered chart U containing t and an ordered chart V containing p such that $\gamma(U) \subseteq V$. If $t' \in U$ then by Lemma 4.3.27 the restriction of γ to $[t, t']$ is constant. In particular t and t' belong to the same connected component of $\gamma^{-1}(\{p\})$, hence $t = t'$ because both are the middle of this connected component. Thus A is a subset of a compact space without any accumulating point, so it is finite. □

Example 4.3.35. The following diagram is a coequalizer, with pr_2 being the second projection

$$\begin{tikzcd} \mathbb{S}^1 \arrow[r, shift left, "z\mapsto(z,0)"] \arrow[r, shift right, "z\mapsto(1,0)"'] & \mathbb{S}^1 \times \mathbb{R}_+ \arrow[r, "pr_2"] & \mathbb{R}_+ \end{tikzcd}$$

Let f be a local pospace morphism defined over $\mathbb{S}^1 \times \mathbb{R}_+$ such that $f(z, 0) = f(1, 0)$ for all $z \in \mathbb{S}^1$. Consider $X = \{x \in \mathbb{R}_+ \mid \forall z \in \mathbb{S}^1\ \forall t \in [0, x],\ f(z, t) = f(1, t)\}$. Then X is a nonempty (because it contains 0) subset of the connected space $\mathbb{R}_+$. First remark that X is closed since the mapping f can be seen as a continuous map from $\mathbb{R}_+$ to $(\operatorname{codom} f)^{\mathbb{S}^1}$. For the same reason, if $x \in X$ and V an ordered chart of codom f that contains x, then x admits a neighborhood U whose image under f is included in V. By Lemma 4.3.27 we deduce that for all $t \in U$ the function $f(t) \in (\operatorname{codom} f)^{\mathbb{S}^1}$ is constant. Therefore X is open, and thus $X = \mathbb{R}_+$. In particular the forgetful functor **LpoTop** → **Haus** does not preserve coequalizers.

Remark 4.3.36. Example 4.3.35 is related to the realization of cubical sets in **LpoTop**. In the spirit of Example 2.4.4 let us start with the square $\square_2$. Its **LpoTop**-realization is $[0, 1]^2$. We add the relation

$$K(\delta^1_{0,0})(s) = K(\delta^1_{0,1})(s) \text{ [rel. 1]}$$

which identifies two opposite edges of the square, the vertical ones say. The resulting cubical set is denoted by C (which stands for "cylinder"). As one can expect, the **LpoTop**-realization of C is $\mathbb{S}^1 \times [0, 1]$ (instead of $\mathbb{S}^1 \times \mathbb{R}_+$). Then we add another relation

$$K(\delta^1_{1,0})(s) = K(\sigma^0_0\delta^0_{0,0}\delta^1_{1,0})(s) \text{ [rel. 2]}$$

which identifies the lower edge of the square with a point (more precisely the lower left corner of the square). Intuitively we have created a vortex and denote the resulting cubical set by D (which stands for "disk"). In terms of **LpoTop**-realization, the subspace $\mathbb{S}^1 \times \{0\}$ of $\mathbb{S}^1 \times [0, 1]$ is identified with a point. In analogy with what we have seen before, the resulting local pospace is $[0, 1]$ (instead of $\mathbb{R}_+$) while the realization in **Top** is homeomorphic to the unit compact disk. Furthermore denote by S (which sounds like "sphere") the cubical set obtained by adding yet another relation

$$K(\delta^1_{1,1})(s) = K(\sigma^0_0\delta^0_{0,1}\delta^1_{1,1})(s) \text{ [rel. 3]}$$

then the **LpoTop**-realization of S is again $[0, 1]$ while its **Top**-realization is $\mathbb{S}^2$.

Definition 4.3.37. A **vortex** is a point any neighborhood of which contains a nonconstant directed loop.

An immediate consequence of Lemma 4.3.27 is

Corollary 4.3.38. A locally ordered space has no vortex.

Example 4.3.39. We deduce from Corollary 4.3.38 that the complex plane $\mathbb{C}$ cannot be provided with a local pospace structure whose dipaths would be

$$\{\rho(t) \cdot \mathrm{e}^{i\theta(t)} \mid r \geqslant 0 \; ; \; \theta : [0, r] \to \mathbb{R} \; ; \; \rho : [0, r] \to \mathbb{R}_+ \; ; \; \rho, \theta \text{ nondecreasing}\}$$

since the origin would then be a vortex.

Remark 4.3.40. Another consequence of Corollary 4.3.38 is that **LpoTop** lacks some infinite products. Consider indeed a family of copies of the directed circle indexed by $\mathbb{N}$. If the product exists then its underlying space is the topological product of countably many copies of the circle because the forgetful functor to **Haus** has a left adjoint. Then for each n consider the directed path γ_n defined over $[0, 2\pi]$ by

$$\gamma_n(t) = e^{int} \; .$$

There is a directed path δ over the product such that $pr_n \circ \delta = \gamma_n$ for all n. Given any open interval J of $[0, 2\pi]$ there exists n such that $pr_n(\delta(J)) = \mathbb{S}^1$. A basis of the product topology is given by the products

$$\prod_{n \in \mathbb{N}} U_n$$

of the families $(U_n)_{n \in \mathbb{N}}$ of nonempty open subsets of $\mathbb{S}^1$ only finitely many members of which differ from $\mathbb{S}^1$. If this product of local pospaces exists, then all its points are vortices, which is a contradiction.

A careful reading will convince the reader that the arguments given in Remark 4.3.40 are still valid when one replaces the infinite family of copies of $\mathbb{S}^1$ by any infinite family of local pospaces whose members contain a directed loop. Therefore we have

Proposition 4.3.41. If a family of local pospaces has infinitely many terms containing a non constant directed loop, then its product in **LpoTop** does not exist.

In fact we have

Proposition 4.3.42. The product of a family of local pospaces exists iff all its members but finitely many ones are pospaces (*i.e.* of the form AX for some pospace X).

Proof. An atlas is given by products in **PoTop** of families of ordered charts. □

Cocompleteness of **LpoTop** is still an open problem.

Example 4.3.43. Consider the following diagram of local pospaces

$$\mathbb{T}^0 \hookrightarrow \mathbb{T}^1 \hookrightarrow \cdots \hookrightarrow \mathbb{T}^n \hookrightarrow \mathbb{T}^{n+1} \hookrightarrow \cdots$$

with $\mathbb{T}^n$ denoting the n-fold Cartesian product of $\mathbb{S}^1$. The colimit in **Haus** of the underlying spaces is the product of sets

$$\prod_{\mathbb{N}} \mathbb{S}^1$$

endowed with the topology a basis of which is given by *all* the products of the families $(U_n)_{n \in \mathbb{N}}$ of nonempty open proper subsets of $\mathbb{S}^1$. Each of them is equipped with the product order, thus providing the atlas.

Remark 4.3.36 suffices to convince us that **LpoTop** is not a good framework to realize cubical sets in. However it could be a convenient one if we restrict to precubical sets. Indeed it is proven in (Fajstrup et al., 2006, Section 6) that one has a realization functor in **LpoTop** defined over the full subcategory of non-self-linked precubical sets (*i.e.* any face y of any cube x, can be written as $y = K(\omega)(x)$ for a unique morphism ω of $\square^{+}$). This restriction is required in order to ensure that the face ordering (*i.e.* $y <_F x$ when $y = K(\omega)(x)$ for some morphism ω of $\square^{+}$) has nice properties. In particular the face ordering is used to provide the topological realization of a non-self-linked precubical set with a *local partial order* in the sense of (Fajstrup et al., 2006, Definition 3.4). Unlike atlases, the local partial orders are not required to be carried by a basis of the topology.

Let K be a precubical set and $|K|$ be its topological realization. For all cubes $x \in K$, we have a continuous map $\phi_x : [0,1]^{\dim x} \to |K|$ whose restriction to $]0,1[^n$ is a homeomorphism on its image. We define $\mathcal{D}$ as the least collection of paths on $|K|$ that is stable under concatenation and contains all the paths of the form $\phi_x \circ \delta$ with $\delta \in \mathbf{PoTop}([a,b],[0,1]^{\dim x})$. For all open subsets U of $|K|$, we write $x \preccurlyeq_U y$ when there exists a path $\delta \in \mathcal{D}$ from x to y whose image is contained in U.

Conjecture 4.3.44 (Unpublished)**.** Let K be a precubical set.

- For all open subsets U of $|K|$, the relation $\preccurlyeq_U$ is closed (*i.e.* $(U, \preccurlyeq_U)$ is a preordered space – see Definition 4.1.10)
- The collection $\mathcal{B}$ of open subsets U such that $\preccurlyeq_U$ is antisymmetric (*i.e.* $(U, \preccurlyeq_U)$ is a pospace – see Definition 4.1.1) forms a basis of the topology.
- The collection $\mathcal{U} = \{(U, \preccurlyeq_U) \mid U \in \mathcal{B}\}$ is an atlas on $|K|$, the topological realization of K.
- The set of dipaths on $(|K|, \mathcal{U})$ is $\mathcal{D}$ and all the charts of $\mathcal{U}$ are directed by the dipaths – see Definition 4.1.4
- The **LpoTop**-realization of K is $(|K|, \mathcal{U})$ – see Definition 2.2.1.

To recap, the category of local pospaces is finitely complete and contains all the coproducts. However it is not complete and we conjecture that it is not cocomplete either. Anyway, some of its existing colimits do not fit with the colimit of the underlying spaces. These unpleasant features are related to the fact that local pospaces have no vortex. However, as shown by the introduction of Section 6.4 and Definition 7.1.2, local pospaces are sufficiently supple to model any parallel automaton. Moreover their fundamental categories (*cf.* Definition 5.2.10) enjoy nice properties (*cf.* Proposition 5.2.13).

4.4 Streams

From a computer science point of view, local pospaces are satisfactory as they are sufficiently supple to model any parallel automata. Nevertheless, the category **LpoTop** is really ill-behaved with regard to algebraic directed topology. Indeed the properties of the geometric realization of cubical sets (Chapter 2) heavily rely on the nature of the aimed category. As local pospaces, streams are intended to formalize "local causality". However, the notion of local pospace is based on the assumption that causality on an open set imposes causality on its open subsets. In this context, causality is represented by partial orders. On the contrary, the notion of stream arises from the assumption that causality on some open subset U should be the sum of the causalities carried by the elements of any open cover of U. This approach fits better with transitivity (*cf.* bad behaviour of colimits in **LpoTop** – see Example 4.3.35). Also note that in the definition of stream the constraint of working with pospaces is relaxed. Given $\mathcal{P}$ a family of preordered sets we denote by

$$\bigvee \mathcal{P}$$

the least preorder (on the union of the underlying sets of the elements of $\mathcal{P}$) containing all the binary relations associated with the elements of $\mathcal{P}$. For the sake of compatibility with pospaces and local pospaces, we suppose that the underlying space of a stream is Hausdorff. The streams were introduced by S. Krishnan (2006).

Definition 4.4.1 (Krishnan (2009))**.** A **circulation** on a topological space is a mapping that associates any open subset W with a preorder relation $\preccurlyeq_W$ on it and such that the following holds for all $\mathcal{U}$ open covers of W

$$\preccurlyeq_W = \bigvee \{\preccurlyeq_U \mid U \in \mathcal{U}\}$$

A **stream** is a Hausdorff space together with a circulation.

Remark 4.4.2. Given two open subsets U and U' of a stream X such that $U \subseteq U'$, we have $\preccurlyeq^X_U \subseteq \preccurlyeq^X_{U'}$ because the following relation holds by Definition 4.4.1.

$$\preccurlyeq^X_{U'} \quad = \quad \preccurlyeq^X_U \vee \preccurlyeq^X_{U'}$$

Remark 4.4.3. Let U be an open subset of a stream X, and suppose that U can be written as the disjoint union of two open sets U_0 and U_1. Then an element of U_0 and an element of U_1 are not comparable with respect to $\preccurlyeq^X_U$.

Example 4.4.4. The third directed atlas described in Example 4.3.24 is actually a direction while the second one is not by Remark 4.4.3. In particular, any interval of $\mathbb{R}$ can be provided with a stream structure assigning to each of its open subsets W the following partial order: for w and w' in W write $w \preccurlyeq_W w'$ when $w \leqslant w'$ and $[w, w'] \subseteq W$. It is referred to as the standard stream on the interval. The standard stream structure on $[0, 1]$ (resp. $\mathbb{R}$) is the directed unit interval of **Strm** (resp. directed line of **Strm**). Following Definition 4.2.12, the dipaths on a stream X are defined as the morphisms from some standard stream $[0, r]$ to X, for $r \in \mathbb{R}_+$.

Example 4.4.5. The second directed atlas described in Example 4.3.29 can be extended to a circulation by associating $\mathbb{S}^1$ to the chaotic preorder (*i.e.* the one such that any point is related to all the others).

Definition 4.4.6. A **stream morphism** from X to Y is is a map f (between the underlying sets of X and Y) such that for all $x \in X$ and all V open subset of Y containing $f(x)$, there exists U open subset of X containing x such that f induces a preorder morphism from $(U, \preccurlyeq^X_U)$ to $(V, \preccurlyeq^Y_V)$.

It follows from Definition 4.4.6 that stream morphisms compose. The category of streams is denoted by **Strm**.

Remark 4.4.7. Definition 4.4.6 and Definition 4.3.12 are formally similar. In fact they only differ in the axioms that express local causality (*i.e.* directed atlases *vs* circulations). In particular, being a stream morphism is a local property so Remark 4.3.19 remains valid writing “stream” instead of “local pospace”. Nevertheless, Lemma 4.3.14 does not hold for stream morphisms. As a counter-example, consider the circulation on the directed circle described in Example 4.4.5. Indeed, it is clear that all continuous maps from the circle to itself preserve the chaotic preorder but many of them are not stream morphisms.

The forgetful functor U from **Strm** to **Haus** admits a left adjoint provided by the discrete relation on every open subset. It also admits a right adjoint by associating any topological space with its greatest stream structure, yet this structure cannot be easily described (*cf.* Krishnan (2009)). The category of streams enjoys many nice properties, in particular it is complete. However the products in **Strm** are not straightforward and require one to weaken the notion of stream. We still follow the terminology of (Krishnan (2009)) calling **precirculation** a mapping that sends any open subset W of a Hausdorff space X to a preorder relation $\preccurlyeq_W$ on W. In addition, this mapping is required to preserve inclusion in the following sense: for all open subsets W_1, W_2 of X such that $W_1 \subseteq W_2$, we have $\preccurlyeq_{W_1} \subseteq \preccurlyeq_{W_2}$. As suggested by the terminology, any circulation is a precirculation – see Remark 4.4.2. A **prestream** is a Hausdorff space together with a precirculation. The prestream morphisms are defined as the stream morphisms. Prestreams and their morphisms form the category **pStrm** and there is a full inclusion **Strm** $\to$ **pStrm**. The forgetful functor **pStrm** $\to$ **Haus** admits both a left adjoint (defined as for streams) and a right one. The latter assigns the chaotic preorder to each open subset. We write $\preccurlyeq^X$ to denote the precirculation of a prestream X. Given two precirculations $\preccurlyeq$ and $\preccurlyeq'$ on the same Hausdorff space, one says that $\preccurlyeq$ is contained in $\preccurlyeq'$ (or that $\preccurlyeq$ is less that $\preccurlyeq'$) when $\preccurlyeq_W \subseteq \preccurlyeq'_W$ for all open subsets W. The **cosheafification**[1] of a prestream X is defined, for all open subsets W, by the least preorder on W that contains all the preorders $\preccurlyeq_W$ with $\preccurlyeq$ ranging through the collection of all circulations contained in $\preccurlyeq^X$. By a slight abuse of language, one may say that a prestream X is less that a prestream X' when so are their corresponding precirculations. Then in an abstract way, the cosheafification of a prestream X, denoted by $\cosh X$, is the greatest stream contained in X. As one can imagine, the cosheafification construction induces a right adjoint to the inclusion **pStrm** $\to$ **Strm**. In particular the following commutative diagram provides the right adjoint to the forgetful functor **Strm** $\to$ **Haus** as a composite of right adjoints.

$$\begin{tikzcd}
 & \mathbf{Haus} \\
\mathbf{Strm} \arrow[ur, "U"] \arrow[r] & \mathbf{pStrm} \arrow[u, "U"']
\end{tikzcd}$$

Furthermore the product of a family of streams is given by the cosheafification of its product in **pStrm** (*cf.* Krishnan (2009)). In other words the cosheafification functor **pStrm** $\to$ **Strm** preserves products. Yet, the product of a family of streams in **pStrm** almost always differ from its product in **Strm**.

Remark 4.4.8. As pointed out in Krishnan (2009), most of the constructions in **Strm** are in fact performed in **pStrm** and then sent to **Strm** via the cosheafification functor. For example, given a pospace X, one can naively define the mapping that associates each open subset of X with the restriction $\sqsubseteq_{X|_U}$ of the partial order of X to U. Doing so one has indeed a full embedding I : **PoTop** $\hookrightarrow$ **pStrm**, nevertheless IX might fail to

[1] The terminology is due to Krishnan (2009). It is motivated by the fact that the mapping $\preccurlyeq$ preserves suprema. Hence it can be seen as a cosheaf over the locale of open subsets of the underlying space.

be a circulation. As a counterexample consider any non trivial pospace X (*i.e.* which contains at least two distinct points x and x' such that $x \sqsubseteq_X x'$). Since the underlying space of a pospace is Hausdorff (*cf.* Remark 4.1.2) there exist open subsets U and U' of X such that $x \in U$, $x' \in U'$, and $U \cap U' = \emptyset$. Therefore we have $x \sqsubseteq_{X|_{U \cup U'}} x'$ and by Remark 4.4.2, IX is not a stream. We have actually proven that IX is a stream iff the partial order of X is discrete. In particular given two pospaces X and Y, $\mathbf{Strm}(\cosh IX, \cosh IY)$ may differ from $\mathbf{PoTop}(X, Y)$.

Example 4.4.9. Consider the directed line $\mathbb{R}$ as a pospace, the cosheafification of $I\mathbb{R}$ is the stream described in Example 4.4.4.

Remark 4.4.10. Finding a canonical functor from **LpoTop** to **pStrm** is not so simple because an atlas $\mathcal{U}$ may not be a precirculation and even if it is so, there can be an equivalent (but different) atlas that is not. In particular we have seen that if an atlas over a Hausdorff space X contains a chart over X then it derives from a pospace. Given a local pospace X one can consider its greatest atlas (*cf.* Lemma 4.3.10) and then associate every open subset with the preorder generated by all the charts contained in it.

Among all streams, some of which being rather tricky, the following ones are especially well-behaved.

Definition 4.4.11. In the light of Definition 4.1.4, a stream X is said to be **directed by dipaths** or **filled** when for all open subsets $W \subseteq X$ one has $w \preccurlyeq_W w'$ if and only if there is a dipath on W from w to w'. The full subcategory of streams that are directed by the dipaths is denoted by $\mathbf{Strm}_{\mathrm{d}}$. It will play an important role in the comparison with the category of d-spaces.

Remark 4.4.12. A stream X is directed by the dipaths iff the condition

$$w \preccurlyeq_W^X w' \Leftrightarrow \text{there exists a dipath on } W \text{ from } w \text{ to } w'$$

is satisfied for all the elements W of some basis of the topology of UX.

The category **Strm** is actually cocomplete (Krishnan, 2009, p.446), and provided we pay some attention to the notion of substream (Krishnan, 2009, p.455, Definition 3.2.6), it is a framework for directed topology. Therefore we can define, for all cubical sets K, the realization $\upharpoonleft K \downharpoonleft_{\mathbf{Strm}}$ of the cubical set K in **Strm**. Since the forgetful functor $U : \mathbf{Strm} \to \mathbf{Haus}$ has both left and a right adjoint, the underlying space of $\upharpoonleft K \downharpoonleft_{\mathbf{Strm}}$ is $|K|_{\mathbf{Haus}}$.

Example 4.4.13. We recall the cubical set described in Remark 4.3.36. It is generated by a single square $\square_2$ and the relations

$$K(\delta^1_{0,0})(s) = K(\delta^1_{0,1})(s) \quad \text{and} \quad K(\delta^1_{1,0})(s) = K(\sigma^0_0 \delta^0_{0,0} \delta^1_{1,0})(s)$$

The **Strm**-realization $\upharpoonleft K \downharpoonleft_{\mathbf{Strm}}$ of the cubical set K can be described as follows. Its underlying space is the compact unit disk $D := \{z \in \mathbb{C} \mid |z| \leqslant 1\}$. Then we guess that a dipath of $\upharpoonleft K \downharpoonleft_{\mathbf{Strm}}$ is a path on D of the form $t \in [0, r] \to \rho(t)e^{i\theta(t) \in \mathbb{C}}$ with ρ and θ being nondecreasing continuous paths on $\mathbb{R}_+$ and $[0, 1]$. The circulation of $\upharpoonleft K \downharpoonleft_{\mathbf{Strm}}$ is therefore the cosheafification of the precirculation which associates an open subset $U \subseteq D$ with the preorder

$$\left\{(z, z') \in \mathbb{C} \times \mathbb{C} \;\middle|\; \text{there exists a dipath } \delta \text{ from } z \text{ to } z' \text{ s.t. } \mathrm{img}(\delta) \subseteq U\right\}$$

If we add the relation $K(\delta^1_{1,1})(s) = K(\sigma^0_0\delta^0_{0,1}\delta^1_{1,1})(s)$, then the underlying space of $|K|_{\mathbf{Strm}}$ is the Riemann sphere $\mathbb{S}^2$ identified with $\mathbb{C} \cup \{\infty\}$, the dipaths being defined as above except that ρ is allowed to take its values in $\mathbb{R}_+ \cup \{\infty\}$. Remark that the preceding realizations actually belong to $\mathbf{Strm}_{\mathrm{d}}$.

Apart from the fact that cosheafification is unavoidable when one deals with streams, the category **Strm** is a nice context for directed topology. Yet we will see in Section 4.5 that for our purpose, $\mathbf{Strm}_{\mathrm{d}}$ is even more convenient than **Strm**. One can even work in a full sub-category of $\mathbf{Strm}_{\mathrm{d}}$ that is Cartesian closed as shown in Goubault-Larrecq (2014).

4.5 D-spaces

As control flow graphs for sequential programs (*cf.* the second section of Introduction), the topological model of a parallel program is supposed to be an overapproximation of its collection of execution traces. To every such trace indeed corresponds a dipath on the model. It is therefore tempting to focus on dipaths and define a framework in which they would be the primary datum instead of being the by-product of some (local) (pre)order. There is also a mathematical argument that pleads for such an approach: the category **Cub** as well as all the realizations in **PoTop**, **LpoTop**, and **Strm** seem to belong to $\mathbf{PoTop}_{\mathrm{d}}$ (*cf.* Definition 4.1.4), $\mathbf{LpoTop}_{\mathrm{d}}$ (*cf.* Definition 4.3.31), and $\mathbf{Strm}_{\mathrm{d}}$ (*cf.* Definition 4.4.11) respectively.

In the belief that the notion of dipath should be primary, Marco Grandis has come to introduce the d-spaces – Grandis (2003, 2009). This approach leads to a formalism that is much more tractable and intuitive than the preceding ones. However, being much more supple it embraces many pathological specimens. So we will strengthen the axioms of the original definition to match our needs. As for local pospaces and streams, we suppose that the underlying space of a d-space is Hausdorff.

We slightly differ from the definition of (Grandis, 2003, p.284) in that we allow paths to be defined on any non empty compact interval (including singletons) instead of just $[0, 1]$. As one can imagine, it does not make any significant difference but fits better with Definition 4.2.15.

Definition 4.5.1. A **d-space** consists of a Hausdorff space X together with a set dX of Moore paths (the **d-paths**) such that

1. Every constant path is a d-path.
2. Any subpath of a d-path path is a d-path.
3. The concatenation of two d-paths is a d-path.

The set dX is called the **direction**. A d-space morphism $f : X \to Y$, also called **d-map**, is a direction preserving continuous map (*i.e.* $f \circ dX \subseteq dY$). The d-spaces and d-maps form the category **dTop** as defined by Marco Grandis (2003). Following the notation introduced in Section 4.2, I_r refers to the compact interval $[0, r]$ together with all the nondecreasing continuous paths on it. The next lemma states that the notion of d-path is in line with the notion of dipath (*cf.* Definition 4.2.12).

Lemma 4.5.2. Given a d-space X, we have

$$\mathbf{dTop}(I_r, X) = \big\{\delta \in dX \;\big|\; \delta \text{ is defined over } [0, r]\big\}$$

Proof. Let δ be in $\mathbf{dTop}(I_r, X)$. The mapping $\mathrm{id}_{[0,r]}$ is, by definition, a d-path of I_r. Therefore the direction preserving map δ (seen as $\delta \circ \mathrm{id}_[0, r]$) belongs to dX. Conversely, let δ be an element of dX defined over $[0, r]$ and let γ be a nondecreasing continuous map on $[0, r]$. Then $\delta \circ \gamma$ is a subpath of δ which belongs to dX by the third axiom of Definition 4.5.1. Therefore δ preserves directions. □

Remark 4.5.3. The collection of constant paths and the collection of all paths form two directions on any Hausdorff space X. The resulting d-spaces are respectively called the **discrete** and the **chaotic** d-spaces on X, and they induce the left adjoint and the right adjoint to the forgetful functor $U : \mathbf{dTop} \to \mathbf{Haus}$ which is faithful.

Lemma 4.5.4. Let X be a topological space. The collection of directions (resp. filled directions) on X is a complete lattice. In particular given a family P of paths on X there is a least direction (resp. filled direction) on X containing P.

Proof. The intersection of all (filled) directions containing P is still a (filled) direction. □

Corollary 4.5.5. The category of d-spaces is complete and cocomplete.

Proof. By Remark 4.5.3 the underlying spaces of limits and colimits are computed in **Haus**. The direction is then generated by the obvious collection of paths – see Lemma 4.5.4. □

Remark 4.5.6. The forgetful functor over **dTop** has both a left and a right adjoint (*cf.* Remark 4.5.3) so it preserves limits and colimits. As a consequence, the underlying space of the d-space realization of any cubical set is its geometric realization.

Remark 4.5.7. Given a framework for directed topology (C, U, I) we deduce from Definition 4.2.12, Lemma 4.2.13, and Definition 4.2.15 that any object of C equipped with its dipaths is a d-space. So we have mapped every object of C to an object of **dTop**. This mapping is functorial by Definition 4.2.18. Conversely, one easily checks that **dTop** is a framework for directed topology. In some sense, **dTop** contains a copy of any framework for directed topology.

Remark 4.5.8. Given a d-space X and a topological subspace Y of UX, the collection of d-paths of X whose images lie in Y forms a direction on Y. The d-spaces obtained this way prove that the forgetful functor $U : \mathbf{dTop} \to \mathbf{Haus}$ satisfies the fourth axiom of Definition 4.2.2. They are the d-subspaces of X.

Example 4.5.9. The canonical functor I from **PoTop** to **dTop** sends any pospace X to the d-space on UX (*i.e.* the underlying space of X) whose direction is the collection of dipaths on X. Note that if the underlying space of X is totally disconnected (*e.g.* Priestley spaces (Davey and Priestley, 2002, p.258)) then the d-space IX is discrete regardless of the partial order on X. However the composite $\mathbf{Cub} \hookrightarrow \mathbf{PoTop} \to \mathbf{dTop}$ is a full embedding that preserves pastings of cubes (*cf.* Remark 4.1.21). The category **dTop** is actually a directed framework for directed topology. Note that the same construction applies to **LpoTop**.

We are going to exhibit a full subcategory of **dTop** that is isomorphic to $\mathbf{Strm}_{\mathrm{d}}$.

Remark 4.5.10. As noted by Sanjeevi Krishnan (2009), d-spaces and streams are related by a pair of adjoint functors $S \dashv D$

$$\begin{tikzcd}\mathbf{Strm} \arrow[r, shift left, "D"] & \mathbf{dTop} \arrow[l, shift left, "S"]\end{tikzcd}$$

Given a stream X the collection of d-paths of the d-space DX is

$$\bigcup_{r \in \mathbb{R}_+} \mathbf{Strm}(I_r, X)$$

with I_r being the interval $[0, r]$ equipped with its standard stream structure. Conversely, given a d-space X the circulation of the stream SX is

$$U \text{ open subset } \mapsto \{(x, y) \in U \times U \mid \exists \delta \in dX; \partial^{-}\delta = x; \partial^{+}\delta = y; \operatorname{img}(\delta) \subseteq U\}$$

Denoting by U both forgetful functors $\mathbf{Strm} \to \mathbf{Top}$ and $\mathbf{dTop} \to \mathbf{Top}$ and by η and ε the unit and the counit of the adjunction, the following holds for all streams (resp. d-spaces) X

$$U(\eta_X) = \mathrm{id}_X \quad \text{and} \quad U(\varepsilon_X) = \mathrm{id}_X$$

Remark 4.5.11. The composite of the functor $\mathbf{LpoTop} \to \mathbf{dTop}$ described in Example 4.5.9 followed by S provides a natural way to embed **Cub** into **Strm**.

Proposition 4.5.12. $D \circ S \circ D = D$ and $S \circ D \circ S = S$

Proof. Let X be a stream. The underlying map of $\big(D * \varepsilon_X\big) : DSD(X) \to D(X)$ is id_{UX} so any d-path on $DSD(X)$ is a d-path on $D(X)$. Conversely, let δ be a d-path on $D(X)$ and W be an open subset of X. Suppose that $t \leqslant t'$ with $t, t' \in \operatorname{dom}\delta$ and $[t, t'] \subseteq \delta^{-1}(W)$. Consider a subpath $\delta \circ \theta$ of δ with $\theta : [0, 1] \to [t, t']$ being nondecreasing, continuous, and surjective. Then $\delta \circ \theta$ is a d-path of $D(X)$ satisfying $\operatorname{img}(\delta \circ \theta) \subseteq W$, $\delta \circ \theta(0) = \delta(t)$ and $\delta \circ \theta(1) = \delta(t')$, hence by definition $\delta(t) \preccurlyeq_W^{SD(X)} \delta(t')$. It follows that δ is a d-path on $DSD(X)$.

Let X be a d-space and let W be an open subset of UX, the underlying map of $\big(S * \eta_X\big) : S(X) \to SDS(X)$ is id_X so $w \preccurlyeq_W^{S(X)} w'$ implies that $w \preccurlyeq_W^{SDS(X)} w'$ for all $w, w' \in W$. Conversely, if $w \preccurlyeq_W^{SDS(X)} w'$ then we have a d-path γ of $DS(X)$ such that $\operatorname{img}(\gamma) \subseteq W$, $\partial^{-}\gamma = w$ and $\partial^{+}\gamma = w'$. In particular γ is a stream morphism so we have $w = \partial^{-}\gamma \preccurlyeq_W^{S(X)} \partial^{+}\gamma = w'$. □

Remark 4.5.13. It is worth noticing that neither D nor S are full. First we treat the case of D by considering the set of rational numbers $\mathbb{Q}$ equipped with the stream structure inherited from the real line $\mathbb{R}$. The elements of $\mathbf{Strm}(\mathbb{Q}, \mathbb{Q})$ consist of the continuous nondecreasing maps from $\mathbb{Q}$ to $\mathbb{Q}$ while $\mathbf{dTop}(D(\mathbb{Q}), D(\mathbb{Q}))$ contains all the continuous maps from $\mathbb{Q}$ to $\mathbb{Q}$ since the only directed paths on $\mathbb{Q}$ are the constant ones. We come to the case of S. Borrowing the example from (Krishnan, 2009, p.459, Example 4.3), we define a **staircase** on $\mathbb{R}^2$ as a mapping $\gamma : I \to \mathbb{R}^2$ that can be written as a finite concatenation $\gamma_n \cdots \gamma_1$ with $\mathrm{pr}_0 \circ \gamma_k$ or $\mathrm{pr}_1 \circ \gamma_k$ being constant for all $k \in \{1, \ldots, n\}$ (pr_0 and pr_1 being the projections $\mathbb{R}^2 \to \mathbb{R}$). The direction generated by the staircases turns the plane $\mathbb{R}^2$ into a d-space denoted by X. One observes that the map $f : (x, y) \in \mathbb{R}^2 \mapsto (x + y, x + y) \in \mathbb{R}^2$ belongs to $\mathbf{Strm}(SX, SX)$ since SX is the plane $\mathbb{R}^2$ with the stream structure induced by the standard order, but it does not belong to $\mathbf{dTop}(X, X)$ since it does not preserve the staircases.

In spite of Remark 4.5.13 the functors S and D are almost inverse isomorphisms. First note that, as a consequence of Proposition 4.5.12, a stream (resp. a d-space) X can be written as SY (resp. DY) for some d-space (resp. stream) Y iff $X = SDX$ (resp. $X = DSX$). Then write $D(\mathbf{Strm})$ and $S(\mathbf{dTop})$ for the full subcategories of **dTop** and **Strm** whose collections of objects are respectively

$$\big\{D(X) \bigm| X \text{ object of } \mathbf{Strm}\big\} \qquad \text{and} \qquad \big\{S(X) \bigm| X \text{ object of } \mathbf{dTop}\big\}$$

Definition 4.5.14. A **pseudo d-path** is a path γ such that for all open subsets U, and all $[t, t'] \subseteq \gamma^{-1}(U)$ with $t \leqslant t'$, there exists $\delta \in dX$ such that $\partial^{-}\delta = \gamma(t)$, $\partial^{+}\delta = \gamma(t')$, and $\mathrm{img}(\delta) \subseteq U$. A d-space is said to be **filled** when all its pseudo d-paths are d-paths. We denote the full subcategory of filled d-spaces by $\mathbf{dTop}_{\mathrm{f}}$.

The fact that a d-space is filled has numerous consequences.

Proposition 4.5.15. Given a filled d-space X, the collection of d-paths whose codomain is a given compact interval $[0, r]$ is a closed subset of $X^{[0,r]}$ – equipped with the compact open topology.

Example 4.5.16. The collection of staircases of the plane (*i.e.* paths obtained as a finite concatenation of vertical and horizontal paths – see Remark 4.5.13) provides $\mathbb{R}^2$ with a direction which is not filled, but the least filled direction that contains it is the collection of all nondecreasing paths.

Remark 4.5.17. One can check that $\mathbf{dTop}_{\mathrm{f}}$ is a directed framework for directed topology (*cf.* Definition 4.2.2) and that the $\mathbf{dTop}_{\mathrm{f}}$-realization (*cf.* Definition 2.2.1) behaves nicely, see Example 4.5.18.

Example 4.5.18. Consider the unit square $[0, 1]^2$ with all the nondecreasing paths as direction. Then identify the points $(0, t)$ and $(1, t)$ for all $t \in [0, 1]$, as well as all the points of the lower edge $[0, 1] \times \{0\}$ with the origin. The resulting colimit exists in both **dTop** and $\mathbf{dTop}_{\mathrm{f}}$ – see Corollary 4.5.5 and Theorem 4.5.21. In the latter case it is, up to isomorphism, the compact unit disk

$$\big\{z \in \mathbb{C} \bigm| |z| \leqslant 1\big\}$$

endowed with the direction

$$\big\{\rho \cdot e^{i\theta} \bigm| \rho : [0, r] \to [0, 1] \text{ and } \theta : [0, r] \to \mathbb{R} \text{ nondecreasing continuous maps}\big\}$$

In the former case the direction is restricted to the paths such that for all segments s from the origin to the border of the disk, $\delta^{-1}(s)$ has finitely many connected components. In particular the path $t \in [0, 1] \mapsto t \cdot e^{\frac{i2\pi}{t}}$ is directed in the latter case, not in the former one – see Figure 4.7. Letting ρ take its values in $\mathbb{R}_{+}$ instead of $[0, 1]$ we recognize the direction over $\mathbb{C}$ introduced in Example 4.3.39. The resulting (filled) d-space is called the **directed complex plane**. Adding to $\mathbb{C}$ a point at infinity ∞ and letting ρ take its values in $\mathbb{R}_{+} \cup \{\infty\}$, we obtain the **directed Riemann sphere** . As in Example 4.5.16, the direction generated by the paths of the form $t \mapsto \rho e^{it}$ and $t \mapsto te^{i\theta}$ with ρ and θ constants ranging through $\mathbb{R}_{+}$ and $\mathbb{R}$, is strictly contained in the directed complex plane (resp. the directed Riemann sphere). Moreover the collection of d-paths of the complex plane (resp. the directed Riemann sphere) that crosses $\mathbb{R}_{+}$ (resp. $\mathbb{R}_{+} \cup \{\infty\}$) finitely many times is strictly between the two other ones.

The previous construction is of course motivated by the realization of the cubical set described in Example 4.4.13. We observe that one might have $\rceil K \lfloor_{\mathbf{dTop}} \not\cong \rceil K \lfloor_{\mathbf{dTop}_{\mathrm{f}}}$ even in a simple situation.

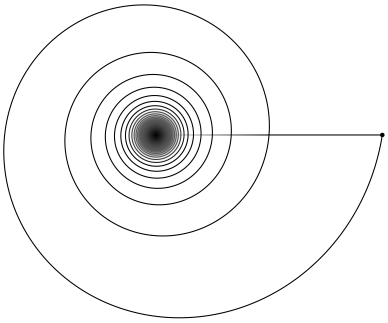


Figure 4.7: The downward spiral

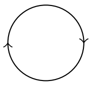


Figure 4.8: The directed circle

Remark 4.5.19. Example 4.5.18 reveals that the realization of cubical sets in $\mathbf{dTop}_\mathrm{f}$ is closer to the intuition than its counterpart in **dTop**. Yet, following his original definition Marco Grandis defines the directed homology and explores its relation to noncommutative geometry – see (Grandis, 2009, Chap.2), which seems to be impossible with filled d-spaces.

Example 4.5.20. The circle $\mathbb{S}^1 = \{e^{i\theta} \mid 0 \leq \theta < 2\pi\}$ inherits a filled d-space structure from the one described in Example 4.5.18 (*i.e.* the d-paths can be written as $p(t) = e^{i\theta(t)}$ for some nondecreasing function θ from some compact interval to $\mathbb{R}$), see Figure 4.8. Note that $\rceil K \lfloor_{\mathbf{dTop}} \cong \rceil K \lfloor_{\mathbf{dTop}_\mathrm{f}}$ with K being the precubical set with a single vertex and a single arrow.

The next theorem highlights the importance of filled d-spaces and streams that are directed by the dipaths. More generally, the notion of pospaces directed by the dipaths (*cf.* Definition 4.1.4) has been extended to locally ordered spaces (*cf.* Definition 4.3.31) and streams (*cf.* Definition 4.4.11 and Remark 4.4.12). Theorem 4.5.21 thus establishes, through the fact that any object of a framework for directed topology can be seen as a d-space (*cf.* Remark 4.5.7), a link between the mathematical objects that are directed by the dipaths (*e.g.* (local) pospaces and streams) and their counterparts in the framework for directed topology $\mathbf{dTop}_\mathrm{f}$. It can also be seen as a characterization of those d-spaces whose direction is locally generated.

Theorem 4.5.21 (Haucourt (2012), see Figure 4.9)**.** The categories $\mathbf{dTop}_\mathrm{f}$ and $\mathbf{Strm}_\mathrm{d}$ are respectively $D(\mathbf{Strm})$ and $S(\mathbf{dTop})$. Moreover the functors S and D induce a pair of (mutually) inverse functors between them. Furthermore $\mathbf{dTop}_\mathrm{f}$ (resp. $\mathbf{Strm}_\mathrm{d}$) is a complete and cocomplete mono and epi reflective (resp. coreflective) subcategory of **dTop** (resp. **Strm**).

$$\begin{tikzcd}
\mathbf{Strm} \arrow[rr, shift left, "D"] \arrow[rr, phantom, "\top"] \arrow[dd, shift left, "\text{corefl}"] & & \mathbf{dTop} \arrow[ll, shift left, "S"] \arrow[dd, shift right, "\text{refl}"'] \\
\arrow[r, phantom, "\dashv"] & & \arrow[r, phantom, "\dashv"] & \\
\mathbf{Strm}_{\mathrm{d}} \arrow[uu, shift left, "\subseteq"] \arrow[rr, shift left, "D"] \arrow[rr, phantom, "\cong"] & & \mathbf{dTop}_{\mathrm{f}} \arrow[ll, shift left, "S"] \arrow[uu, shift right, "\subseteq"']
\end{tikzcd}$$

Figure 4.9: Relating streams and d-spaces

Proof. See (Haucourt, 2012, Sect.5). In particular, $\mathbf{Strm}_{\mathrm{d}}$ and $\mathbf{dTop}_{\mathrm{f}}$ are the codomains of the functors S and D. Thus we have, by corestriction, two functors $S' : \mathbf{dTop} \to \mathbf{Strm}_{\mathrm{d}}$ and $D' : \mathbf{Strm} \to \mathbf{dTop}_{\mathrm{f}}$. Denote by S^{-} and D^{-} the inverses of the functors induced by S and D. Then the reflection and the coreflection are given by $S^{-} \circ S'$ and $D^{-} \circ D'$. □

Remark 4.5.22. Remark 4.5.3, Lemma 4.5.4, and Remark 4.5.6 remain valid for filled d-spaces.

We compare the geometric realizations of cubical sets in $\mathbf{dTop}$, $\mathbf{dTop}_{\mathrm{f}}$, $\mathbf{Strm}$, and $\mathbf{Strm}_{\mathrm{d}}$.

Corollary 4.5.23. For all cubical sets K we have
$D^{-1}(|K|_{\mathbf{dTop}_{\mathrm{f}}}) = |K|_{\mathbf{Strm}_{\mathrm{d}}}$ and $S^{-1}(|K|_{\mathbf{Strm}_{\mathrm{d}}}) = |K|_{\mathbf{dTop}_{\mathrm{f}}}$.

Proof. The functors S^{-1} and D^{-1} are inverse of each other by Theorem 4.5.21. □

Corollary 4.5.24. For all cubical sets K we have $|K|_{\mathbf{Strm}_{\mathrm{d}}} = |K|_{\mathbf{Strm}}$.

Proof. By Theorem 4.5.21 the inclusion functor $I : \mathbf{Strm}_{\mathrm{d}} \hookrightarrow \mathbf{Strm}$ has a right adjoint, hence it preserves colimits and we have $|K|_{\mathbf{Strm}_{\mathrm{d}}} = I(|K|_{\mathbf{Strm}_{\mathrm{d}}}) = |K|_{\mathbf{Strm}}$. □

Corollary 4.5.25.
For all cubical sets K, we have $S(|K|_{\mathbf{dTop}}) = |K|_{\mathbf{Strm}}$ and $D(|K|_{\mathbf{Strm}}) = |K|_{\mathbf{dTop}_{\mathrm{f}}}$.

Proof. The first equality immediately comes from the fact S preserves colimits (as a left adjoint) and the n-dimensional directed cube in $\mathbf{dTop}$ is sent to its counterpart in $\mathbf{Strm}$. By Corollary 4.5.23 and Corollary 4.5.24 we have

$$|K|_{\mathbf{dTop}_{\mathrm{f}}} = D(|K|_{\mathbf{Strm}_{\mathrm{d}}}) = D(|K|_{\mathbf{Strm}}) .$$

□

Corollary 4.5.26. For all cubical sets K, we have $|K|_{\mathbf{dTop}_{\mathrm{f}}} = |K|_{\mathbf{dTop}}$ if and only if $|K|_{\mathbf{dTop}} \in \mathbf{dTop}_{\mathrm{f}}$.

Proof. We have $S(|K|_{\mathbf{dTop}}) = |K|_{\mathbf{Strm}}$ by Corollary 4.5.25. If $|K|_{\mathbf{dTop}} \in \mathbf{dTop}_{\mathrm{f}}$ then by Theorem 4.5.21 we deduce that $|K|_{\mathbf{dTop}} = D(|K|_{\mathbf{Strm}})$. By Corollary 4.5.25 it comes

$$|K|_{\mathbf{dTop}} = D(|K|_{\mathbf{Strm}}) = |K|_{\mathbf{dTop}_{\mathrm{f}}} .$$

□

We claim that the difference between $\rceil K\lfloor_{\mathbf{dTop}}$ and $\rceil K\lfloor_{\mathbf{dTop_f}}$ that one may observe is entirely due to vortices (*cf.* Example 4.5.18). From Corollary 4.3.38, it is thus natural to extend Conjecture 4.3.44 as follows.

Conjecture 4.5.27. For all precubical sets K, we have

$$I(\rceil K\lfloor_{\mathbf{LpoTop}}) \cong \rceil K\lfloor_{\mathbf{dTop}} \cong \rceil K\lfloor_{\mathbf{dTop_f}}$$

with I : **LpoTop** $\hookrightarrow$ **dTop** being the "inclusion" functor described in Example 4.5.9.

A special instance of Conjecture 4.5.27 is given by Proposition 6.1.7. The next definition is borrowed from (Fahrenberg and Raußen, 2007, p.20, Definition 4.3).

Definition 4.5.28. A **reparametrization** is a non-decreasing continuous map of a compact interval onto another. A reparametrization might not be one-to-one but it is surjective. By extension a reparametrization of a Moore path γ is a composite of the form $\gamma \circ \theta$ where θ is a reparametrization. A d-space is said to be **saturated** when any path on its underlying space is a d-path as soon as it admits a reparametrization that is a d-path.

The relevance of saturated d-spaces appears when one studies the compact-open topology over the set of d-paths of a d-space, and then defines the notion of trace space – see Raußen (2009b) and Section 5.6. It is actually not so easy to exhibit a nonsaturated d-space. The example we provide heavily relies on results from (Fahrenberg and Raußen (2007)).

Example 4.5.29. An interval $[a, b]$ with $a < b$ is a stop-interval of a path γ when the restriction of γ to $[a, b]$ is constant and $[a, b]$ is maximal with this property. A stop-value of γ is an element v of its codomain such that the interior of $\gamma^{-1}(\{v\})$ is nonempty. As a consequence of (Fahrenberg and Raußen, 2007, Lemma 2.10 and Corollary 4.11), any countable subset of $\mathbb{R}^2$ on which the order induced by $\mathbb{R}^2$ is total, is the set of stop-values of some increasing path on $\mathbb{R}^2$. Then take $\mathbb{R}^2$ as underlying space and say an increasing path on $\mathbb{R}^2$ is directed when its set of stop-values is dense in its image. The resulting directed space is not saturated.

The pathology described in Example 4.5.29 never happens in with filled d-spaces:

Lemma 4.5.30. Any filled d-space is saturated.

Proof. Suppose that $\gamma \circ \theta$ is directed and let U be an open set such that $\gamma([t, t']) \subseteq U$. Since θ is a reparametrization, there is s, s' such that $\theta(s) = t$ and $\theta(s') = t'$, and therefore $\theta([s, s']) = [t, t']$. The restriction of the directed path $\gamma \circ \theta$ to $[s, s']$ is thus directed and (has its image) contained in U. □

The converse is false as shown by the example of increasing staircase paths on $\mathbb{R}^2$. I also wanted to mention the approach of Hirschowitz et al. (2013) which defines an alternative notion of saturation and thus provides a full subcategory of **dTop** which rules out many of its pathologies but enjoys its "good" properties.

Another important feature of the d-spaces that one meets in nature (*cf.* Proposition 4.5.34) is the following.

Name	Objects	Reference
dTop	d-spaces	Definition 4.5.1
$\mathbf{dTop}_{\mathrm{f}}$	filled d-spaces	Definition 4.5.14
$\mathbf{dTop}_{\mathrm{s}}$	saturated d-spaces	Definition 4.5.28
$\mathbf{dTop}_{\mathrm{c}}$	complete d-spaces	Definition 4.5.31
$\mathbf{dTop}_{\mathrm{cf}}$	complete filled d-spaces	Definition 4.5.31

Figure 4.10: Full subcategories of **dTop**

Definition 4.5.31. A d-space X is said to be **complete** when for all d-maps $\delta : \mathbb{R} \to X$, if both following limits exist then δ extends to a d-map $\overline{\delta} : \mathbb{R} \cup \{\text{-}\infty, \text{+}\infty\} \to X$.

$$\lim_{t\to-\infty} \delta(t) \quad \text{and} \quad \lim_{t\to+\infty} \delta(t)$$

We denote by $\mathbf{dTop}_{\mathrm{c}}$ (resp. $\mathbf{dTop}_{\mathrm{cf}}$) the full subcategory of **dTop** (resp. $\mathbf{dTop}_{\mathrm{f}}$) whose objects are complete.

Remark 4.5.32. The staircases of the plane Remark 4.5.13 do not form a complete d-space (*e.g.* consider an infinite staircase whose n^{th} step is of size $\frac{1}{2^n}$).

One easily checks that Lemma 4.5.4 is still valid for complete (resp. complete filled) directions so $\mathbf{dTop}_{\mathrm{f}}$ and $\mathbf{dTop}_{\mathrm{cf}}$ are both complete and cocomplete.

Example 4.5.33. The embedding of **Cub** into **dTop** (*cf.* Example 4.5.9) is actually an embedding in $\mathbf{dTop}_{\mathrm{cf}}$ which is indeed a directed framework for directed topology.

The list of full subcategories of **dTop** we have introduced so far is summarized in Figure 4.5. The rest of this section is a list of remarks that advocate the use of $\mathbf{dTop}_{\mathrm{cf}}$ as a reasonable model of directed topology, although all its interesting objects come from $\mathbf{PoTop}_{\mathrm{d}}$, $\mathbf{LpoTop}_{\mathrm{d}}$, or $\mathbf{Strm}_{\mathrm{d}}$. The next result sharpens Example 4.5.9 in which the functor I : **LpoTop** → **dTop** is defined.

Proposition 4.5.34. The collection of dipaths of a local pospace induces a complete filled d-space.

Proof. Suppose that δ is a local pospace morphism from the directed line to X and that x, the limit of δ at $+\infty$, exists. Let U be a chart containing x, there exists t_0 such that $\delta([t_0, +\infty[) \subseteq U$. Since U is a pospace, x is actually the least upper bound of $\delta([t_0, +\infty[)$ – see (Nachbin, 1965, p.26, Proposition 1). Therefore δ extends to a local pospace morphisms defined over $\mathbb{R}\cup\{-\infty, +\infty\}$. Let γ be a pseudo d-path of IX (*i.e.* the d-space induced by X) and consider $t \in \operatorname{dom}\gamma$. Let U be a chart of X containing $\gamma(t)$, and let J be an open interval of $\operatorname{dom}\gamma$ containing t and such that $\gamma(J) \subseteq U$. Given $t', t'' \in J$ there exists a dipath on X from $\gamma(t')$ to $\gamma(t'')$ whose image is contained in U because γ is a pseudo d-path, hence $\gamma(t') \sqsubseteq_U \gamma(t'')$. Therefore γ is a local pospace morphism that is to say a d-path on IX. □

Remark 4.5.35. Let $\delta : \mathbb{R}_+ \to X$ be a morphism of filled d-spaces whose limit at $+\infty$ exists and denote it by x. Also suppose that x admits a neighborhood from any point of which x can be reached along a directed path. Then δ extends to a d-space morphism $\overline{\delta} : \mathbb{R} \cup \{+\infty\} \to X$. This is precisely the situation that we have met in Example 4.5.18.

Remark 4.5.36. The complete d-spaces (*cf.* Definition 4.5.31) are also better adapted to d-spaces extensions. Suppose that one is given two d-spaces X and Y such that $UX \subseteq UY$ and dY is the least direction that contains dX and makes the inclusion $UX \hookrightarrow UY$ a d-map. Then one has

$$dY = dX \cup \{\text{constant paths on } UY\} \ .$$

In this setting the only d-paths starting from or arriving at $UY \setminus UX$ are the constant ones. The same remark applies if we work with filled d-spaces instead. On the contrary if we consider complete d-spaces, then any path from a point of X to a point of Y whose strict initial segments are d-paths, is a d-path itself. From this point of view, the direction of a d-space extension should (and will unless otherwise stated) at least contains all the paths from a point of X (resp. Y) to a point of Y (resp. X) whose proper initial (resp. final) segments are d-paths.

Among the plethora of extensions of a topological space, the **compactifications** (*i.e.* the compact extensions in which the space is dense (Kelley, 1955, p.151)) are of special interest.

Definition 4.5.37. A **compactification** of a complete d-space X is a complete compact d-space Y such that UX is dense in UY and the direction on Y is the least complete one that makes the inclusion $X \hookrightarrow Y$ a d-space morphism.

Example 4.5.38. Following Definition 4.5.37, the directed compact unit interval and the directed circle are two compactifications of the directed open unit interval.

Example 4.5.39. There are many ways to compactify a topological space. The **Stone-Čech** compactification is one of the most familiar to category theorists because it can be defined as the left adjoint β to the inclusion functor **CHaus** $\hookrightarrow$ **Top** (Borceux, 1994a, p.113, 3.3.9c) or (Johnstone, 1982, p.131). Indeed the celebrated Tychonoff theorem (Kelley, 1955, p.143) precisely states that **CHaus** $\hookrightarrow$ **Top** preserves products. It also preserves equalizers since any closed subspace of a compact Hausdorff space is compact Hausdorff. In a more explicit way, if we denote by $C^*(X)$ the set of bounded real-valued functions defined over X, and by I_f the closure of img(f) for each such f, then we have the evaluation map at X

$$ev_X \quad : \quad x \in X \quad \mapsto \quad (f(x))_{f \in C^*(X)} \in \prod_{f \in C^*(X)} I_f \ ,$$

and βX is the closure of the image of ev_X while the unit of the adjunction at X is the induced map $X \longrightarrow \beta X$. The latter is an embedding (*i.e.* a homeomorphism on its image) iff X is a Tychonoff space (*i.e.* the singletons are closed and for all closed subsets C and all $x \in X \setminus C$ there exists $f \in C^*(X)$ such that $f(C) = 1$ and $f(x) = 0$).

Remark 4.5.40. The Stone-Čech theorem (Kelley, 1955, p.153) states that if X is a Tychonoff space, then any continuous map from X to some compact Hausdorff space can be extended to βX. As a consequence $\beta(]0,1[)$ is not homeomorphic with $[0,1]$ because no bounded function oscillating at the boundary (*e.g.* $t \mapsto \frac{1}{t(1-t)}$) admits a continuous extension over $[0,1]$. In fact the cardinalities of the underlying sets of $\beta\mathbb{R}$ and $\beta\mathbb{R} \setminus \mathbb{R}$ are $2^{2^{\aleph_0}}$ (Willard, 1970, p.141).

Example 4.5.41. One of the most common and intuitive compactifications is the **Alexandroff** one (Kelley, 1955, p.150). Given a space X, it consists of adding a point at infinity. Formally we denote the additional point by ∞ and define the open subsets of $X \cup \{\infty\}$ as the subsets that are open in X or equal to the complement of some closed compact subset of X. Note in particular that if X is compact, then ∞ is isolated. Then it is a well know fact that $X \cup \{\infty\}$ is Hausdorff iff X is locally compact and Hausdorff (Kelley, 1955, p.150).

Definition 4.5.42. A continuous map $f : X \to Y$ is said to be **continuous at infinity** when there exists $y \in Y$ such that for all neighborhoods W of y there exists a closed compact subset K of X such that $f(X \setminus K) \subseteq W$.

Note that if X is compact Hausdorff then any map defined over X is continuous at infinity. Also note that if Y is not Hausdorff, there might be several possible y satisfying the requirement of Definition 4.5.42. We denote by $\mathbf{Haus}_\infty$ the subcategory of **Haus** whose morphisms are continuous at infinity, the embedding $\mathbf{CHaus} \hookrightarrow \mathbf{Haus}_\infty$ admits a left adjoint α as which is defined as follows:

– if X is compact then $\alpha X = X$,

– otherwise αX is the Alexandroff compactification of X.

Example 4.5.43. The underlying space of the Alexandroff compactification of $\mathbb{R}^n$ is $\mathbb{S}^{n+1}$. While the topology of $\mathbb{R}^n \cup \{\infty\}$ is easily understood, its direction is much more intricate. In particular ∞ is its unique vortex (*cf.* Definition 4.3.37) and therefore $\mathbb{R}^2 \cup \{\infty\}$ it is not isomorphic to the directed Riemann sphere described in Example 4.4.13.

A refinement of the Alexandroff compactification will be described in Section 6.1 to state Theorem 6.1.20 and Theorem 6.1.42.

In the remaining of the section, we establish a relation between vector fields and directions. Many of the d-spaces we have seen so far are indeed carried by **smooth manifolds** (*i.e.* the ones with C^∞ transition maps) instead of mere topological spaces. The directions of these specimens derive from vectors fields in a way that we subsequently describe. Once again we will notice that the complete filled d-spaces fit better with examples in nature. Informally speaking, something is said to be smooth when derivation operators can be applied to it as many times as one wishes. Following (Bishop and Goldberg, 1980, p.43, Section 1.5) we define a smooth curve as a map of an interval of real numbers into a smooth manifold such that there is an extension to an open interval which is a smooth map. Since we only consider smooth manifolds and smooth maps between them we omit the adjective "smooth". On the contrary we write "continuous map" to stress that the map is merely continuous but not necessarily smooth or even derivative. In particular we keep the word "path" to mean continuous path. On the contrary the word "curve" always refers to a smooth map. The **tangent bundle** of an n-dimensional **manifold** $\mathcal{M}$ is the $2n$-dimensional manifold which gathers all the tangent spaces $T_x\mathcal{M}$ with x ranging through $\mathcal{M}$. It comes with the canonical map $p : T\mathcal{M} \to \mathcal{M}$. As suggested by its name, the tangent bundle is a **vector bundle** – see (Lang, 1999, Chapter III). A **vector field** on $\mathcal{M}$ is defined as a section of p (*i.e.* a map $s : \mathcal{M} \to T\mathcal{M}$ such that $p \circ s = \mathrm{id}$). The mapping s thus associates to every point x of $\mathcal{M}$ a vector $s(x)$ that is tangent to $\mathcal{M}$ at x. These vectors can be seen as arrows therefore providing a direction. This intuition is made formal by a classical theorem which claims that for every point $x_0 \in \mathcal{M}$ and every sufficiently small $\varepsilon > 0$ there exists a unique curve $\gamma :]-\varepsilon, +\varepsilon[\to \mathcal{M}$ such that $\gamma(0) = x_0$ and

$$\frac{\partial \gamma}{\partial t}(t) = s(\gamma(t))$$

for all $t \in]-\varepsilon, +\varepsilon[$ (Lang, 1999, Chapter IV). Such a curve is said to be **integral**. It is then natural to consider the collection of curves on compact intervals as the generating set of a d-space over $\mathcal{M}$. The structure of the latter d-space is rather poor since any point of $\mathcal{M}$ is visited by a unique maximal integral curve (*i.e.* that cannot be extended anymore). Their images form a partition of $\mathcal{M}$ so two d-paths that can be appended must lie on the image of the same maximal integral curve. Things become more interesting if one allows several vector fields $s_1, \ldots, s_k$ to generate the direction. Within the spirit of considering, for all point $x \in \mathcal{M}$, the set

$$F_x := \Big\{ \sum_{i=1}^{k} \lambda_i \cdot s_i(x) \,\Big|\, \lambda_i \geqslant 0 \text{ for } i = 1, \ldots, k \Big\}$$

as the forward cone of $\mathcal{M}$ at x, we have

Definition 4.5.44. A curve γ is said to be **forward** (with respect to $s_1, \ldots, s_k$) when its derivative at time t belongs to $F_{\gamma(t)}$ for all $t \in \operatorname{dom}\gamma$:

$$\frac{\partial \gamma}{\partial t}(t) \in F_{\gamma(t)}$$

The d-space generated by the vector fields $s_1, \ldots, s_n$ on the manifold $\mathcal{M}$ is the least direction on $\mathcal{M}$ that contains all the forward curves, it is denoted by $d\mathcal{M}_s$ with s being understood as the set $\{s_1, \ldots, s_k\}$. The closure can also be taken in the lattice of filled (resp. complete, complete filled) directions, in that case the notation is changed into $d\mathcal{M}_s^f$, $d\mathcal{M}_s^c$, or $d\mathcal{M}_s^{cf}$ accordingly.

Remark 4.5.45. In particular if s and s' are two sets of vectors fields on $\mathcal{M}$ we can consider $d\mathcal{M}_s^* \vee d\mathcal{M}_{s'}^*$ with the upper bound being taken in the lattice according to what is put instead of $*$. We have the obvious fact that $d\mathcal{M}_s^* \vee d\mathcal{M}_{s'}^* \subseteq d\mathcal{M}_{s\cup s'}^*$ but the converse inclusion might not be satisfied.

Let us revisit some examples.

Example 4.5.46. The unit circle (embedded in $\mathbb{R}^2$) together with the vector field that associates each point (x, y) with the vector $(-y, x)$ generates the directed circle.

Example 4.5.47. Consider the Euclidean plane $\mathcal{M}$ with its standard manifold structure and the constant vector fields s_v and s_h (which stands for "vertical" and "horizontal") that associates each point of the plan with the vectors $(1, 0)$ and $(0, 1)$ respectively. The maximal integral curves related to s_v and s_h are respectively the mappings $t \in \mathbb{R} \mapsto (t, 0) \in \mathbb{R}^2$ and $t \in \mathbb{R} \mapsto (0, t) \in \mathbb{R}^2$. The direction $d\mathcal{M}_{s_v,s_h}^f$ is the set of all nondecreasing paths while the elements of $d\mathcal{M}_{s_v,s_h}$ are the subpaths of the finite concatenations of nondecreasing curves. Moreover we have

$$d\mathcal{M}_{s_v,s_h}^f \quad = \quad d\mathcal{M}_{s_v}^f \vee d\mathcal{M}_{s_h}^f \qquad \text{but} \qquad d\mathcal{M}_{s_v,s_h} \quad \neq \quad d\mathcal{M}_{s_v} \vee d\mathcal{M}_{s_h}$$

since $d\mathcal{M}_{s_v} \vee d\mathcal{M}_{s_h}$ is the set of staircases of the plane (*cf.* Remark 4.5.13) and any pseudo d-path on $d\mathcal{M}_{s_v}^f \vee d\mathcal{M}_{s_h}^f$ is order preserving (with respect to the product order of $\mathbb{R}^2$).

Example 4.5.48. The Euclidean plane $\mathcal{M}$ can also be equipped with the vector fields s_a and s_r (which stands for "attract" and "rotate") that associate the point (x, y) with the vectors $(-x, -y)$ and $(-y, x)$. The d-paths of $d\mathcal{M}_{s_a,s_r}^f$ are the paths of the form $t \mapsto \rho(t)e^{i\theta(t)}$ for all mappings ρ and θ such that

- θ, ρ are nondecreasing paths (on $\mathbb{R}$) defined over a compact interval, and
- $\rho(t) > 0$ for all t of its domain of definition.

We point out that the only d-paths of $d\mathcal{M}^f_{s_a,s_r}$ going through the origin are the constant ones. Consequently the d-space $d\mathcal{M}^f_{s_a,s_r}$ is isomorphic (in **dTop**) to the disjoint union of the origin and the directed cylinder

$$\{0\} \sqcup \mathbb{S}^1 \times \mathbb{R}$$

This is due to the fact that the origin is a singular point for both vector fields (*i.e.* $s_a(0) = s_r(0) = 0$). In order to obtain the full fledged directed complex plane one has to consider the filled complete direction $d\mathcal{M}^{cf}_{s_a,s_r}$.

Conjecture 4.5.49. Given a set of vector fields $\{s_1, \ldots, s_k\}$ on a manifold $\mathcal{M}$ of dimension $n \geqslant k$ such that for all $x \in \mathcal{M}$ at least one of the vectors $\{s_1(x), \ldots, s_k(x)\}$ is nonzero, $\mathcal{M}$ admits a directed atlas $\mathcal{U}$ (in the sense of Definition 4.3.1) such that for all $U \in \mathcal{U}$ one has $x \sqsubseteq_U x'$ iff there exists a forward curve from x to x' whose image is contained in U. Moreover the resulting local pospace is directed by the dipaths (*cf.* Definition 4.1.4).

Example 4.5.50. A set of vector fields $\{s_1, \ldots, s_n\}$ over a smooth manifold $\mathcal{M}$ is called a **parallelization** (of $\mathcal{M}$) (Bishop and Goldberg, 1980, p.160) when for all points $p \in \mathcal{M}$, the set $\{s_1(x), \ldots, s_n(x)\}$ forms a vector basis of the tangent space $T_x\mathcal{M}$. In this case the manifold $\mathcal{M}$ is said to be **parallelizable** (Bishop and Goldberg, 1980, p.160, Appendix 3B) or (Conlon, 2008, p.97, Definition 3.3.10). A manifold admits at most one parallelization up to isomorphism. More precisely, the **frame bundle** of $\mathcal{M}$ gathers all the groups of automorphisms of the tangent spaces $T_x\mathcal{M}$, for $x \in \mathcal{M}$, in a single manifold denoted by $GL(\mathcal{M})$. As any bundle, it comes with a canonical smooth projection $f : GL(\mathcal{M}) \to \mathcal{M}$. Given g a section of it (*i.e.* a smooth map that sends $x \in \mathcal{M}$ to some automorphism of $T_x\mathcal{M}$) and a vector field s over $\mathcal{M}$, we can define the vector field $g \cdot s$ by $(g \cdot s_i)(x) = g(x)(s_i(x))$ for all $x \in \mathcal{M}$. Then one proves that $GL(\mathcal{M})$ transitively acts on the set of framings of $\mathcal{M}$ in the following sense: given a section g of the frame bundle and a framing $\{s_1, \ldots, s_n\}$ of $\mathcal{M}$, the set $g \cdot \{s_1, \ldots, s_n\} = \{g \cdot s_1, \ldots, g \cdot s_n\}$ is another parallelization of $\mathcal{M}$ and all of them can be recovered that way. As a consequence, the d-space $d\mathcal{M}^*_s$ is uniquely defined up to isomorphism and we write $d^*\mathcal{M}$ when $\mathcal{M}$ is a parallelizable manifold directed by some of its parallelizations. In the same fashion all the local pospaces induced by framings (*cf.* Conjecture 4.5.49) are isomorphic, thus we denote by $L\mathcal{M}$ their isomorphism class in **LpoTop**.

Conjecture 4.5.51. Given a parallelizable manifold $\mathcal{M}$, we denote by $I\mathcal{M}$ the d-space obtained as in Example 4.5.9 from the local pospace $L\mathcal{M}$. For all parallelizable manifolds $\mathcal{M}$, we have

$$I\mathcal{M} \cong d\mathcal{M}^{cf}$$

Example 4.5.52. Every Lie group is parallelizable (*cf.* (Bishop and Goldberg, 1980, p.161)) thus providing a very broad class of examples among which one finds $\mathbb{R}$ and $\mathbb{S}^1$, not to mention $\mathbb{S}^3$ and $\mathbb{S}^7$ which are, together with $\mathbb{S}^0 \cong \{\pm 1\}$ and $\mathbb{S}^1$, the only parallelizable spheres (Bishop and Goldberg, 1980, p.5, Theorem 1.2.13).

Remark 4.5.53. The notion of vector field is related to the control flow structures of Paml programs – see Remark 3.4.5.

4.6 Other Formalisms

This section gathers all the other approaches of directed topology I'm aware of. All of them are more or less related to ideas and tools from model category theory (Quillen (1967); Hovey (1999)).

As an alternative to framework for directed topology (Grandis, 2009, p.98) defines a "good topological setting" for directed topology as a (non-reversible) dIP1-category with all limits and colimits, together with a forgetful strict dI1-functor to **Top** that admits a right adjoint. As for our frameworks for directed topology the underlying idea is that the compact unit interval with its standard order should play in directed algebraic topology the same role as the compact unit segment plays in algebraic topology. The general principles of (Grandis (2009)) ruled out **PoTop** and **LpoTop** since the forgetful functor does not preserve colimits. Note that (Grandis (2009)) refers to the "precirculations" of Krishnan (2009) as "locally preordered spaces".

By analogy with Grothendieck's homotopy hypothesis, which says that ∞-groupoids (*i.e.* $(\infty, 0)$-categories) are spaces, it was suggested that $(\infty, 1)$-categories correspond to directed topology, see (Porter (2015)) for details. The former being notoriously related to topologically enriched categories, it is natural to consider them as potential models for "directed homotopy types", a notion that is not well-defined yet. In the early nineties Philippe Gaucher thus introduced and thoroughly studied the notion of **flow** which is, roughly speaking, a graph G in which the sets of parallel arrows (*i.e.* the ones that have the same source and the same target) are equipped with a topology. Then Gaucher (2003) exhibits a model structure over the category of flows and Gaucher (2008) provides the CCS process algebra with a model category structure whose weak equivalences are tightly related to bisimulations. Going further Gaucher (2010a,b) studies higher dimensional automata/transition systems from a directed topology point of view.

Worytkiewicz (2010) has also defined a model structure from the (small) subcategory of pospaces whose objects are the hypercubes (*i.e.* n-fold products of the compact unit interval) and the morphisms are the faces inclusion and the projections (*i.e.* the copy of $\square$ in **PoTop**). His construction is based on rather advanced tools from model category theory: sheaves, completion, and localization – May and Ponto (2010); Hirschhorn (2003).

Yet another approach has been proposed by Fajstrup and Rosický (2008) in order to obtain a locally presentable category. Indeed they prove that if the functor $U : C \to \mathbf{Set}$ induces a topological fibre-small category (*i.e.* for all set S the collection of objects X of C satisfying $UX = S$ is a set) and $\mathcal{I}$ is a small full subcategory of C, then the category of $\mathcal{I}$-generated objects form a locally presentable category which is coreflective in C and admits $\mathcal{I}$ as a dense subcategory. The result is then applied to **dTop** and its full subcategory of directed hypercubes.

The last two works are motivated by the idea that the directed hypercubes (no matter which category they are seen as an object of) are the building blocks of the topological models of concurrency. While K. Worytkiewicz has applied an extension method that generates a model category from a small one, L. Fajstrup and J. Rosický have restricted a large (and rather "wild") category to a more reasonable one. Yet both are, in some sense, built from hypercubes. Choosing the (pre)cubical sets as combinatorial models for directed topology or concurrency is motivated by the same idea (*cf.* Pratt (1991) and Chapter 2).

5

The Fundamental Category

The fundamental category is to algebraic directed topology as the fundamental groupoid is to algebraic topology. The construction applies to any framework for directed topology (*cf.* Definition 4.2.2). The standard notion of homotopy of paths together with its basic properties are recalled in Section 5.1 while the notion of alternating homotopy (*cf.* Definition 5.2.5) and the related fundamental category construction are detailed in Section 5.2. The fundamental category of an object is compared to the fundamental category of its image under a framework morphism in Section 5.3. Special attention is paid to the realizations of cubical sets. In Section 5.4, we state and prove the directed counterpart of the standard Seifert-van Kampen theorem (Brown, 2006, p.240) in the context of a framework for directed topology. The main application is the computation of the fundamental category of the directed circle. In Section 5.5 we fix a framework for directed topology and compare the enveloping groupoids of the fundamental categories of its objects to the fundamental groupoids of their underlying spaces. A fleeting glimpse at Martin Raußen's trace space theory is given in Section 5.6. The associated construction endows the homsets of certain fundamental categories with a topology and computes its homotopy type.

5.1 Homotopies of Paths and 2-Categories

Let γ and δ be two paths on some space X and defined over a compact interval $[0, r]$ with $r \geqslant 0$ (*i.e.* both are of shape r – see Definition 4.0.1), and sharing the same source and the same target.

Definition 5.1.1. A homotopy h from γ to δ is a continuous map defined over $[0, r] \times [0, q]$ (for some $q \geqslant 0$ the **shape** of the homotopy) such that

$$h(0, s) = \partial^{-}(\gamma) \qquad \text{and} \qquad h(r, s) = \partial^{+}(\gamma) \qquad \text{for all } s \in [0, q]$$

and

$$h(t, 0) = \gamma(t) \qquad \text{and} \qquad h(t, q) = \delta(t) \qquad \text{for all } t \in [0, r]$$

A homotopy of paths can be viewed as a "tile" – seeFigure 5.1. Note that there is an obvious homotopy h from a path γ to itself: just put $h(t, s) = \gamma(t)$. Such a homotopy is said to be constant and denoted by γ without any reference to q. Also note that if h is a homotopy from γ to δ and f is a map then $f \circ h$ is a homotopy from $f \circ \gamma$ to $f \circ \delta$. Homotopies can be vertically composed or "piled up" as illustrated on Figure 5.2.

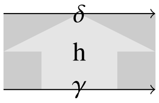


Figure 5.1: Homotopy of paths as “tile”

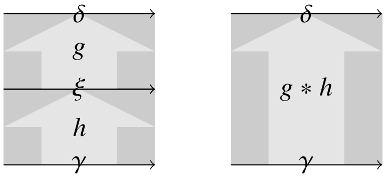


Figure 5.2: Piled up homotopies

Lemma 5.1.2. If h and g are homotopies of shapes q and q' from γ to ξ and from ξ to δ, then γ, ξ and δ have the same shape r and the mapping $g * h$ defined over $[0, r] \times [0, q + q']$ by $g * h(t, s) = h(t, s)$ if $0 \leqslant s \leqslant q$; $g(t, s - q)$ if $q \leqslant s \leqslant q + q'$ is a homotopy from γ to δ.

Homotopies can also be horizontally composed or “juxtaposed” as illustrated on Figure 5.3.

Lemma 5.1.3. If h and h' are homotopies of the same shape q from γ to δ and from γ' to δ' with $\partial^+(\gamma) = \partial^-(\gamma')$, then the following mapping $h' \cdot h$ defined over $[0, r+r'] \times [0, q]$ by $h' \cdot h(t, s) := h(t, s)$ if $0 \leqslant t \leqslant r$; $h'(t, s)$ if $r \leqslant t \leqslant r + r'$ is a homotopy from $\gamma' \cdot \gamma$ to $\delta' \cdot \delta$.

Remark 5.1.4. The usual Godement exchange law applies: it doesn’t matter if we put homotopies side-by-side or stack them up first, the resulting homotopy is the same. Formally, if we have the situation depicted on Figure 5.4 then we have

$$(g' * h') \cdot (g * h) = (g' \cdot g) * (h' \cdot h)$$

Remark 5.1.5. Homotopies and natural transformations enjoy constructions satisfying similar rules. These analogies are related to the concept of 2-categories – see (Mac Lane, 1998, p.272) or (Borceux, 1994a, p.281), the standard example of which being provided by categories, functors, and natural transformations – see Figure 5.5.

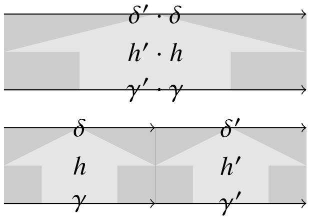


Figure 5.3: Juxtaposed homotopies

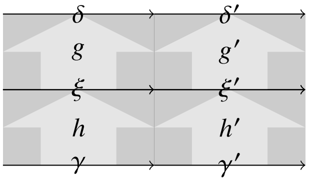


Figure 5.4: Godement exchange law

Directed Topology	Category theory
point	category
path	functor
homotopy	natural transformation
path concatenation	composition of functors
“piled up” homotopies	composition of natural transformations
“juxtaposed” homotopies	juxtaposition of natural transformations

Figure 5.5: Homotopies of paths vs natural transformations

5.2 Generic Construction

In this section, which heavily leans on Section 4.2, we describe the construction of the fundamental category functor over $\mathcal{C}$, a given framework for directed topology. In particular we have the forgetful functor $U : \mathcal{C} \to \mathcal{K}$ with $\mathcal{K}$ being some subcategory of **Top** – see the introduction of Section 4.2, and the embedding I : **Cub** $\hookrightarrow \mathcal{C}$ (*cf.* Figure 4.2). Following Remark 4.2.4 we write I_r instead of $I([0,r])$ for all $r \geqslant 0$. Let X be an object of $\mathcal{C}$. The category $\mathcal{P}X$ of dipaths on X is given by Definition 4.2.18.

Definition 5.2.1. Let γ and δ be two dipaths on X, a **dihomotopy** from γ to δ is a morphism from $h : I_r \times I_q \to X$ such that $U(h)$ is a homotopy from $U(\gamma)$ to $U(\delta)$ in the sense of Definition 5.1.1. In particular both γ and δ have to be defined on the same object I_r.

Remark 5.2.2. As a consequence of Theorem 4.1.6 any dihomotopy on a pospace is not far from being an isotopy (*i.e.* a homotopy whose intermediate mappings $h(\_, x)$ are topological embeddings).

Remark 5.2.3. Let h and h' be two dihomotopies respectively defined over $I_r \times I_q$ and $I_{r'} \times I_{q'}$. If $r = r'$ and $h(\_, q) = h'(\_, 0)$ (resp. $q = q'$ and $h(r, \_) = h'(0, \_)$) then by the first axiom of Definition 4.2.2 we can define the **vertical composition** $h' * h$ (resp. **horizontal composition** $h' \cdot h$) by pasting their domains of definition. By the third axiom of Definition 4.2.2, the constructions $h' * h$ and $h' \cdot h$ are characterized by

$$U(h' * h) = U(h') * U(h) \quad \text{and} \quad U(h' \cdot h) = U(h') \cdot U(h)$$

the right hand terms being given by Lemma 5.1.2 and Lemma 5.1.3. The Godement exchange law (*cf.* Remark 5.1.4) is still valid.

Remark 5.2.4. By contrast with usual homotopies $h^{op} : (t, x) \mapsto h(t, -x)$ might not be a dihomotopy even if h is so.

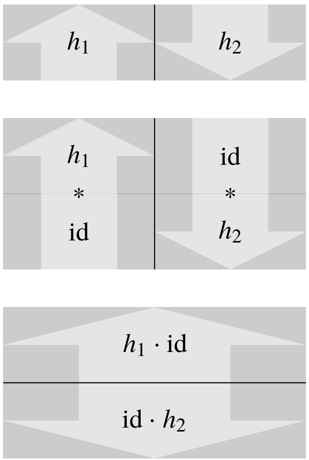


Figure 5.6: Swapping horizontal for vertical composition

Definition 5.2.5. The homotopy h is said to be an **antidihomotopy** when h^{op} is a dihomotopy. An **alternating** homotopy is a finite concatenation (*i.e.* finite sequence of vertical compositions) of dihomotopies and antidihomotopies. Two directed paths related by an alternating homotopy are said to be **dihomotopic**.

Remark 5.2.6. Let $h = h_n \cdot \ldots \cdot h_1$ be a finite sequence of horizontal compositions of homotopies and antidihomotopies forming a homotopy between the dipaths γ and δ. There exist an alternating homotopy between γ and δ. This is a consequence of the Godement exchange law – see Figure 5.6.

Intuitively two dipaths sharing their extremities should be declared "equivalent" when there is an alternating homotopy between them. However, as set theoretic mappings, the domains of definition of two such dipaths may differ. However, the first axiom of Definition 4.2.2 guarantees that any reparametrization (*cf.* Definition 4.5.28) from $[0, r]$ to $[0, r']$ induces a morphism from I_r to $I_{r'}$, so we write $\gamma \sim \delta$ when there exist two reparametrizations θ and θ' of (the domains of definition of) γ and δ such that there is an alternating homotopy between $\gamma \circ \theta$ and $\delta \circ \theta'$. In particular both θ and θ' have to be defined on the same compact segment.

Example 5.2.7. For example two dipaths on $[0, 1]^n$ sharing their extremities are dihomotopic.

Proposition 5.2.8. The relation $\sim$ is a congruence over $\mathcal{P}X$ in the sense of (Mac Lane, 1998, p.52).

Proof. First $\sim$ is reflexive considering the trivial dihomotopy. Then note that a homotopy is alternating iff its opposite is so. Therefore $\sim$ is symmetric. By definition of an alternating homotopy, the relation $\sim$ is transitive. As a consequence of Remark 5.2.6, it is a congruence. □

Remark 5.2.9. A **weak dihomotopy** is a homotopy whose intermediate paths are directed. It was introduced in (Raußen (2000)) relaxing the definition from (Fajstrup et al. (1999)) which demanded that all intermediate dipaths be *inextendible*. In particular, a map h is a weak dihomotopy if and only if the map h^{op} is so. Two dipaths are said to be

weakly dihomotopic when there exists a weak dihomotopy between them. However, weakly dihomotopic paths may not be dihomotopic. As a counter-example consider the unit sphere $\mathbb{S}^2$ partially ordered by $p \sqsubseteq q$ when both p and q lies on the same meridian and q is northernmost (Raußen, 2003, p.260). Yet, L. Fajstrup has proven that both notions coincide for local pospace realizations of geometric precubical sets (Fajstrup, 2005, Theorem 5.7). Later on, S. Krishnan has extended this result to all quadrangulable streams (Krishnan, 2013, Theorem 8.13).

Definition 5.2.10. The relation $\sim$ is called the **homotopy congruence** over X and the fundamental category of X, denote it by $\overrightarrow{\pi_1}X$, is the quotient

$$\mathcal{P}X/{\sim}$$

and we denote by $\overrightarrow{q_X}$ the quotient functor from $\mathcal{P}X$ to $\overrightarrow{\pi_1}X$. The notation does not refer to the category C when the context leaves no ambiguity[1].

Lemma 5.2.11. The construction of Definition 5.2.10 is functorial and the collection $\overrightarrow{q}$ of all functors $\overrightarrow{q_X}$ forms a natural transformation from $\mathcal{P}$ to $\overrightarrow{\pi_1}$.

Proof. Let γ and δ be directed paths on X, $f : X \longrightarrow Y$ be a morphism of the framework C, and $h = h_n * \cdots * h_1$ be an alternating homotopy between γ and δ (with each h_k being a dihomotopy or an antidihomotopy). Then $f \circ h = (f \circ h_n) * \cdots * (f \circ h_1)$ is an alternating homotopy between $f \circ \gamma$ and $f \circ \delta$. By definition of $\overrightarrow{\pi_1}$ and $\mathcal{P}$, for all morphisms $f : X \longrightarrow Y$ and all dipaths γ on X (*i.e.* all morphisms of $\mathcal{P}X$) we have

$$(q_Y \circ \mathcal{P}f)(\gamma) = q_Y(f \circ \gamma) = [f \circ \gamma] = \overrightarrow{\pi_1}f([\gamma]) = (\overrightarrow{\pi_1}f \circ q_X)(\gamma)$$

□

Mimicking (Brown, 2006, Section 6.4.4, p.222) we obtain

Lemma 5.2.12. The fundamental category preserves binary products.

The fundamental category of a local pospace enjoys a nice property.

Proposition 5.2.13. The dihomotopy class of a nonconstant directed loop on a local pospace is nonzero. In particular, the fundamental category of a local pospace has no nilpotent element.

Proof. Otherwise there would be a directed homotopy an extremity of which being a constant path, so the local pospace would contain a vortex, which would contradict Corollary 4.3.38. □

We provide some examples.

Example 5.2.14. The real line $\mathbb{R}$ can be equipped with the standard direction (*i.e.* only nondecreasing paths are directed), the discrete one (*i.e.* no path is directed but the constant ones), and the chaotic one (*i.e.* all paths are directed). The corresponding fundamental categories are the posets $(\mathbb{R}, \leqslant)$ and $(\mathbb{R}, =)$ in the first two cases, and the total equivalence relation over $\mathbb{R}$ in the last one.

Example 5.2.15. If we consider the framework **Top**, then $\overrightarrow{\pi_1}$ is the usual fundamental groupoid functor – see (Brown, 2006, Section 6.2, pp. 207–215).

[1] We write $\mathcal{P}X_C$ and $\overrightarrow{\pi_1}X_C$ to insist on the underlying framework for directed topology is C.

Example 5.2.16. The fundamental category of the directed complex plane $\mathbb{C}$ (*cf.* Example 4.5.18) can be described as follows: its objects are the complex numbers and its morphisms are

$$\left\{\, (z_0, n, z_1) \,\middle|\, z_0 \neq 0 \; ; \; |z_0| \leqslant |z_1| \; ; \; n \in \mathbb{N} \,\right\} \cup \left\{\, (0, \bot, z) \,\middle|\, z \in \mathbb{C} \,\right\}$$

with the convention that $\bot + n = n + \bot = \bot$ for all $n \in \mathbb{N} \cup \{\bot\}$. By definition the source and the target of (z_0, n, z_1) are z_0 and z_1. Given $x \in \mathbb{C}\backslash\{0\}$ we define $\mu(z) := \frac{z}{|z|}$ and $\overset{\curvearrowleft}{z_0 z_1}$ as the anticlockwise arc from $\mu(z_0)$ to $\mu(z_1)$ for any $z_0, z_1 \in \mathbb{C}\backslash\{0\}$. The composition is defined by

$$(z_1, m, z_2) \circ (z_0, n, z_1) = \begin{cases} (z_0, n + m, z_2) & \text{if } \overset{\curvearrowleft}{z_0 z_1} \cup \overset{\curvearrowleft}{z_1 z_2} \neq S^1 \\ (z_0, n + m + 1, z_2) & \text{if } \overset{\curvearrowleft}{z_0 z_1} \cup \overset{\curvearrowleft}{z_1 z_2} = S^1 \end{cases}$$

Note that if z, z', or z'' is 0, then n or m is $\bot$, and therefore $n + m = n + m + 1 = \bot$. The fundamental category of the directed Riemann sphere follows, adding a point at infinity ∞ and the morphisms $(z, \bot, \infty)$ for all complex numbers z.

Remark 5.2.17. Consider $\mathbb{S}^1$ as a d-subspace of $\mathbb{C}$. Then any dipath on $\mathbb{C}$ whose extremities lie in $\mathbb{S}^1$ entirely lies in it. Consequently, if $h : [0, r] \times [0, q] \to \mathbb{C}$ is an alternating homotopy such that $h(0, \_)$ and $h(r, \_)$ lie in $\mathbb{S}^1$, then the whole image of h is actually contained in $\mathbb{S}^1$. Two dipaths with their extremities in $\mathbb{S}^1$ that are dihomotopic in $\mathbb{C}$, are therefore dihomotopic in $\mathbb{S}^1$. In particular $\overrightarrow{\pi_1}\mathbb{S}^1$ is the full subcategory of $\overrightarrow{\pi_1}\mathbb{C}$ whose set of objects is $\mathbb{S}^1$. The fact is radically different from the classical case.

Example 5.2.18. As we have seen in Example 4.5.52, any Lie group G has a canonical direction. It is therefore natural to wonder what its fundamental category looks like. The special case of the unit circle has been treated in Remark 5.2.17. Since the product of two Lie groups is a Lie group (the underlying manifold being the product of their underlying manifolds while the group structure is the same as for usual groups) it is natural to restrict our attention to **irreducible** Lie groups (*i.e.* the ones that are not trivial and that cannot be written as the product of two non trivial Lie groups). If Conjecture 4.5.49 and Conjecture 4.5.51 are satisfied then the canonical direction of a Lie group derives from a local pospace. Then from Proposition 5.2.13 we know that its fundamental category has no nilpotent element. The spheres $\mathbb{S}^1$, $\mathbb{S}^3$ and $\mathbb{S}^7$ are known to be the only parallelizable spheres (*cf.* Example 4.5.52), each of them inherits a parallelization from its Lie group structure which is provided by complex numbers, quaternions, and octonions of magnitude 1.

Conjecture 5.2.19. For all x, we have $\overrightarrow{\pi_1}\mathbb{S}^1(x, x) \cong \overrightarrow{\pi_1}\mathbb{S}^3(x, x) \cong \overrightarrow{\pi_1}\mathbb{S}^7(x, x) \cong \mathbb{N}$.

We would like to generalize Conjecture 5.2.19 to all irreducible compact Lie groups. However, as an example of a connected compact Lie group whose fundamental group is $\mathbb{Z}_2$, the special orthogonal group $SO(n)$ (for $n \geqslant 3$) is worth a close examination.

5.3 Comparison

The genericity of the construction described in Section 5.2 allow us to compare fundamental categories through morphisms of frameworks for directed topology – see Definition 4.2.22. Let D be such a morphism from (C, U) and (C', U') and X be an object of C. Given $r, q \in \mathbb{R}_+$ we have

$$D(I_r \times I_q) = D \circ I([0, r] \times [0, q]) = I'([0, r] \times [0, q]) = I'_r \times I'_q$$

The first and the last equalities hold because both I and I' preserves binary Cartesian products (*cf.* Definition 4.2.2). The middle equality derives from the fact that D is a morphism of frameworks. In particular, if h is a directed homotopy on X from γ to δ, then $D(h)$ is a directed homotopy on DX from $D(\gamma)$ to $D(\delta)$. The same holds for antidihomotopies. Therefore $\gamma \sim \delta$ implies $D(\gamma) \sim D(\delta)$, with $\sim$ denoting the homotopy congruence over X or DX accordingly. Consequently we have a functor $\beta_X : \overrightarrow{\pi_1}X_C \to \overrightarrow{\pi_1}DX_{C'}$ whose object part is an identity, the morphism part being given by

$$\beta_X\big([\delta]\big) = \big[D(\delta)\big]$$

where $[\delta]$ and $[D(\delta)]$ are the dihomotopy classes of the directed paths δ and $D(\delta)$.

Lemma 5.3.1. The functors β_X for X running through the collection of objects of C form a natural transformation from $\overrightarrow{\pi_1}_C$ to $\overrightarrow{\pi_1}_{C'}$. Moreover the functor β_X is an isomorphism if and only if for all directed paths δ'_1 and δ'_2 on DX such that $\delta'_1 \sim_{DX} \delta'_2$ there exist two directed paths δ_1 and δ_2 on X such that $D(\delta_1) \sim_{DX} \delta'_1$, $D(\delta_2) \sim_{DX} \delta'_2$ and $\delta_1 \sim_X \delta_2$.

Proof. From the description of the morphism part of the functor β_X. □

Corollary 5.3.2. Given r and ρ in $\mathbb{R}_+$, the maps A_r and $B_{r,\rho}$ are defined as below.

$$\begin{tikzcd} C(I_r, X) \arrow[r, "A_r"] & C'(I'_r, D(X)) \\ \delta \arrow[r, mapsto] & D(\delta) \end{tikzcd} \qquad \begin{tikzcd} C(I_r \times I_\rho, X) \arrow[r, "B_{r,\rho}"] & C'(I'_r \times I'_\rho, D(X)) \\ h \arrow[r, mapsto] & D(h) \end{tikzcd}$$

Assuming that the maps A_r are bijective and that the maps $B_{r,\rho}$ are surjective, we have $\mathcal{P}X \cong \mathcal{P}DX$ and $\overrightarrow{\pi_1}X \cong \overrightarrow{\pi_1}DX$.

Proof. As an immediate consequence of the Lemmas 4.2.23 and 5.3.1. □

Corollary 5.3.3. Suppose that D admits a left adjoint S and denote the unit by η. Also suppose that for all $r, q \in \mathbb{R}_+$ we have

$$\eta_{I'_r} = \mathsf{id}_{I'_r} \quad \text{and} \quad S(I'_r) = I_r \quad \text{and} \quad S(I'_r \times I'_q) = I_r \times I_q$$

Then the natural transformations $\alpha : \mathcal{P}X_C \to \mathcal{P}DX_C$ and $\beta : \overrightarrow{\pi_1}X_C \to \overrightarrow{\pi_1}DX_C$ are isomorphisms.

Proof. The mappings below are bijections because S is left adjoint to D.

$$\begin{tikzcd} C[S(I'_r), X] \arrow[r] & C'[I'_r, D(X)] \\ \delta \arrow[r, mapsto] & D(\delta) \circ \eta_{I'_r} \end{tikzcd} \qquad \begin{tikzcd} C[S(I'_r \times I'_\rho), X] \arrow[r] & C'[I'_r \times I'_\rho, D(X)] \\ h \arrow[r, mapsto] & D(h) \circ \eta_{I'_r \times I'_\rho} \end{tikzcd}$$

In addition we have

$$C'[I'_r, D(X)] = C[S(I'_r), X] = C[I_r, X]$$

and

$$C'[I'_r \times I'_\rho, D(X)] = C[S(I'_r \times I'_\rho), X] = C[I_r \times I_\rho, X]$$

then we conclude applying Corollary 5.3.2 (with $\eta_{I'_r} = \mathrm{id}_{I'_r}$) and Lemma 5.3.4[2] which gives $\eta_{I'_r \times I'_\rho} = \eta_{I'_r} \times \eta_{I'_\rho} = \mathsf{id}_{I'_r \times I'_\rho}$.

[2] I explicitly provide the statement and its proof for I could not find them anywhere in the "classics" of Category Theory.

Lemma 5.3.4. Let $F \dashv U : \mathcal{A} \underset{F}{\overset{U}{\rightleftarrows}} \mathcal{B}$ be an adjunction and let B, B' be two objects of $\mathcal{B}$ such that $F(B \times B') = FB \times FB'$, then $\eta_{B\times B'} = \eta_B \times \eta_{B'}$.

The proof of Lemma 5.3.4 relies on basic facts about adjunction that can be found in Borceux (1994a). Let η and ε be the unit and the counit, it suffices to check that

$$\begin{cases} \eta_B \circ \Pi_B &= \Pi_{UFB} \circ \eta_{B\times B'} \\ \eta_{B'} \circ \Pi_B &= \Pi_{UFB'} \circ \eta_{B\times B'} \end{cases}$$

Taking the equality $F(B \times B') = FB \times FB'$ into account, the following maps are bijections

$$\begin{tikzcd} \mathcal{A}[FB \times FB', FB] \arrow[r, shift left] & \mathcal{B}[B \times B', UFB] \arrow[l, shift left] \\ \gamma \arrow[r, mapsto] & U(\gamma) \circ \eta_{B\times B'} \\ \varepsilon_{FB} \circ F(\delta) & \delta \arrow[l, mapsto] \end{tikzcd}$$

Moreover $\varepsilon_{FB} \circ F(\eta_B) = \mathrm{id}_{FB}$ hence we have

$$\varepsilon_{FB} \circ F(\eta_B \circ \Pi_B) = \big(\varepsilon_{FB} \circ F(\eta_B)\big) \circ F(\Pi_B) = F(\Pi_B) = \Pi_{FB}\ ,$$

the last equality holds because F preserves the product $B \times B'$. Thus $U(\Pi_{FB}) \circ \eta_{B\times B'} = \eta_B \circ \Pi_B$. As a right adjoint, U preserves products hence $U(\Pi_{FB}) = \Pi_{UFB}$ and we have $\eta_B \circ \Pi_B = \Pi_{UFB} \circ \eta_{B\times B'}$. □

Remark 5.3.5. In most of practical cases, the extra hypotheses of Corollary 5.3.3 about the adjunction are obviously satisfied. It suffices, for example, that S be a morphism of frameworks and $\eta_{I'_r} = \mathrm{id}_{I'_r}$.

As we shall see, there might be objects X' of C' such that $\overrightarrow{\pi_1}X' \not\cong \overrightarrow{\pi_1}SX'$. However, if $D \circ S(X') = X'$ then $\overrightarrow{\pi_1}SX' = \overrightarrow{\pi_1}DSX' = \overrightarrow{\pi_1}X'$.

We apply the previous results together with facts about streams and d-spaces from Chapter 4 to compare their fundamental categories.

Corollary 5.3.6. For all streams X, we have $\overrightarrow{\pi_1}X = \overrightarrow{\pi_1}DX$.

Proof. We have seen that D is right adjoint to S (*cf.* Remark 4.5.10). We check that S preserves binary products and we conclude applying Corollary 5.3.3. □

Corollary 5.3.7. For all d-space X, if there exists a stream X' such that $D(X') = X$ then $\overrightarrow{\pi_1}SX = \overrightarrow{\pi_1}X$.

Proof. By Proposition 4.5.12 we have $DSDX' = DX'$ and by Corollary 5.3.6 we have $\overrightarrow{\pi_1}DSDX' = \overrightarrow{\pi_1}SDX'$. □

Corollary 5.3.8. Given a cubical set K the fundamental categories of the following realizations are isomorphic $D(\rceil K\lfloor_{\mathbf{Strm}})$, $\rceil K\lfloor_{\mathbf{Strm}}$, $\rceil K\lfloor_{\mathbf{Strm}_\mathrm{d}}$, $S(\rceil K\lfloor_{\mathbf{dTop}_\mathrm{f}})$ and $\rceil K\lfloor_{\mathbf{dTop}_\mathrm{f}}$.

Proof.

$$\begin{array}{rcll} \overrightarrow{\pi_1}D(\rceil K\lfloor_{\mathbf{Strm}}) &=& \overrightarrow{\pi_1}\rceil K\lfloor_{\mathbf{Strm}} & \text{by Corollary 5.3.6} \\ \overrightarrow{\pi_1}\rceil K\lfloor_{\mathbf{Strm}} &=& \overrightarrow{\pi_1}\rceil K\lfloor_{\mathbf{Strm}_\mathrm{d}} & \text{by Corollary 4.5.24} \\ \overrightarrow{\pi_1}\rceil K\lfloor_{\mathbf{Strm}_\mathrm{d}} &=& \overrightarrow{\pi_1}S(\rceil K\lfloor_{\mathbf{dTop}_\mathrm{f}}) & \text{by Corollary 4.5.23} \\ \overrightarrow{\pi_1}S(\rceil K\lfloor_{\mathbf{dTop}_\mathrm{f}}) &=& \overrightarrow{\pi_1}\rceil K\lfloor_{\mathbf{dTop}_\mathrm{f}} & \text{by Corollary 5.3.7} \end{array}$$

□

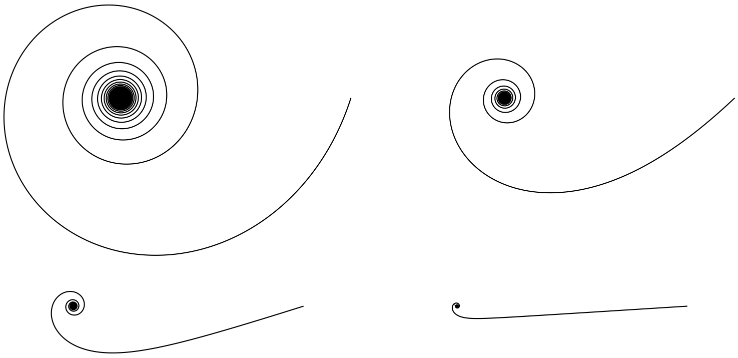


Figure 5.7: A dihomotopy from a directed path to a pseudo-directed path

Remark 5.3.9. The "downward spiral" (*cf.* Figure 4.7) illustrates the fact that if K is the cubical set described in Example 4.5.18, then the canonical map $\rceil K \lfloor_{\mathbf{dTop}} \hookrightarrow \rceil K \lfloor_{\mathbf{dTop}_\mathrm{f}}$ is not an isomorphism. However the following map induces a dihomotopy from $t \in [0,1] \mapsto t$ to the "downward spiral" in $\rceil K \lfloor_{\mathbf{dTop}_\mathrm{f}}$.

$$h : (t,s) \in [0,1]^2 \mapsto t \cdot e^{s \cdot \frac{i2\pi(1-t)}{t}} \in \mathbb{C}$$

Then we observe that $h(\_,0)$ is a d-path though all the other intermediate paths $h(\_,s)$ for $s > 0$ are just pseudo d-paths (*cf.* Definition 4.5.14). The intermediate paths corresponding to parameters $s \in \{0.5, 02, 0.05, 0.01\}$ are shown on Figure 5.7. In fact we have $\overrightarrow{\pi_1} \rceil K \lfloor_{\mathbf{dTop}_\mathrm{f}} \cong \overrightarrow{\pi_1} \rceil K \lfloor_{\mathbf{dTop}}$.

In the same vein as Conjecture 4.5.27, we expect that the fundamental categories of precubical set realizations behave well.

Conjecture 5.3.10. For all precubical sets K, we have the following equality.

$$\{\text{dipaths on } \rceil K \lfloor_{\mathbf{dTop}}\} = \{\text{dipaths on } \rceil K \lfloor_{\mathbf{Strm}}\} = \{\text{dipaths on } \rceil K \lfloor_{\mathbf{LpoTop}}\}$$

The latter actually extends to an equality between fundamental categories.

$$\overrightarrow{\pi_1} \rceil K \lfloor_{\mathbf{dTop}} = \overrightarrow{\pi_1} \rceil K \lfloor_{\mathbf{Strm}} = \overrightarrow{\pi_1} \rceil K \lfloor_{\mathbf{LpoTop}}$$

Remark 5.3.11. Note that if $\rceil K \lfloor_{\mathbf{dTop}} = \rceil K \lfloor_{\mathbf{dTop}_\mathrm{f}}$ then by Corollary 4.5.25 and $S \dashv D$ we have

$$\mathbf{dTop}(I_r, \rceil K \lfloor_{\mathbf{dTop}}) = \mathbf{dTop}(I_r, D(\rceil K \lfloor_{\mathbf{Strm}})) = \mathbf{Strm}(S(I_r), \rceil K \lfloor_{\mathbf{Strm}}),$$

and also by Corollary 5.3.6

$$\overrightarrow{\pi_1} \rceil K \lfloor_{\mathbf{dTop}} \cong \overrightarrow{\pi_1}(D(\rceil K \lfloor_{\mathbf{Strm}})) \cong \overrightarrow{\pi_1} \rceil K \lfloor_{\mathbf{Strm}} .$$

5.4 A Computation Tool: The Seifert - van Kampen Theorem

The next result provides a tool for computing fundamental categories. An element α of $\mathcal{C}(X,Y)$ is called an **inclusion** when $U(X)$ is a subspace of $U(Y)$ (in the sense of Remark 4.2.1) and $U(\alpha)$ is the corresponding inclusion. In this case the notation $\mathrm{int}(UX)$ stands for the topological interior of $U(X)$ seen as a subset of $U(Y)$. Then we have the generic form of the van Kampen theorem.

Theorem 5.4.1 (Seifert - van Kampen)**.**
The functors $\mathcal{P}$ and $\overrightarrow{\pi_1}$ send any square of inclusions of C such that $\mathrm{int}(UX_1)$ and $\mathrm{int}(UX_2)$ cover $U(X)$ and $U(X_0) = U(X_1) \cap U(X_2)$ to pushout squares of **Cat**.

$$\begin{tikzcd} X_0 \arrow[r] \arrow[d] & X_1 \arrow[d] \\ X_2 \arrow[r] & X \end{tikzcd} \qquad \begin{tikzcd} \mathcal{P}X_0 \arrow[r] \arrow[d] & \mathcal{P}X_1 \arrow[d] \\ \mathcal{P}X_2 \arrow[r] & \mathcal{P}X \end{tikzcd} \qquad \begin{tikzcd} \overrightarrow{\pi_1}X_0 \arrow[r] \arrow[d] & \overrightarrow{\pi_1}X_1 \arrow[d] \\ \overrightarrow{\pi_1}X_2 \arrow[r] & \overrightarrow{\pi_1}X \end{tikzcd}$$

Proof. Provided we pay some attention to the details pointed out below, it suffices to mimic the proof of the classical van Kampen theorem for groupoids given in (Higgins, 1971, Chapter 17) and (Brown, 2006, Section 6.7). Compactness actually remains the cornerstone of the argument.

First we need that for all $\gamma \in C[I_r, X]$, all $h \in C[I_\rho \times I_q, X]$, all closed intervals $\iota \subseteq [0,r]$ and all closed rectangles $\iota_1 \times \iota_2 \subseteq [0,\rho] \times [0,s]$, the restriction of α to ι and the restriction of h to $\iota_1 \times \iota_2$ induce morphisms of C. Writing these restrictions as the following composites, it is an immediate consequence of the first axiom of Definition 4.2.2.

$$\alpha|_\iota = \iota \hookrightarrow [0,r] \xrightarrow{\alpha} X \qquad\qquad h|_{\iota_1\times\iota_2} = \iota_1 \times \iota_2 \hookrightarrow [0,\rho] \times [0,s] \xrightarrow{h} X$$

The classical proof also uses the fact that any two paths on a rectangle sharing the same extremities are homotopic. Our context requires a similar result involving increasing paths and alternating homotopies. Given α and β two continuous increasing maps from $[0,r]$ to some rectangle $R = [0,a] \times [0,b]$ such that $\alpha(0) = \beta(0)$ and $\alpha(r) = \beta(r)$, we remark that the map γ from $[0,r]$ to R defined by $\gamma(t) = \max(\alpha(t), \beta(t))$ is still continuous and increasing. It follows that the map h from $[0,r] \times [0,1]$ to R defined by

$$h(t,s) := (1-s)\cdot\alpha(t) + s\cdot\gamma(t) = \alpha(t) + s\cdot(\gamma(t) - \alpha(t))$$

is continuous and increasing with respect to the product order on R. Therefore, in virtue of the first axiom of Definition 4.2.2, the map h induces a morphism of C. Consequently, it is a dihomotopy from α to γ. In the same way, we obtain a dihomotopy from β to γ, thus providing the alternating homotopy. □

Remark 5.4.2. We insist that weak dihomotopies (*cf.* Remark 5.2.9) cannot be substituted for alternating homotopies in the proof of Theorem 5.4.1. The last part implicitly relies on the fact that the domain of definition of an alternating homotopy h can be divided into rectangles on which either the restriction of h or that of h^{op} is a morphism of C, which might fail for weak dihomotopies.

Theorem 5.4.1 can be summarized by the commutative cube in Figure 5.8, whose upper and lower faces are pushout squares.

Example 5.4.3. Following Brown (2006) we use the complex number notation and cover the directed circle with $\mathbb{S}^1 \setminus \{i\}$ and $\mathbb{S}^1 \setminus \{-i\}$ to compute its fundamental category. The result has already been given in Remark 5.2.17.

$$\begin{tikzcd}
 & & \mathcal{P}X_1 \arrow[dr] \arrow[ddd] & \\
\mathcal{P}X_0 \arrow[urr] \arrow[dr] \arrow[ddd] & & & \mathcal{P}X \arrow[ddd] \\
 & \mathcal{P}X_2 \arrow[urr] \arrow[ddd] & & \\
 & & \overrightarrow{\pi_1}X_1 \arrow[dr] & \\
\overrightarrow{\pi_1}X_0 \arrow[urr] \arrow[dr] & & & \overrightarrow{\pi_1}X \\
 & \overrightarrow{\pi_1}X_2 \arrow[urr] & &
\end{tikzcd}$$

Figure 5.8: The Directed Seifert-van Kampen Theorem.

In the framework **Top** Theorem 5.4.1 amounts to the classical van Kampen Theorem for fundamental groupoids. It actually admits a much stronger (and useful) form established in (Brown, 2006, Chap.9). This generalization applies to adjunction spaces and its statement involves the notion of homotopy equivalence of groupoids. Unfortunately, no directed counterpart of the latter notion has been found yet – see also Section 10.2 for a discussion around what the notion of directed homotopy equivalence should satisfy.

5.5 Enveloping Groupoids *vs* Fundamental Groupoids

The inclusion functor **Grd** $\hookrightarrow$ **Cat** has a left adjoint G which is obtained by providing all the morphisms of a given small category with an inverse. The construction is an instance of **category of fractions/localization** – see (Gabriel and Zisman, 1967, Chap.1), (Borceux, 1994a, Chap.5). The groupoid $G(C)$ is called the **enveloping groupoid** (or **groupoid hull**) of the category C. Given an object X of a framework for directed topology one may ask whether the enveloping groupoid of its fundamental category $G(\overrightarrow{\pi_1}X)$ matches the fundamental groupoid of its underlying space $\Pi_1(UX)$, in other words

$$G(\overrightarrow{\pi_1}X) \quad \cong \quad \Pi_1(UX) \tag{5.1}$$

Even more accurately, the universal property of enveloping groupoids gives a collection of groupoid morphisms $g_X : G \circ \overrightarrow{\pi_1}(X) \rightarrow \Pi_1 \circ U(X)$ which actually forms a natural transformation from $G \circ \overrightarrow{\pi_1}$ to $\Pi_1 \circ U$. Then we would like to know when g_X is an isomorphism.

One easily finds counterexamples (*e.g.* $\mathbb{R}$ seen as a pospace with the discrete order) but we also note that the result holds for the directed versions of $\mathbb{R}$, $\mathbb{S}^1$, $\mathbb{C}$, the Riemann sphere Σ, and $\rceil G \lfloor$ for any graph G. Moreover, during his internship, Rémi Géraud (2012) sketched a proof assuming that X is the directed realization of a geometric precubical set – see Fajstrup (2005). The fact that Relation 5.1 holds for the directed complex plane lets us think that it could actually hold for many directed geometric realizations of cubical sets, maybe all of them. To put it another way, we would like to know whether the outer shape of the diagram on Figure 5.9 (where $|\_|$ and $\rceil\_\lfloor$ should be understood as the geometric and the filled d-space realization functors) is commutative (note that the left-hand triangle on Figure 5.9 commutes by Remark 4.5.22).

$$\begin{tikzcd}
 & \mathbf{dTop}_{\mathrm{f}} \arrow[rr, "\overrightarrow{\pi_1}"] \arrow[drrr, bend left=15, "G\circ\overrightarrow{\pi_1}", ""{name=A, below}] \arrow[drrr, bend right=25, "\overrightarrow{\pi_1}\circ U"', ""{name=B, above}] \arrow[ddr, "U"'] & & \mathbf{Cat} \arrow[dr, "G"] & \\
\mathbf{CSet} \arrow[ur, "\rceil\_\lfloor"] \arrow[drr, "|\ \_\ |"'] & & & & \mathbf{Grd} \\
 & & \mathbf{Top} \arrow[urr, "\Pi_1"'] & &
\arrow[Rightarrow, from=A, to=B, "g"']
\end{tikzcd}$$

Figure 5.9: Enveloping groupoid vs fundamental groupoid

Remark 5.5.1. If Conjecture 5.2.19 is true, then both $\mathbb{S}^3$ and $\mathbb{S}^7$ provide nontrivial counterexamples to Relation 5.1. Moreover, if the latter holds for all realizations of cubical sets, then the directions associated with $\mathbb{S}^3$ and $\mathbb{S}^7$ are nontrivial example of d-spaces that cannot be realized as cubical sets (though one can obtain their underlying spaces that way).

In particular for all cubical sets K, Relation 5.1 is satisfied for $X = \rceil K \lfloor_{\mathbf{dTop}_{\mathrm{f}}}$ iff it is satisfied for $X = \rceil K \lfloor_{\mathbf{Strm}_{\mathrm{d}}}$ (*cf.* Corollary 5.3.6 and Corollary 5.3.7). Provided Conjecture 4.3.44 turns out to be true, we could have also substituted local pospaces for filled d-spaces (resp. filled streams) in the statement of the problem – see Definition 4.4.11 and Definition 4.5.14. We also remark that if $X_1, \ldots, X_n$ satisfy Relation 5.1, then so does their Cartesian product $X_1 \times \ldots \times X_n$ because the functors U, G, Π_1, and $\overrightarrow{\pi_1}$ preserve binary products. As a consequence Relation 5.1 holds for any tensor product of graphs. Pushing this further, we expect the following is true.

Conjecture 5.5.2. For all isothetic regions X (*cf.* Definition 6.2.1) seen as local pospaces, we have the following isomorphism.

$$G(\overrightarrow{\pi_1}X) \cong \Pi_1(UX)$$

From a computer scientist point of view, the matter raised in this section is related to the notion of reversible computation – Krivine (2006); Danos et al. (2007); Krivine (2012); Cristescu et al. (2013). For a programming language implementing it, the instruction pointer is allowed, under some circumstances, to go backward so that the system state can be restored as it was before the execution of an instruction. In this context, if the execution traces are organized into a category whose objects are the system states, then it is a groupoid. However this groupoid is polarized in the sense that it is generated by a collection of morphisms any element of which being either seen as forward (or positive say) or backward (negative). As a consequence, it is still possible to say whether a computation is progressing or regressing. This feature is especially interesting as a systemic way to prevent concurrent programs from deadlocks.

5.6 Trace Spaces

The set of directed paths $P(X)$ on the geometric model X of a program P is aimed to be a rather sharp overapproximation of its set of execution traces. In particular $P(X)$ comes equipped with the compact open topology – Kelley (1955), and it is therefore natural to study it as a topological space. Martin Raußen has dedicated a series of papers to this question. In first, the directed paths are reduced to traces (*i.e.* considered up to reparametrization) thus giving rise to the so-called trace space $T(X)$ – see Fahrenberg and Raußen (2007); Raußen (2007, 2009a). Then Raußen (2009b) has proven that the trace space of a precubical complex has the homotopy type of a CW-complex (though it is obtained as the quotient of some function space). The trace of a path can thus be understood as a parameter and therefore the trace space $T(X)$ as a moduli space of $P(X)$. More applications, descriptions, and effective computations can be found in Raußen (2010, 2012b,a); Fajstrup et al. (2012). Note in particular that the fundamental category $\overrightarrow{\pi_1}X$ can be completely recovered from $T(X)$. The homsets of the fundamental category $\overrightarrow{\pi_1}X(x, y)$ are indeed in bijection with the (arc-)connected components of $T(X)[x, y]$. The last chapter of the book (Fajstrup et al. (2016)) is dedicated to trace spaces.

6

Isothetic Regions

This chapter is dedicated to a special class C of directed topological models that enjoys the following crucial properties:

- it is broad enough to model all parallel automata (*cf.* Definition 1.1.7),
- it is simple enough to be handled by computers, and
- all its members can be indifferently seen as local pospaces, streams, or d-spaces.

From a theoretical point of view, the parallel automata are strongly related to the tensor products of graphs in the category of precubical sets, hence to special instances of higher dimensional automata.

The second point means that C is stable under all the operations one may need to model concurrency. These operations are furthermore implemented in an OCaml library.

An element of C can be seen as an object of several frameworks for directed topology (*cf.* Definition 4.2.2). As we have defined the fundamental category functor for any such framework (*cf.* Definition 5.2.10), it is natural to ask whether switching to another framework alters the fundamental category of an object of C. The answer to this last point is 'No'.

Mathematically speaking, the class C can be described as follows. One starts with $|G|$ the topological realization of a given graph G. The graph G is supposed to be so that the finite unions of connected components of $|G|$ form a Boolean algebra (*cf.* Section 6.1). All it takes for this property to be satisfied is for G to be finite (*e.g.* a disjoint union of finitely many control flow graphs). A block of dimension $n \in \mathbb{N}$ is a n-fold product of connected subsets of $|G|$ and the n-dimensional elements of C are the finite union of blocks (*cf.* Section 6.4). So any isothetic region of dimension n can be seen as a (likely infinite) set of points of $|G|^n$ (*i.e.* words of length n over the alphabet $|G|$) or as a finite family of blocks (*i.e.* words of length n over the alphabet of connected subsets of $|G|$). The second point of view is not canonical because different families of blocks can cover the same isothetic region. The abstract context for relating both approaches is described in Section 6.2. Following these points of view we note that the product of two blocks of dimensions n and m is a block of dimension $n + m$ which canonically corresponds to the concatenation of the words that represent them. By extension, the latter remark applies to isothetic regions seen as languages (*cf.* Section 6.3).

Following (Mac Lane, 1998, p.48-51), all the graphs under consideration are directed and may have several arrows from a vertex to another. Directed graph realizations are thoroughly studied in Section 6.1. We state and prove Theorems 6.1.20 and 6.1.42 which characterize the graphs whose realizations will be handled in the rest of the chapter. Any finite graph matches the requirements. In Section 6.3, we introduce some notation related to sets of words whose elements all have the same length, the so-called homogeneous languages (*cf.* Definition 6.3.1). In Section 6.2, we explain how block coverings, seen as homogeneous languages, are related to isothetic regions by an isomorphism of Boolean algebras. Additional operators on the Boolean algebra of isothetic regions encode the topology and the direction naturally held by any such isothetic region. They are described in Section 6.4. We exploits the concepts introduced in the preceding sections to define the continuous models of Paml programs in Section 7.1. The content of Section 6.5 is more exploratory, its purpose is to equip every isothetic region with quantitative structures, namely a metric and a measure.

6.1 The Directed Geometric Realization of a Graph

We study the realizations of graphs as their collections of connected components will be used as alphabets to define the isothetic regions in Section 6.4. A graph can be understood as a 1-dimensional precubical set so the category **Grph** is a full subcategory of **pCSet**. In particular one can restrict the realization functors defined over **pCSet** to **Grph** (*cf.* Section 2.4). In this section, we provide a simple explicit description of the directed realization of any graph, from which we easily determine its group of automorphisms. We also introduce a special class of graph (*cf.* Definition 6.1.43) whose members are associated with Boolean structures. They will play a crucial role in the sequel, mostly because the model of a parallel program $P_1|\cdots|P_n$ is actually a subobject of $\rceil G_1 \otimes \cdots \otimes G_n \lfloor$ where each G_i is the control flow graph of P_i (*cf.* Section 7.1). The notation $\rceil\_\lfloor$ refer to the realization (*cf.* Section 2.2) in some of the categories described in Chapter 4.

Let G be a graph and v be one of its vertices. An arrow α is said to be v**-ingoing** (resp. v**-outgoing**) when $\partial^+\alpha = v$ (resp. $\partial^-\alpha = v$), it is said to be v**-adjacent** when it is v-ingoing or v-outgoing. A vertex v' is a **neighbor** of v when $v \neq v'$ and there exists an arrow that is adjacent to both v and v'. The **degree** of a vertex v is

$$\mathrm{deg}(v) = \#\{v\text{-ingoing}\} + \#\{v\text{-outgoing}\}$$

A **subgraph** G' of G is a subset A' of the set of arrows of G together with a subset V' of the set of vertices of G such that the extremities of any element of A' belong to V'. A subgraph G' of G is said to be **full** when it contains any arrow of G which is adjacent to some vertex of G'. The **neighborhood** of a vertex v is the full subgraph of G containing v and all its neighbors. Then $\mathrm{deg}(v) = 2$ iff the neighborhood of v is isomorphic to one of the graphs on Figure 6.1. The **LpoTop**-realization of a graph $G : A \rightrightarrows V$ exists by (Fajstrup et al., 2006, p.262, Theorem 6.23) and its underlying space is the geometric realization of G. We describe it extensively. The underlying set of $\rceil G\lfloor_{\mathbf{LpoTop}}$ is the disjoint union of V and $A\times]0,1[$. A basis $\mathcal{B}$ of the topology of $|G|$ (*i.e.* the underlying space) is given by the open subsets $\{\alpha\} \times U$ with U being an open interval of $]0,1[$ and $\alpha \in A$; and $\{v\} \cup v_\varepsilon^+ \cup v_\varepsilon^-$ with v_ε^+ (resp. v_ε^-) being the union of $\alpha\times]0,\varepsilon[$ (resp. $\alpha\times]1-\varepsilon,1[$) for all $\alpha \in A$ such that $\partial^-\alpha = v$ (resp. $\partial^+\alpha = v$) and $0 < \varepsilon < \frac{1}{2}$. As a consequence we have the next two lemmas.

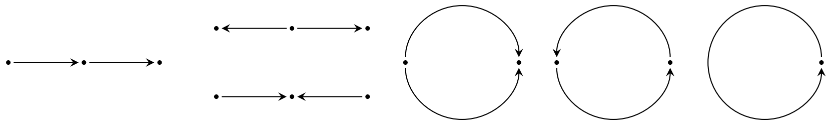


Figure 6.1: The neighborhoods of a vertex of degree 2

Lemma 6.1.1. The geometric realization of a graph is locally connected and locally simply connected.

Lemma 6.1.2. The geometric realization of a graph is locally compact iff the degrees of its vertices are finite.

The open subsets of the first kind (*i.e.* $\{\alpha\} \times U$) inherits their pospace structure from the open interval U while the open subsets of the second kind (*i.e.* $\{v\} \cup v_\varepsilon^+ \cup v_\varepsilon^-$) are (up to isomorphism) the open stars St_i^o with $i = \#\{v\text{-ingoing}\}$ and $o = \#\{v\text{-outgoing}\}$ (*cf.* Example 4.1.15 and Figure 4.1). The partial order $\sqsubseteq$ on $\{v\} \cup v_\varepsilon^+ \cup v_\varepsilon^-$ is indeed determined by the following constraints:

- $v_\varepsilon^- \sqsubset \{v\} \sqsubset v_\varepsilon^+$
- $v_\varepsilon^\pm$ inherits from the total order of $\mathbb{R}$

Remark 6.1.3. The numerical constraint $\varepsilon < \frac{1}{2}$ ensures that two “branches” of the chart $\{v\} \cup v_\varepsilon^+ \cup v_\varepsilon^-$ do not intersect unless they are equal. The problematic case arises when the arrow α is a loop since the chart then contains the set $\{\alpha\} \times (]0, \varepsilon[\cup]1 - \varepsilon, 1[)$ and $(\alpha, t') \sqsubset (\alpha, t)$ for $t \in]0, \varepsilon[$ and $t \in]1 - \varepsilon, 1[$.

Remark 6.1.4. The geometric realization of a graph G is metrizable, in fact one can easily describe a metric of $|G|$ that induces its topology (Bridson and Haefliger, 1999, p.6, Section 1.9).

Each element of the basis $\mathcal{B}$ is thus equipped with a closed partial order thus providing an atlas $\mathcal{A}$ over $|G|$ (*cf.* Definition 4.3.1).

Remark 6.1.5. Any point x of the underlying space $|G|$ has a basis of neighborhoods all the elements of which are isomorphic to the colimit (in **Top**) of n copies of $\mathbb{R}_+$ sharing their origin. If the point x belongs to V then $n = \deg(x)$ otherwise $n = 2$. Note that a point is isolated iff its degree is null. Also note that n may be an infinite cardinal. The cardinal n is the degree of the point $x \in |G|$.

Remark 6.1.6. There is a canonical bijection between the connected components of G and the ones of $|G|$.

The local pospace $\rceil G \lfloor_{\mathbf{LpoTop}}$ induces a d-space $I(\rceil G \lfloor_{\mathbf{LpoTop}})$ (*cf.* Example 4.5.9) whose direction is generated by the paths $p_\alpha : t \in [0, 1] \mapsto (\alpha, t)$ if $0 < t < 1$; $\partial^{\text{-}}\alpha$ if $t = 0$; and $\partial^+\alpha$ if $t = 1$. The following result is implicitly used to avoid the subscript that indicates in which category the realization of a graph is considered.

Proposition 6.1.7 (Haucourt (2012))**.** Given any graph G, the following d-spaces are isomorphic:

$$\rceil G\lfloor_{\mathbf{dTop}} \cong \rceil G\lfloor_{\mathbf{dTop_f}} \cong \rceil G\lfloor_{\mathbf{dTop_c}} \cong \rceil G\lfloor_{\mathbf{dTop_{cf}}} \cong I(\rceil G\lfloor_{\mathbf{LpoTop}})$$

In addition, for all directed paths δ and all connected subsets C of $\rceil G \lfloor$, the inverse image $\delta^{\text{-}1}(C)$ has finitely many connected components.

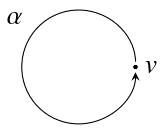


Figure 6.2: The loop graph

name of G	$\lvert G\rvert$	Characterization
point	$\{0\}$	the degree of some vertex is 0
segment	[0,1]	there are two vertices of degree 1
half-line	$\mathbb{R}_+$	there is one vertex of degree 1
line	$\mathbb{R}$	all the vertices have degree 2 and G is infinite
circle	$\mathbb{S}^1$	all the vertices have degree 2 and G is finite

Figure 6.3: Classification of the linear graphs

Proof. See Haucourt (2012). □

Lemma 6.1.8. The realization functors described in Proposition 6.1.7 reflect the isomorphisms.

Proof. Given a graph morphism f with domain G, the morphism $\rceil f \lfloor$ is defined by $\rceil f \lfloor (v) = v$ for all vertices of G, and by $\rceil f \lfloor (\alpha, t) = (f(\alpha), t)$ for all arrows of G and all $t \in]0, 1[$. This setting holds regardless of the category in which the graph is realized. Suppose that $\rceil f \lfloor$ is an isomorphism, in particular it is a bijection. By construction of $\rceil f \lfloor$ the morphism f induces both a bijection between the sets of vertices and a bijection between the sets of arrows. In other words it is an isomorphism of graphs. □

Example 6.1.9. For example, the directed circle is obtained by considering the graph with a single vertex and a single arrow.

We recall that $|G|$ denotes the topological realization of the graph G and introduce a structure that will be of crucial importance in the sequel. In particular the static analyzer ALCOOL is based on it.

Remark 6.1.10. The topological realization of a graph G is locally compact iff all its vertices have finitely many adjacent arrows. Following the description of $|G|$, a neighborhood of a vertex v should indeed contain $\{v\} \cup v_\varepsilon^+ \cup v_\varepsilon^-$ for some $\varepsilon > 0$, and therefore it cannot be compact if v has infinitely many adjacent arrows. Conversely, if v has finitely many adjacent arrows then the closure of $\{v\} \cup v_\varepsilon^+ \cup v_\varepsilon^-$ is compact.

The next result can be seen as a generalization of the well-known fact that $|G|$ is compact iff G is finite. In some sense, it characterizes the graphs that could be considered as finite regardless of the direction. A connected graph is said to be **linear** when the degree of any of its vertices is at most 2. A classification of linear graphs is given by Figure 6.3. We will also need the notion of Freudenthal extension which we now explain, see also (Porter, 1995, p.130-136). For any topological space X, denote the collection of closed compact subspaces of X by $\mathcal{K}(X)$. If K_0 and K_1 are closed compact subspaces of X with $K_0 \subseteq K_1$, and C_1 is a connected component of $X \setminus K_1$, then there exists a unique connected component C_0 of $X \setminus K_0$ such that $C_1 \subseteq C_0$. Therefore we have a natural mapping from the collection of connected components of $X \setminus K_1$ to the connected components of $X \setminus K_0$.

Definition 6.1.11. The resulting diagram admits a limit in **Set** whose elements are called the **ends** of X. They correspond to the order-reversing maps e from $\mathcal{K}(X)$ to the collection of subspaces of X such that $e(K)$ is a connected component of $X \setminus K$. The collection of ends of X is denoted by $\mathcal{E}X$.

The set $X \sqcup \mathcal{E}X$ is equipped with the topology whose open subsets are those U such that $U \setminus \mathcal{E}X$ is open in X and for all ends $e \in U$, there exists some $K \in \mathcal{K}(X)$ such that $e(K) \subseteq U$.

Definition 6.1.12. The preceding topological space is denoted by $\mathcal{F}X$ and called the **Freudenthal extension** of X. In the case where $\mathcal{F}X$ is actually compact, it is called the **Freudenthal compactification** of X.

Remark 6.1.13. According to Definition 6.1.12, $(X \setminus K) \cup \mathcal{E}X$ is a neighborhood of $\mathcal{E}X$ for all *closed* compact sets K of X. This is the main reason why $\mathcal{K}(X)$ only contains *closed* compact subsets instead of all of them. However we will mainly consider Hausdorff spaces, in which any compact subset is necessarily closed.

Remark 6.1.14. Assuming that e is an end of X, the closure (in X) of $e(K)$ is not compact. Otherwise we would have $e(K \cup \mathrm{clo}(e(K))) \subseteq e(K) \subseteq \mathrm{clo}(e(K))$ because e is order reversing. But we would also have $e(K \cup \mathrm{clo}(e(K))) \subseteq X \setminus \mathrm{clo}(e(K))$ by definition of an end. As a consequence, if the connected components of X are compact, then $\mathcal{E}X = \emptyset$ and $\mathcal{F}X = X$. In particular the Freudenthal extension of a topological space may not be compact, yet the construction nicely behaves for a large class of topological spaces.

Example 6.1.15. The Freudenthal extension of a locally compact Hausdorff space in which the complement of any compact subset is connected is its Alexandroff compactification.

Definition 6.1.16. (Steen and Seebach, 1996, p.21) A space X is said to be **σ-locally compact** when there exists a $\subseteq$-nondecreasing sequence $(K_n)_{n\in\mathbb{N}}$ of compact Hausdorff subspaces of X such that

$$X \quad = \quad \bigcup{}^{\uparrow} \mathrm{int}(K_n)$$

According to Definition 6.1.16 a σ-locally compact space is Hausdorff.

Definition 6.1.17. A **generalized continuum** is a σ-locally compact, connected, and locally connected space.

Lemma 6.1.18. The Freudenthal extension of a generalized continuum is compact.

Lemma 6.1.19. Freudenthal extension preserves the disjoint unions.

Theorem 6.1.20 (Haucourt and Ninin)**.** Given a graph G the following are equivalent.

1. The collection of finite unions of connected subsets of $|G|$, denoted by $\mathcal{R}_{|G|}$ in the proof of this theorem, induces a Boolean subalgebra of $\mathrm{Pow}(|G|)$.

2. The graph G has finitely many connected components, all its vertices have finitely many adjacent arrows, and the degree of almost all[1] its vertices is 2. In other words the following sum is finite.

$$\sum_{v \text{ vertex}} |\deg(v) - 2| + \#\{\text{connected components}\}$$

3. The graph G can be obtained as a coequalizer of $D \rightrightarrows L$ with D being finite and discrete, and L being a finite disjoint union of points, segments, and half-lines.

4. The Freudenthal extension of $|G|$ is homeomorphic with the geometric realization of some finite graph.

When the preceding statements are satisfied, the number of ends of $|G|$ is the number of half-lines appearing in L.

Proof. We prove that the first assertion implies the second one. If G has infinitely many connected components then $\mathcal{R}_{|G|}$ has no greatest element. From now on suppose that G is connected.

Suppose some vertex v has infinitely many adjacent arrows and let $S \subseteq |G|$ be an open star centered in v. If $|G| \setminus S$ has infinitely many connected components, then S is a connected subset of $|G|$ whose complement does not belong to $\mathcal{R}_{|G|}$. Otherwise $(|G| \setminus S) \cup \{v\}$ is a finite union of connected components of $|G|$ whose complement, which is the union of infinitely many pairwise disjoint segments, does not belong to $\mathcal{R}_{|G|}$.

Suppose that there are infinitely many vertices whose number of adjacent arrows is not 2 and let T be a covering tree of G (*i.e.* a connected subgraph of G containing all the vertices of G such that T loses its connectedness if one removes a single arrow from it). The set theoretic difference $|G| - |T|$ is thus a disconnected union of segments $B\times]0,1[$ with B being a set of arrows of G. If B is infinite then we have a connected component of $|G|$ whose complement in $|G|$ does not belong to $\mathcal{R}_{|G|}$. Assume that B is finite. It follows that all the vertices v but finitely many ones have the same neighborhood in T than in G. In particular T has infinitely many vertices whose number of adjacent arrows is not 2. From a general fact about trees we deduce that T has infinitely many vertices whose number of adjacent arrows is at least 3. By an easy induction we build a linear subgraph L of T containing infinitely many vertices with (at least) 3 adjacent arrows. The subgraph $L \subseteq T$ is connected and $|T| - |L|$ has infinitely many connected components (at least one for each vertex of L with at least 3 neighbors in T). We note that any connected component of $|G| - |L|$ is actually obtained as the union of connected components of $|T| - |L|$ related by "bridges", that is to say $B'\times]0,1[$ for a subset $B' \subseteq B$. As B was assumed to be finite $|G| - |L|$ has infinitely many connected components. Whether B is finite or not, the collection $\mathcal{R}_{|G|}$ is not a Boolean subalgebra of $2^{|G|}$.

Conversely, since G has finitely many connected components, the collection $\mathcal{R}_{|G|}$ is a sub Boolean algebra of $2^{|G|}$ iff $\mathcal{R}_{|C|}$ is a sub Boolean algebra of $2^{|C|}$ for all the connected components of G. We can thus suppose that G is connected. We write $|G|$ as $D \cup D^{\mathrm{c}}$ with

$$D = \{x \in |G| \mid x \text{ admits a neighborhood that is not isomorphic to } \mathbb{R}\}$$

[1] that is to say all but finitely many ones

First remark that D is a discrete subspace of $|G|$. Also, an element of $|G|$ belongs to D iff its degree is not 2. Therefore, by hypothesis, D is finite. Because G is a connected graph, its realization $|G|$ is a connected space. On the contrary D^{c} (*i.e.* the complement of D in $|G|$) is a disconnected union of copies of $\mathbb{R}$ (in fact $]0,1[$ according to our description of $|G|$). Let us consider their boundaries in $|G|$. If one of them has an empty boundary, then it is both open and closed, and disconnected from its complement in $|G|$. Therefore $|G| \cong \mathbb{R}$. Suppose now that the boundary of any connected component of D^{c} contains at least one element. This element belongs to D. For each $v \in D$ and each connected component C of D^{c} whose boundary contains v, the degree of v is augmented by at least 1 (if $C \cup \{v\} \cong \mathbb{R}_+$) and at most 2 (if $C \cup \{v\} \cong \mathbb{S}^1$). Hence D^{c} has finitely many connected components, let us say $C_1, \ldots, C_n$. In order to conclude, remark that a finite union of connected components of $|G|$ can be written as a disjoint union

$$D' \cup X_1 \cup \cdots \cup X_n$$

where $D' \subseteq D$ and X_k is a finite union of intervals of C_k.

Suppose that the second point is satisfied. Then consider the graph G' obtained from G as follows: G and G' have the same set of isolated vertices (*i.e.* those with null degree), they share their set of arrows and we have $\partial^{-}\alpha' = \partial^{+}\alpha$ in G' iff the same holds in G and $\deg \partial^{+}\alpha = 2$ (in G). As a consequence the degree of a vertex in G' does not exceed 2 so G' is a disjoint union of linear graphs. In particular we have a canonical morphism from G' to G which is entirely defined by the fact that it is the identity map on arrows and vertex of zero degree. Moreover the number of connected components of G' is finite because it is less than

$$\#\big\{\text{linear connected components of } G\big\} \;+ \sum_{\substack{v \text{ vertex s.t.}\\ \deg_G(v) \neq 2}} \deg_G(v)\,.$$

Some connected component of G' may be a circle or a line, yet both can be obtained as the coequalizer of the form $\{0,1\} \rightrightarrows L''$ with L'' being a segment in the former case, and the disjoint union of two half lines in the latter.

Conversely, consider a graph G obtained as the coequalizer of f and g as in the third assertion. The coequalizer morphism induces a map from the connected components of L' onto the connected components of G so there are finitely many of them. So the vertices of G are the classes of the least equivalence relation over the vertices of L' that contains

$$\big\{(f(x), g(x)) \,\big|\, x \text{ vertex of } L'\big\}$$

Since D is finite there are finitely many classes that are not reduced to a singleton (hence G has finitely many vertices whose degree differs from 2), and each class is finite (hence the degree of each vertex of G is finite).

Suppose that some (and then all) of the first three statements is (are) satisfied. From Lemma 6.1.2 we know that $|G|$ is locally compact. As a left adjoint, the realization functor preserves the coequalizer given by the third statement so any copy of the half-line in L gives rise to a copy of $\mathbb{R}_+$. Let n be the number of copies of the half-line in L. Given $k \in \mathbb{N}$ consider G_k the coequalizer of $D \rightrightarrows L_k$ with L_k being obtained from L by keeping the k-length initial or final segment of every half-line of K. Then

$$|G_0| \subset \cdots \subset |G_k| \subset |G_{k+1}| \subset \cdots$$

forms an exhaustion[2] of $|G|$, and for k sufficiently large, $|G| \setminus |G_k|$ is homeomorphic

[2] a $\subset$-increasing family of compact subsets that covers the whole space

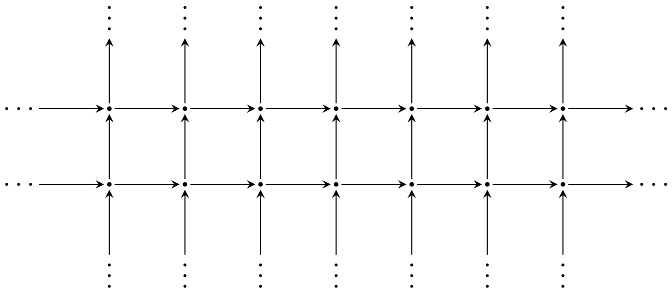


Figure 6.4: The infinite grid.

with n copies of the real line $\mathbb{R}$. It follows that for k sufficiently large, the Freudenthal extension of $|G|$ is homeomorphic with $|G_k|$.

Suppose that the fourth statement is satisfied. Let ϕ be some embedding of $|G|$ in $|G'| \cong \mathcal{F}|G|$ with G' a finite graph. Following Remark 6.1.5 one has $\deg(x) = \deg(\phi(x))$ so the degrees of all the points of $|G|$ (and thus all the vertices of G) are finite. Moreover ϕ induces an embedding of the discrete subspace $\{x \in |G| \mid \deg(x) \neq 2\}$ into the discrete subspace $\{x \in |G'| \mid \deg(x) \neq 2\}$ which is finite. Therefore G has finitely many vertices with a degree that differs from 2. By Lemma 6.1.19 we conclude that $|G|$ (and therefore G) has finitely many connected components. □

Remark 6.1.21.
Any graph satisfying one of the assertions of Theorem 6.1.20 is countable.

Remark 6.1.22. Following Definition 6.1.12, if G satisfies one of the assertions of Theorem 6.1.20, $\mathcal{F}|G|$ is called the **Freudenthal compactification** of $|G|$.

Example 6.1.23. The **infinite grid** is the graph G depicted on Figure 6.4. Formally, its set of vertices is $\mathbb{Z} \times \{0, 1\}$ with one arrow from (a, b) to (c, d) iff $a \leqslant c$, $b \leqslant d$, and $(c - a) + (d - b) = 1$. Its geometric realization can be embedded into the plane.

$$|G| \quad \cong \quad \bigcup_{n\in\mathbb{Z}} \big(\mathbb{R} \times \{n\} \ \cup \ \{n\} \times \mathbb{R} \big) \quad \subseteq \quad \mathbb{R}^2$$

As a consequence of Example 6.1.15, the Freudenthal extension of the (geometric realization of the) infinite grid is compact and has a single end. It is not locally simply connected since no neighborhood of ∞ is simply connected. Hence by Lemma 6.1.1, it cannot be the realization of a graph.

Example 6.1.24. The **infinite comb** is the graph G depicted on Figure 6.5. Formally, its set of vertices is $\mathbb{Z} \times \{0, 1\}$ with one arrow from $(n, 0)$ to $(n + 1, 0)$ (resp. $(n, 1)$) for all $n \in \mathbb{Z}$. Its geometric realization can be embedded into the plane.

$$|G| \quad \cong \quad \left(\mathbb{R} \times \{0\} \quad \cup \quad \bigcup_{n\in\mathbb{Z}} \{n\} \times [0, 1] \right) \quad \subseteq \quad \mathbb{R}^2$$

The Freudenthal extension of its geometric realization is compact and has two ends. However it is not homeomorphic to the geometric realization of a graph. Indeed an open star S (with at least 3 branches) contains at most one point p such that $V - \{p\}$

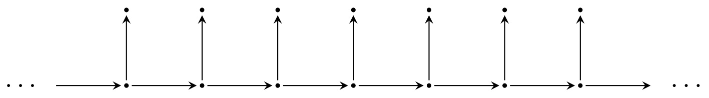
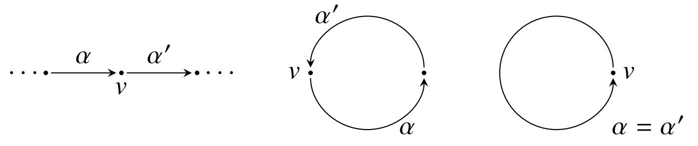


Figure 6.5: The infinite comb



two neighbors one neighbor no neighbor

Figure 6.6: Vertices v with a single outgoing arrow and a single ingoing arrow

has at least 3 connected components. Hence no neighbourhood of an end of $\mathcal{F}(|G|)$ is homeomorphic to an open star.

Remark 6.1.25. Concerning the third point of the Theorem 6.1.20, note that the finiteness hypothesis on D cannot be dropped. Consider indeed the case where D is the set of vertices of a line L. Then let the first morphism of the coequalizer diagram be the inclusion of D in L while the second one sends all the elements of D to a single point. The coequalizer is a graph with a single vertex and infinitely many arrows.

The next definition and the remaining of this section are related to the notion of systems of weak isomorphisms (*cf.* Definition 8.2.4). Intuitively, we want to discard the vertices of a graph around which there is no branching. In the directed topology realm, the open stars (*cf.* Example 4.1.15) provide the prototypical examples of branchings, except for St_1^1 which is isomorphic to $\mathbb{R}$. The definition below should be understood with Figure 6.1 in mind.

Definition 6.1.26. The **branching degree** $\deg_b(v)$ of a vertex v is defined as

$$|\#\{v\text{-ingoing arrows}\} - 1| + |\#\{v\text{-outgoing arrows}\} - 1|$$

The branching degree of a graph is the sum of the branching degrees of all its vertices. A vertex is said to be **expandable** when it has at least one neighbor and its branching degree is null – see Figure 6.6.

Taking the direction into account, we sharpen the classification of linear graphs given by Figure 6.3.

Definition 6.1.27. The n**-cycle**, for $n \geqslant 1$, is the graph whose vertices are $\{0, \ldots, n-1\}$ with an arrow from k to $k+1$ modulo n, a graph that is isomorphic to some n-circle with $n \geqslant 1$ is said to be **cyclic**. Then a finite connected graph is cyclic iff the branching degrees of all its vertices are null. When the context is clear, the following graph is denoted by $\mathbb{Z}$

A graph that is isomorphic to some connected subgraph of $\mathbb{Z}$ is called a **chain**. A chain is said to be **proper** when it not isomorphic to $\mathbb{Z}$.

Remark 6.1.28. The set of nonexpandable vertices with null branching degree in a graph G is (up to the obvious bijection) the set of connected components of G that are isomorphic to the 1-cycle.

Remark 6.1.29. Any chain (resp. cycle) is a point, a segment, a half-line or a line (resp. a circle) yet the converse is false (*e.g.* Figure 6.1). Note that $\mathbb{Z}$ is the only connected infinite graph all the vertices of which are expandable.

As suggested by the terminology we are going to remove all the expandable vertices from a connected graph G yet preserving the "branching structure" of the graph. Let $\mathrm{Expnd}(G)$ be the full subgraph of G containing all its expandable vertices.

Remark 6.1.30. The graph $\mathrm{Expnd}(G)$ is a disjoint union of chains and cycles. The cycles, and the chains of $\mathrm{Expnd}(G)$ that are isomorphic to $\mathbb{Z}$, are connected components of G.

Following Remark 6.1.30 we should think of the finite chains of $\mathrm{Expnd}(G)$ as bridges between the non expandable vertices of G. However we also have to care for the cycles and the infinite chains.

Definition 6.1.31. The **reduced** graph of a connected graph G, denoted by $\mathrm{red}(G)$, is defined as follows:

– $\mathrm{red}(\mathbb{Z}) \cong \{\cdot \longrightarrow \cdot\}$ and the reduced graph of any cycle is the 1-cycle,

– if G is neither cyclic nor isomorphic to $\mathbb{Z}$, then any connected component C of $\mathrm{Expnd}(G)$ is either a finite chain, $\mathbb{N}$, or $\mathbb{N}^{\mathrm{op}}$. Then $\mathrm{red}(G)$ is defined as follows: for all connected components C of $\mathrm{Expnd}(G)$,

  - if $C \cong \mathbb{N}$ (resp. $C \cong \mathbb{N}^{\mathrm{op}}$) then remove[3] from G all the vertices appearing in C but the first (resp. last) one.
  - if C is a finite chain, there is a unique arrow from some (necessarily unique) vertex C^- of G to the first element of C and a unique arrow from the last element of C to some (necessarily unique) vertex C^+ of G. Then remove from G all the vertices appearing in C and add an arrow from C^- to C^+.

By extension, is G is not connected, then $\mathrm{red}(G)$ is the disjoint union of its reduced connected components.

Example 6.1.32. The reduced graph of a chain is a point iff the chain is a point. Otherwise it is $\{\cdot \longrightarrow \cdot\}$. Also note that $\mathrm{red}(\emptyset) = \emptyset$.

Example 6.1.33. The infinite zigzag

$$\cdots \longrightarrow \bullet \longleftarrow \bullet \longrightarrow \bullet \longleftarrow \bullet \longrightarrow \bullet \longleftarrow \bullet \longrightarrow \bullet \longleftarrow \bullet \longrightarrow \bullet \longleftarrow \cdots$$

is connected, the degree of all is vertices are 2, and its geometric realization is $\mathbb{R}$. Then it matches the statements of Theorem 6.1.20. However, from a directed point of view it would not be wise to consider it as "almost finite".

[3] Note that removing a vertex from a graph implies that all the adjacent arrows are also removed.

We provide an alternative description of the reduced graph, based on the directed geometric realization, which explains the construction. Given the disjoint sets a and b define G_a^b as the connected graph whose arrows are

$$\big\{(k,0)\;\big|\;k\in a\big\}\;\cup\;\{(0,k)\mid k\in b\}$$

where 0 neither belongs to a nor b. The directed open star St_a^b is defined as $\rceil G_a^b\lfloor\backslash(a\cup b)$ (compare with the definition given in Section 4.3). The following two results are easy yet they are the cornerstones to determine the isomorphisms between open stars.

Lemma 6.1.34. The open star St_a^b has a least (resp. greatest) element iff a is empty (resp. b is empty)

Lemma 6.1.35. Given $x\in\mathsf{St}_a^b$ t.f.a.e.

1. $\mathsf{St}_a^b\setminus\{x\}$ is a disjoint sum of copies of $\mathbb{R}$
2. all the open connected neighborhoods of x are isomorphic to St_a^b

Lemma 6.1.36. If a (resp. b) is not a singleton then 0 is the unique point of St_a^b satisfying the properties 1) and 2) of Lemma 6.1.35. It is referred to as the **node** of the star.

Proposition 6.1.37. The open stars St_a^b and $\mathsf{St}_{a'}^{b'}$ are isomorphic iff $a\cong a'$ and $b\cong b'$. Moreover if one (at least) of the sets a and b is not a singleton then

$$\mathrm{Aut}\,(\mathsf{St}_a^b)\cong(\mathrm{Aut}\,\mathbb{R})^{a\cup b}\rtimes\big(\mathfrak{S}_a\times\mathfrak{S}_b\big)$$

otherwise both a and b are singletons and

$$\mathrm{Aut}\,(\mathsf{St}_a^b)\cong\mathrm{Aut}\,\mathbb{R}\cong\{\text{nondecreasing mappings from }\mathbb{R}\text{ onto }\mathbb{R}\}$$

Proof. Let Φ be an isomorphism from St_a^b to $\mathsf{St}_{a'}^{b'}$. If a and b are singletons then, for any $t\in\mathbb{R}$, $\mathbb{R}\setminus\{t\}$ has exactly two connected components. Counting the connected components of $\Phi(\mathsf{St}_a^b)\setminus\{\Phi(t)\}$, we deduce that $(\#a',\#b')$ is either $(0,2)$, $(2,0)$, or $(1,1)$. From Lemma 6.1.34 we conclude that $(\#a',\#b')=(1,1)$. Otherwise Lemma 6.1.36 applies and $\Phi(0)=0$. As a consequence Φ induces a d-space isomorphism Φ' between the disjoint sums $(a\cup b)\times\mathbb{R}$ and $(a'\cup b')\times\mathbb{R}$. It follows that $a\cup b\cong a'\cup b'$ and there is a permutation $\sigma\in\mathfrak{S}_{a\cup b}$ and a collection $\{f_k\mid k\in a\cup b\}$ of d-space automorphisms of $\mathbb{R}$ such that $\Phi'(k,t)=(\sigma(k),f_{\sigma(k)}(t))$. Moreover for $k\in a$, $\Phi((\{k\}\times\mathbb{R})\cup\{0\})$ is a d-subspace of $\mathsf{St}_{a'}^{b'}$ which is isomorphic to $\mathbb{R}^+$. Since Φ preserves nodes we have $\Phi(\{k\}\times\mathbb{R})=\{k'\}\times\mathbb{R}$ for some $k'\in a'$. Then we have a one-to-one mapping from a to a'. Repeating the proof with Φ^{-1} instead of Φ we deduce that $a\cong a'$ and $\sigma(a)=a'$. In the same way, we obtain that $b\cong b'$ and $\sigma(b)=b'$. □

If a and b are finite we write $\mathsf{St}_{\#a}^{\#b}$ instead of St_a^b. The following result is obvious from the description of $\rceil G\lfloor$.

Lemma 6.1.38. Let x be a point of $\rceil G\lfloor$ for some graph G. There exists an open connected neighborhood of x that is isomorphic to an open star, moreover two such neighborhoods are isomorphic.

Definition 6.1.39. As a consequence of Proposition 6.1.37 and Lemma 6.1.38 we define the **type** of a point $x\in\rceil G\lfloor$ as the pair (a,b) such that some neighborhood of x is isomorphic to St_a^b.

The next result provides an alternative definition of the reduced graph in terms of directed geometric realization.

Proposition 6.1.40. Given a graph G we have a partition $\rceil G\lfloor = D \cup D^{\mathrm{c}}$ where

$$D = \{x \in \rceil G\lfloor \ \mid \text{type of } x \text{ is not } (1,1)\}$$

Then D is a discrete subspace of $\rceil G\lfloor$ and $\rceil G\lfloor\backslash D$ is a disjoint union of connected components which are all isomorphic to the d-spaces $\mathbb{R}$ or $\mathbb{S}^1$. We denote the topological closure of $X \subseteq \rceil G\lfloor$ endowed with the inherited d-space structure by $\mathrm{clo}(X)$. The graph $\mathrm{red}(G)$ can be described as below. Its vertices are

- all the elements of D, plus
- all the connected components C of $\rceil G\lfloor\backslash D$ such that $\mathrm{clo}(C)$ is isomorphic to one of the d-spaces $\mathbb{S}^1$, $\mathbb{R}_+$, and $\mathbb{R}_-$, plus
- two copies of each connected component C of $\rceil G\lfloor\backslash D$ such that $\mathrm{clo}(C)$ is isomorphic to the d-space $\mathbb{R}$.

Its arrows are all the connected components of $\rceil G\lfloor\backslash D$. Given such a component C:

- if $\mathrm{clo}(C) \cong \mathbb{S}^1$ then its source and its target are C itself.
- if $\mathrm{clo}(C) \cong \mathbb{R}_+$ (resp. $\mathbb{R}_-$) then its source (resp. target) is the unique element of the boundary of $\mathrm{clo}(C)$; the target (resp. source) is C itself.
- if $\mathrm{clo}(C) \cong \mathbb{R}$ then its source and its target are the two vertices associated with C. A choice has to be made here which anyway does not change the resulting graph, up to isomorphism.

Proof. Suppose that the type of x, denoted by (a,b), differs from $(1,1)$. By Lemma 6.1.38 we have an open connected neighborhood U of x that is isomorphic to St_a^b. Since the type of x is not $(1,1)$ then all points of U but x have type $(1,1)$. So x is isolated in D. Suppose $D \neq \emptyset$. Then any connected component of $\rceil G\lfloor \backslash D$ can be written as

$$\bigcup_{j=1}^{n} \{\alpha_j\} \times [0,1] \setminus \{\partial^-\alpha_1, \partial^+\alpha_n\}$$

with $\alpha_1, \ldots, \alpha_n$ an acyclic path on the graph G, and $\partial^-\alpha_1$ and $\partial^+\alpha_n$ in D. All sources and targets of the arrows $\alpha_1, \ldots, \alpha_n$ but $\partial^-\alpha_1$ and $\partial^+\alpha_n$ have a single predecessor and a single successor (otherwise their type in $\rceil G\lfloor$ would not be $(1,1)$). In other words they are expandable. □

Corollary 6.1.41. Let G be a connected graph. If G is not circular then

$$\mathrm{Aut}\ \rceil G\lfloor \ \cong\ (\mathrm{Aut}\,\mathbb{R})^n \rtimes \mathrm{Aut}\,\big(\mathrm{red}(G)\big)$$

with n being the number of arrows of $\mathrm{red}(G)$.

Proof. From Proposition 6.1.40 we know that any automorphism of $\rceil G\lfloor$ induces an automorphism of $\mathrm{red}(G)$. In particular it induces a bijection from the set of arrows of $\mathrm{red}(G)$ to itself, these arrows being precisely the connected components of $\rceil G\lfloor\backslash D$. Since G is connected and noncircular, they are all isomorphic to $\mathbb{R}$. □

Still following (Haucourt and Ninin (2014)), we refine Theorem 6.1.20 assuming that the directed Freudenthal extension is understood in $\mathbf{dTop}_{\mathrm{c}}$ – see Remark 4.5.36. Also remember that the directed realization of graphs is indifferently defined in most full subcategories of **dTop** – see Proposition 6.1.7 so we write $\rceil G\lfloor$ without subscript.

Theorem 6.1.42. Given a graph G the following are equivalent.

1. The reduced graph of G is finite.
2. The graph G has finitely many connected components, all its vertices have a finite branching degree, and all its vertices but finitely many ones are expandable. In other words the following sum is finite
$$\sum_{v \text{ vertex}} \deg_b(v) + \#\{\text{connected components}\}$$
3. The graph G is the amalgamated sum of finitely many proper chains over a finite discrete graph.
4. The directed Freudenthal extension of $\rceil G\lfloor$ is isomorphic to the directed geometric realization of some finite graph.

When the preceding statements are satisfied we have
$$\mathcal{F}\rceil G\lfloor \cong \rceil \mathrm{red}(G)\lfloor$$
and the number of ends of $\rceil G\lfloor$ is the number of proper chains mentioned in the third statement that are isomorphic to $\mathbb{N}$ or $\mathbb{N}^{\mathrm{op}}$.

Proof. The reformulation of the second claim is a consequence of Remark 6.1.28 and the second statement readily derives from the first one and Definition 6.1.31.

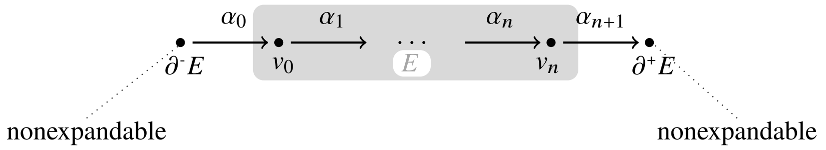
Suppose that the second assertion is satisfied. Then red(G) has finitely many connected components so we can suppose that G is connected by Definition 6.1.31. Each connected component of Expnd(G) increases the branching degree of the graph by 1, so there are finitely many of them. By Definition 6.1.31 red(G) is thus finite.

Suppose that the third statement holds. Then note that the number of connected components of G is bounded by the number of proper chains, and also that the branching degree of the graph is bounded by the cardinal of the finite discrete graph.

Following the construction described in Definition 6.1.31, the first statement implies the third one.

Suppose that the fourth assertion is true. From the embedding
$$\rceil G\lfloor \hookrightarrow \mathcal{F}\rceil G\lfloor \cong \rceil G'\lfloor$$
we deduce that $\rceil G\lfloor$ has finitely many connected components, all the points of $\rceil G\lfloor$ have type (a, b) with finite a, b, and all but finitely many of them have type other than $(1, 1)$. In other words the second statement is satisfied.

Conversely, $U(\rceil \mathrm{red}(G)\lfloor) \cong |\mathrm{red}(G)|$ which is compact since red(G) is finite (because G matches Theorem 6.1.42). We can suppose that G is connected and then describe a directed embedding $\Phi : \rceil G\lfloor \to \rceil \mathrm{red}(G)\lfloor$ whose image is dense. First put $\Phi(v) = v$ and $\Phi(\alpha, t) = (\alpha, t)$ for all nonexpandable vertices v and all arrows α whose extremities are not expandable. Then let $E \in$ Expnd(G). If E is finite with set of vertices $\{v_0, \ldots, v_n\}$ and set of arrows $\{\alpha_1, \ldots, \alpha_n\}$, we define α_0 (resp. α_{n+1}) as the unique v_0-ingoing (resp. v_n-outgoing) arrow:



Then:

- we choose a finite strictly increasing sequence $0 < t_0 < \cdots < t_n < 1$ of points of $]0,1[$,
- we set $\Phi(v_n) = t_n$ for all $k \in \{0,\ldots,n\}$, and
- we define Φ on $\{\alpha_k\}\times]0,1[$ as the unique increasing affine mapping to the open segment $\{\alpha_E\}\times]t_{k-1},t_k[$ with the convention that $k \in \{0,\ldots,n+1\}$, $t_{-1} = 0$, $t_{n+1} = 1$, and α_E is the arrow of $\mathrm{red}(G)$ corresponding to E.

If E is infinite then it is either isomorphic to $\mathbb{N}$ or $\mathbb{N}^{\mathrm{op}}$. The preceding construction still applies replacing the finite increasing sequence $0 < t_0 < \cdots < t_n < 1$ by an infinite one that converges to 1 (or decreasing to 0 in the second case). Proving that Φ induces a dihomeomorphism (*cf.* Definition 4.2.2) on its image is a routine verification. Moreover the only points of $\rceil\mathrm{red}(G)\lfloor$ that are not reached by Φ are the vertices corresponding to each infinite $E \in \mathrm{Expnd}(G)$. Hence the image of Φ is dense in its codomain. Doing so we have proven that $\rceil\mathrm{red}(G)\lfloor$ is a directed compactification of $\rceil G\lfloor$. We still have to check that it is the Freudenthal one. To do so we refer to Proposition 6.1.40 which describes $\mathrm{red}(G)$ in terms of $\rceil G\lfloor$. The ends of $\rceil G\lfloor$ indeed correspond to the connected components of $\rceil G\lfloor \setminus D$ (with the notation of the proof of Proposition 6.1.40) whose closure in $\rceil G\lfloor$ are isomorphic to $\mathbb{R}$, $\mathbb{R}_+$, or $\mathbb{R}_-$: each component of the first kind gives rise to two ends while every component of the second and third ones gives rise to a single end. Indeed, when G satisfies the statements of the theorem, any compact subset of $\rceil G\lfloor$ is contained in a larger one whose complement is isomorphic to n copies of $\mathbb{R}$ with $n \in \mathbb{N}$ only depending on G. □

It is worth noticing that the preceding result would not have been true if we had considered the Freudenthal extension of a d-space X endowed with the least direction containing X instead of the least complete direction. Indeed the directed paths arriving at or starting from a vertex corresponding to some infinite $E \in \mathrm{Expnd}(G)$ would have been constant.

Definition 6.1.43. Any graph satisfying one of (therefore all) the assertions of Theorem 6.1.42 is said to be **essentially finite**.

Remark 6.1.44. Any essentially finite graph satisfies the statements of Theorem 6.1.20.

Proposition 6.1.45.
Given the essentially finite graphs G and G', the directed Freudenthal compactifications of $\rceil G\lfloor$ and $\rceil G'\lfloor$ are isomorphic iff $\mathrm{red}(G) \cong \mathrm{red}(G')$.

Proof. Since $\rceil\_\lfloor$ is functorial $\mathrm{red}(G) \cong \mathrm{red}(G')$ implies $\rceil\mathrm{red}(G)\lfloor\cong\rceil\mathrm{red}(G')\lfloor$ hence $\mathcal{F}\rceil G\lfloor \cong \mathcal{F}\rceil G'\lfloor$ by Theorem 6.1.42. The converse implication comes from the description of the reduced graph provided by Proposition 6.1.40. □

What precedes is to be compared to (Nadler Jr., 1992, Chap.IX) which defines graphs as union of finitely many copies of $[0, 1]$ with some of their endpoints identified. Then observe that the objects of a framework for directed topology (*cf.* Section 4.2) arising as the realization of some graph can be characterized in Continuum Theory terminology[4] – see (Nadler Jr., 1992, Th.9.10 p.144). Also note that Diestel and Kühn (2003) thoroughly compares the notion of ends depending on whether they are expressed in graph theoretic or topological terms.

6.2 Block Coverings

As we will see in Section 7.1, certain parallel programs can be analyzed handling subsets of $\rceil G\lfloor^n$ where $n \in \mathbb{N}$. The powerset of $\rceil G\lfloor^n$ is untractable in practice, fortunately there exists a subcollection of it that contains the models of all the programs in our scope, and that can be handled by computers. Indeed we only need to take into account the subsets of $\rceil G\lfloor^n$ of the form

$$\bigcup_{i=1}^{k}\prod_{j=1}^{n} a_{i,j}$$

with $a_{i,j}$ being taken in a distinguished collection of subsets of $\rceil G\lfloor$. As a preamble, let us describe an abstract situation to which we will go back in Section 9.5. The powerset of a set E is denoted by $\mathrm{Pow}(E)$. Given a set X, a **field of sets** over X is a Boolean subalgebra of powerset $\mathrm{Pow}(X)$. Given a finite family $\mathcal{B}_1, \ldots, \mathcal{B}_n$ ($n \in \mathbb{N}$) of fields of sets over the sets $X_1, \ldots, X_n$, it is natural to take an interest in the field of sets generated by the subsets of the Cartesian product $X_1 \times \cdots \times X_n$ of the form $\mathrm{proj}_i^{-1} A_i$ where $A_i \in \mathcal{B}_i$ and $\mathrm{proj}_i : X_1 \times \cdots \times X_n \longrightarrow X_i$ is the i^{th} projection. This well-known construction (Givant and Halmos, 2009, Chap. 44) actually provides the coproduct of $\mathcal{B}_1, \ldots, \mathcal{B}_n$ in the category **BoolAlg** of Boolean algebras. The category of Boolean algebras is actually cocomplete though its coproducts are usually called **free products** by the experts of the domain (Koppelberg, 1989, Chap.4.11, p.157-158). We will go back to the theoretical aspects of that construction in Section 9.5. The purpose of the present one is to provide a canonical representation for the elements of that field of sets. For example, in the case where $n = 2$ and $X_1 = X_2 = \mathbb{R}$, its elements are precisely the finite unions of rectangles of the plane $\mathbb{R}^2$. Rectangles should be understood in the broad sense here, namely as the Cartesian product of two non-empty connected subsets of $\mathbb{R}$.

Definition 6.2.1. Let $\mathcal{B}$ be a Boolean subalgebra of $\mathrm{Pow}(E)$ and let n be a natural number. A **$\mathcal{B}$-block** of dimension n is a subset of E^n of the form $B_1 \times \cdots \times B_n$ with $B_i \in \mathcal{B}$ and $B_i \neq \emptyset$ for all $i \in \{1, \ldots, n\}$. An **isothetic $\mathcal{B}$-region** of dimension $n \in \mathbb{N}$ is a subset of E^n that can be written as a finite union of n-dimensional $\mathcal{B}$-blocks. When the context is clear, we omit the prefix referring to $\mathcal{B}$. A routine verification shows that the collection $\mathcal{R}_n$ of n-dimensional isothetic regions forms a Boolean subalgebra of $\mathrm{Pow}(E^n)$. The maximal elements of the collection of blocks contained in $X \in \mathcal{R}_n$ are called the **maximal blocks** of X.

The next two lemmas easily derive from the following exchange law for $i \in \{1, \ldots, n\}$ and $B_i, B'_i \in \mathcal{B}$:

$$(B_1 \times \cdots \times B_n) \cap (B'_1 \times \cdots \times B'_n) \quad = \quad (B_1 \cap B'_1) \times \cdots \times (B_n \cap B'_n)$$

[4] A continuum is a Hausdorff compact space

Lemma 6.2.2. The maximal blocks of the complement of a block $B_1 \times \cdots \times B_n$ are the blocks $E^{i-1} \times B_i^c \times E^{n-i}$ for $i \in \{1, \ldots, n\}$. Moreover any block of this complement is contained in a maximal block.

Lemma 6.2.3. Let X and Y be subsets of E^n having finitely many maximal blocks. If each block of the subset X (resp. Y) is contained in a maximal block of X (resp. Y) then each block of the intersection $X \cap Y$ is contained in a maximal block of $X \cap Y$. Moreover, all the maximal blocks of $X \cap Y$ have the form $B \cap B'$ for some maximal blocks B and B' of X and Y, hence there are finitely many of them.

Proof. By hypothesis a block B of $X \cap Y$ is contained in $B_X \cap B_y$ for some maximal blocks B_X and B_Y of X and Y. By the exchange law, the intersection $B_X \cap B_y$ is a block of $X \cap Y$. Because X and Y have finitely many maximal blocks, we can suppose that $B_X \cap B_Y$ is maximal. □

The next lemma characterizes the elements of $\mathcal{R}_n$ in terms of their maximal blocks. It is a key ingredient in the proof of Theorem 9.5.26.

Lemma 6.2.4. A subset of E^n belongs to $\mathcal{B}_n$ iff its has finitely many maximal blocks and their union covers it.

Proof. By the De Morgan's law any element of $\mathcal{B}_n$ can be written as an intersection of complement of blocks. By Lemma 6.2.2 each block of the complement of a block is contained in a maximal block and there are finitely many of them. The conclusion follows from Lemma 6.2.3. □

Definition 6.2.5. A **block covering** of $X \in \mathcal{R}_n$ is a finite collection $\mathcal{F}$ of blocks whose union is X. A block covering is thus a finite family of blocks. We denote by $\mathrm{FinCov}_n(\mathcal{B})$ the collection of all n-block coverings. We define the **covering preorder** writing $\mathcal{F} \preccurlyeq_c \mathcal{F}'$, for block coverings $\mathcal{F}$ and $\mathcal{F}'$, to mean that any element of $\mathcal{F}$ is included in an element of $\mathcal{F}'$. By Lemma 6.2.4 one soundly defines the **maximal block covering** of X as the collection of its maximal blocks. It is the greatest element of the collection of block coverings of X with respect to $\preccurlyeq_c$. We also define the **gathering preorder** writing $\mathcal{F} \preccurlyeq_g \mathcal{F}'$ to mean that the union of the elements of $\mathcal{F}$ is included in that of the elements of $\mathcal{F}'$. Of course $\mathcal{F} \preccurlyeq_c \mathcal{F}'$ implies that $\mathcal{F} \preccurlyeq_g \mathcal{F}'$ but the converse is false.

Remark 6.2.6. Let $\mathcal{F}$ and $\mathcal{F}'$ be two maximal block coverings, the following are equivalent:

- $\mathcal{F} = \mathcal{F}'$ (*i.e.* both languages contain the same words)
- $\mathcal{F}$ and $\mathcal{F}'$ are equivalent with respect to the gathering preorder (*i.e.* $\bigcup \mathcal{F} = \bigcup \mathcal{F}'$)
- $\mathcal{F}$ and $\mathcal{F}'$ are equivalent with respect to the covering preorder (*i.e.* any connected block of one of the languages $\mathcal{F}$ and $\mathcal{F}'$ is contained in some connected block of the other).

Proposition 6.2.7. Assuming that $\mathrm{FinCov}_n(\mathcal{B})$ is equipped with the covering preorder, we have a Galois connection

$$\begin{tikzcd}
\mathrm{FinCov}_n(\mathcal{B}) \arrow[r, shift left, "\gamma_n"] & \mathcal{R}_n \arrow[l, shift left, "\alpha_n"]
\end{tikzcd}$$

defining $\gamma_n(\mathcal{F})$ as $\bigcup \mathcal{F}$ and $\alpha_n(X)$ as the maximal block covering of X. In particular $\gamma_n \circ \alpha_n = \mathrm{id}$ and $\alpha(\emptyset)$ is the empty family. The Galois connection becomes an isomorphism of Boolean algebras if one substitutes $\mathrm{FinCov}_n(\mathcal{B})$ with the image of α_n (*i.e.* the collection of all maximal block coverings).

Proof. The monotonicity of γ_n exactly means that the covering preorder is finer that the gathering one (*cf.* Definition 6.2.5). The monotonicity of α_n and the relation $\gamma_n \circ \alpha_n = \mathrm{id}$ are immediate consequences of Lemma 6.2.4. By Remark 6.2.6 the restriction of the relation $\preccurlyeq_c$ to the image of α_n induces a partial order that is isomorphic to that of the Boolean structure of $\mathcal{B}$, namely the inclusion relation inherited from $\mathrm{Pow}(E)$. □

The covering and the gathering preorders are tightly related: one has $\mathcal{F} \preccurlyeq_g \mathcal{F}'$ if and only if there exist coverings $\mathcal{F}''$ and $\mathcal{F}'''$ such that $\bigcup \mathcal{F} = \bigcup \mathcal{F}''$, $\bigcup \mathcal{F}' = \bigcup \mathcal{F}'''$, and $\mathcal{F}'' \preccurlyeq_c \mathcal{F}'''$ (*cf.* Proposition 6.2.7).

How far the powerset $\mathrm{Pow}(E^n)$ is captured by $\mathcal{R}_n$ directly depends on the choice of $\mathcal{B}$. If $\mathcal{B}$ is reduced to the collection $\{E, \emptyset\}$, then $\mathcal{R}_n$ is reduced to the collection $\{E^n, \emptyset\}$, which is too coarse. On the other hand, setting $\mathcal{B} = \mathrm{Pow}(E)$ does not lead to a tractable Boolean subalgebra of $\mathrm{Pow}(E^n)$.

A striking fact is that replacing $\mathcal{B}$ by a collection of subsets of E which behaves like the collection of connected subsets of a topological space, one obtains a result that is similar to Proposition 6.2.7.

Definition 6.2.8. A **connectology** over a set E is a collection $C \subseteq \mathrm{Pow}(E)$, whose elements are said to be **connected**, containing the empty set and all the singletons, and such that for all $\mathcal{F} \subseteq C$, if $\mathcal{F} \neq \emptyset$ and $\bigcap \mathcal{F} \neq \emptyset$, then $\bigcup \mathcal{F} \in C$.

Remark 6.2.9. As one can expect, the connected subsets of a topological space form a connectology. In particular C is stable under monotonic union and the family of all connected subsets containing a given connected subset C admits a greatest element, actually its union, which is called the **connected component** of C.

Connectologies can be defined in many other ways:

Lemma 6.2.10. Given $C \subseteq \mathrm{Pow}(\upharpoonleft G \downharpoonright)$, consider the following assertions

1. $\forall \mathcal{F} \subseteq C,\ \bigcap \mathcal{F} \neq \emptyset \ \Rightarrow\ \bigcup \mathcal{F} \in C$
2. $\forall \mathcal{F} \subseteq C,\ \big((\forall F, F' \in \mathcal{F},\ F \cap F' \neq \emptyset) \ \Rightarrow\ \bigcup \mathcal{F} \in C\big)$
3. $\forall \mathcal{F} \subseteq C,\ \big((\forall F, F' \in \mathcal{F},\ F \cup F' \in C) \ \Rightarrow\ \bigcup \mathcal{F} \in C\big)$
4. $\forall F, F' \in C,\ F \cap F' \neq \emptyset \ \Rightarrow\ F \cup F' \in C$

then 1. ⇔ 2. ⇔ (3. and 4.)

Proof. The second point clearly implies the first one. Conversely, define $\mathcal{M}$ as the set of maximal elements of

$$\{A \in C \mid A \subseteq \bigcup \mathcal{F}\}$$

From the first point one deduces that for all $M \in \mathcal{M}$ and all $F \in \mathcal{F} \cup \mathcal{M}$ one has $F \subseteq M$ or $F \cap M = \emptyset$. Let $M, M' \in \mathcal{M}$ and $F, F' \in \mathcal{F}$ that meet respectively M and M'. By hypothesis we have $F \cap F' \neq \emptyset$ and therefore $M = M'$.
We prove the same way that the first point implies the third one, it suffices to change the last argument to " by hypothesis we have $F \cup F' \in C$ and thus $M = M'$ ". □

Remark 6.2.11. The third point does not imply the second one: consider for example the collection of chains of a poset in which there are two noncomparable points with a common lower bound.

We mimic Definition 6.2.1, the natural number n is still supposed to be fixed.

Definition 6.2.12. Let C be a connectology over E. A **connected n-block** is a subset of E^n of the form $C_1 \times \cdots \times C_n$ with $C_i \in C$ and $C_i \neq \emptyset$ for all $i \in \{1, \dots, n\}$. The maximal elements of the collection of blocks contained in $X \subseteq E^n$ are called the **maximal connected n-blocks** of X.

Lemma 6.2.13.
Any connected block of $X \subseteq E^n$ is contained in a maximal connected block of X.

Proof. Let C be a connected block of X. By Zorn's Lemma there exists some maximal $\subseteq$-chain of connected blocks of X in which C appears. The union of this chain is calculated component by component as a monotonic union of connected subsets of E, so it is a maximal connected block of X containing C. □

Definition 6.2.14. A **connected block covering** of $X \subseteq E^n$ is a collection $\mathcal{F}$ of connected blocks whose union is X. Note that it is not required to be finite (compare with Definition 6.2.5). A connected block covering is thus a family of connected blocks. We denote by $\mathrm{ConnCov}_n(C)$ (resp. $\mathrm{FinConnCov}_n(C)$) the collection of all (resp. finite) connected n-block coverings. The collection of all the singletons contained in X is a connected block covering. Of course it is the worst one in the sense that even connected blocks contain infinitely many points in general. However, by Lemma 6.2.13 one soundly defines the **maximal connected block covering** of X as the collection of its maximal connected blocks. It is the greatest element of the collection of connected block coverings of X with respect to the covering preorder.

Remark 6.2.15. The equivalence stated in Remark 6.2.6 for maximal block coverings still holds for maximal connected block coverings.

Proposition 6.2.16. The set $\mathrm{ConnCov}_n(C)$ is equipped with the covering preorder, and the powerset of E^n is ordered by inclusion. Then we have a Galois connection

$$\begin{tikzcd}\mathrm{ConnCov}_n(C) \arrow[r, shift left, "\gamma_n"] & \mathrm{Pow}(E^n) \arrow[l, shift left, "\alpha_n"]\end{tikzcd}$$

defining $\gamma_n(\mathcal{F})$ as $\bigcup \mathcal{F}$ and $\alpha_n(X)$ as the maximal connected block covering of X. In particular $\gamma_n \circ \alpha_n = \mathrm{id}$ and $\alpha_n(\emptyset)$ is the empty family. The Galois connection becomes an isomorphism of Boolean algebras if one substitutes $\mathrm{ConnCov}_n(C)$ with the image of α_n (*i.e.* the collection of maximal connected block coverings).

Proof. As before the monotonicity of γ_n exactly means that the covering preorder is finer that the gathering one (*cf.* Definition 6.2.5). Any point of X can be seen as a connected block since C contains all the singletons. The monotonicity of α_n and the relation $\gamma_n \circ \alpha_n = \mathrm{id}$ are thus immediate consequences of Lemma 6.2.13. By Remark 6.2.15 the restriction of the relation $\preccurlyeq_c$ to the image of α_n induces a partial order that is isomorphic to that of the Boolean structure of $\mathcal{B}$, namely the inclusion relation inherited from $\mathrm{Pow}(E)$. □

Example 6.2.17. Let E be the real line and C be the collection of its intervals. The compact unit disk contains infinitely many maximal connected blocks (*i.e.* compact rectangles).

Remark 6.2.18. As in topology, the maximal connected blocks of a finite Cartesian product are easy to determine. Indeed, given $X_i \subseteq E$ for $i \in \{1, \dots, n\}$ the following equality

$$\alpha_n(X_1 \times \cdots \times X_n) \quad = \quad \alpha_1(X_1) \times \cdots \times \alpha_1(X_n)$$

holds because $A_1 \times \cdots \times A_n \subseteq B_1 \times \cdots \times B_n$ iff $A_i \subseteq B_i$ for all $i \in \{1, \dots, n\}$.

Both Proposition 6.2.7 and Proposition 6.2.16 provide normal forms to describe certain elements of E^n. Assuming that Boolean operations in $\mathcal{B}$ are computationally tractable, Proposition 6.2.7 provides a Boolean subalgebra of $\mathrm{Pow}(E^n)$ that is also computationally tractable (up to the problem of complexity). However, if $\mathcal{B}$ is too restrictive, then so is the associated Boolean subalgebra. On the other hand Proposition 6.2.16 provides a normal form for any element of $\mathrm{Pow}(E^n)$ but implementing computation might not be feasible if one has to deal with infinite families of connected blocks. Assuming an extra hypothesis on connectologies, one can take advantage of both approaches by studying subsets of E^n that can be written as a finite union of connected blocks.

Definition 6.2.19. A connectology C is said to be **regional** when the collection of finite unions of elements of C forms a Boolean subalgebra of $\mathrm{Pow}(E)$. It amounts to suppose that the collection of finite union of connected subsets of E is stable under complement.

Example 6.2.20. The collection of connected subsets of $\mathbb{R}$ (*i.e.* its subintervals) is a regional connectology while the collection of connected subsets of $\mathbb{R}^2$ is not since one can easily find a connected subset of the plane whose complement has infinitely many connected components (*e.g.* $\mathbb{R}^2 \setminus \mathbb{Z}^2$).

The notions introduced in Definition 6.2.1, 6.2.5, and 6.2.12 heavily depend on the subsets of E that are allowed to form blocks, viz $\mathcal{B}$ or C, hence we sometimes add a prefix to stress that fact. This will be useful in the context of Theorem 6.2.21.

Theorem 6.2.21. If $\mathcal{B}$ is the Boolean subalgebra of $\mathrm{Pow}(E)$ generated by a regional connectology C then the Galois connection of Proposition 6.2.16 restricts as follows:

$$\begin{tikzcd} \mathrm{FinConnCov}_n(C) \arrow[r, shift left, "\gamma_n"] & \mathcal{R}_n \arrow[l, shift left, "\alpha_n"] \end{tikzcd} .$$

Moreover, it becomes an isomorphism of Boolean algebras if $\mathrm{FinConnCov}_n(C)$ is replaced by the image of α_n (*i.e.* the collection of maximal connected block coverings).

Proof. A maximal connected block C of $X \in \mathcal{R}_n$ is contained in a maximal $\mathcal{B}$-block B of X. Hence C is also a maximal connected block of B. We known from Remark 6.2.18 that B has finitely many maximal connected blocks. Since X has finitely many maximal $\mathcal{B}$-blocks, we conclude that it also has finitely many maximal connected blocks. □

Remark 6.2.22. Formally speaking $\mathcal{R}_0$ contains exactly two elements which are the empty set and the singleton containing the empty word, it is therefore isomorphic to the Boolean algebra {true, false}.

Remark 6.2.23. Assuming that C is a regional connectology, the Boolean algebra $\mathcal{R}_1$ is precisely the Boolean subalgebra of $\mathrm{Pow}(E)$ generated by C.

Example 6.2.24. One should keep in mind the situation where E is the real line and C is the collection of its intervals: the elements of $\mathcal{R}_n$ are then called the **cubical regions**.

Example 6.2.25. Example 6.2.24 is actually a special instance of the case where E is the (underlying set of) the geometric realization of some essentially finite graph G – see Definition 6.1.43, and C is the collection of its connected subsets.

Notation α_n refers to the Galois connection introduced in Proposition 6.2.16 and C to a connectology. The following lemmas provides the tools to compute the normal form associated to a connected block covering, in other words the mapping $\alpha_n \circ \gamma_n$.

Lemma 6.2.26. Let C and C' be n-dimensional connected blocks. Then

$$\alpha_n(C \cap C') = \alpha_1(C_1 \cap C'_1) \times \cdots \times \alpha_1(C_n \cap C'_n)$$

If C is regional, then $C \cap C'$ has finitely many maximal connected blocks.

Proof. By the exchange law and Remark 6.2.18. □

Lemma 6.2.27. Let C be a connected block, a connected block M of $E^n \setminus C$ is maximal iff there exists $i \in \{1, \ldots, n\}$ such that M_i is a connected component of $E \setminus C_i$ and for $j \neq i$, M_j is a connected component of E. If C is regional, then $E^n \setminus C$ has finitely many maximal connected blocks.

Proof. The complement of a connected block C can be written as the finite union of the sets

$$E \times \cdots \times E \times \underbrace{E \setminus C_i}_{i^{th}\text{ position}} \times E \times \cdots \times E$$

Then note that the maximal connected blocks of $E^n \setminus C$ are precisely the maximal connected blocks of the products given above. We conclude by Remark 6.2.18. □

Lemma 6.2.28. Given isothetic regions X and Y we have the following inclusion:

$$\alpha_n(X \cap Y) \subseteq \bigcup \left\{ \alpha_n(C \cap C') \,\middle|\, C \in \alpha_n(X);\ C' \in \alpha_n(Y) \right\} .$$

If C is regional, then $X \cap X'$ has finitely many maximal connected blocks.

Proof. Let $C'' \in \alpha_n(X \cap X')$, there exist $C \in \alpha_n(X)$ and $C' \in \alpha_n(X')$ such that $C'' \subseteq C \cap C'$, so $C'' \in \alpha_n(C \cap C')$. The finiteness comes from Theorem 6.2.21. □

The subtle distinction between a regional connectology and its associated Boolean algebra actually matters. To explain this, let us consider the case where the regional connectology C is the collection of intervals of the real line. Then

$$X = [0, 1] \times [0, 1] \cup [0, 1] \times [2, 3] \cup [2, 3] \times [0, 1]$$

has 3 maximal connected blocks and 2 maximal blocks – see Figure 6.7. In practice, the Boolean operations on isothetic regions are based on their analog on blocks or connected blocks, according to the way one has chosen to represent the isothetic regions. From a combinatorial point of view, the block based approach leads to much smaller representation so it should be preferred. As a dramatic example the block $([0, 1] \cup [2, 3])^n$ is made of 2^n maximal connected blocks – see Figure 6.7.

Furthermore the basic operations on blocks are built on the Boolean operations on finite unions of intervals. The latter can be implemented so that their complexity is linear in the number of maximal intervals. A more naive approach leads to quadratic algorithms. Of course this optimization cannot be exploited if one represents the isothetic regions by means of connected blocks.

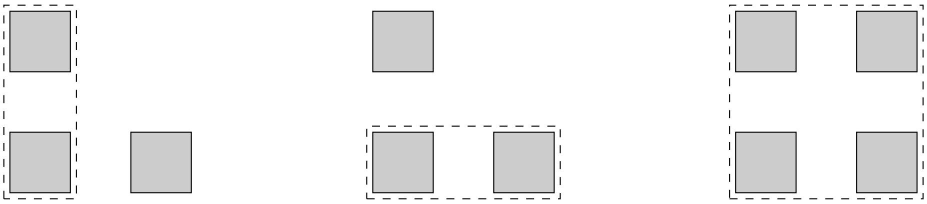


Figure 6.7: A fine connectology and its associated Boolean algebra

6.3 Product of Isothetic Regions

We describe the combinatorial framework for the product of isothetic regions. Let $\mathbb{A}$ be a set called the **alphabet**. The free monoid generated by $\mathbb{A}$ is denoted by $\mathbb{A}^*$. Its elements are the **words** (*i.e.* the finite sequences) over $\mathbb{A}$, and its composition law is the **concatenation**: given words w and w' of length n and n', the word $w * w'$ of length $n + n'$ is defined by

$$(w * w')_k = \begin{cases} w_k & \text{if} \quad 1 \leqslant k \leqslant n \\ w'_{k-n} & \text{if} \quad n+1 \leqslant k \leqslant n+n' \end{cases}$$

The empty word ε is the neutral element of $\mathbb{A}^*$. The concatenation extends to **languages** (*i.e.* sets of words): given $D, D' \subseteq \mathbb{A}^*$ we define

$$D * D' := \{w * w' \mid w \in D; w' \in D'\}$$

The set of languages (*i.e.* the powerset of $\mathbb{A}^*$) is thus endowed with a structure of (non-commutative) monoid $\mathcal{D}(\mathbb{A})$ whose neutral element is $\{\varepsilon\}$ (*i.e.* the singleton containing the empty word). Note that the empty language $\emptyset$ is the absorbing element of $\mathcal{D}(\mathbb{A})$, that is for all $D \subseteq \mathbb{A}^*$ we have

$$\emptyset * D = D * \emptyset = \emptyset$$

Unless $\mathbb{A}$ is a singleton (and then $\mathbb{A}^* \cong (\mathbb{N}, +, 0)$), the monoid $\mathbb{A}^*$ is noncommutative in a very strong sense: the only word that commutes with all the others is the empty one. The length of a word w is also referred to as $\ell(w)$.

Definition 6.3.1. A language is said to be **homogeneous** when all the words it contains share the same length. The notion of length thus extends to homogeneous languages with the convention that $\ell(\emptyset) = -\infty$, and we have the following relation for all homogeneous languages D and D'

$$\ell(D * D') = \ell(D) + \ell(D')$$

The homogeneous languages form a submonoid $\mathcal{D}_h(\mathbb{A}) \subseteq \mathcal{D}(\mathbb{A})$ and we sometimes write $\mathcal{D}_n(\mathbb{A})$ (resp. $\mathcal{D}_{nf}(\mathbb{A})$) for the collection of homogeneous languages (resp. finite languages) of length $n \in \mathbb{N}$. Hence we have the disjoint unions

$$\mathcal{D}_h(\mathbb{A}) = \bigsqcup_{n \in \mathbb{N}} \mathcal{D}_n(\mathbb{A}) \quad \text{and} \quad \mathcal{D}_{hf}(\mathbb{A}) = \bigsqcup_{n \in \mathbb{N}} \mathcal{D}_{nf}(\mathbb{A})$$

* * *

We are now ready to describe a special class of subsets of the disjoint union

$$\bigsqcup_{n\in\mathbb{N}} \rceil G \lfloor^n$$

that can be handled algorithmically – see Section 6.4, and provides a framework in which parallel composition can be modelled – see Proposition 7.1.5. From now on and until the end of the section, the graph G is supposed to be essentially finite (*cf.* Example 6.2.25). With notation of Section 6.2, the idea is to see each block as a word over a regional connectology (*i.e.* $\mathbb{A} = \mathcal{C}$) or its associated Boolean algebra (*i.e.* $\mathbb{A} = \mathcal{B}$), and thus interpreting a covering as a language over the corresponding alphabet. The only subtlety one has to deal with is the empty set. Indeed, if some letter of a word is the empty set, then the whole word should be understood as the empty set and could therefore be dropped from the language. Nevertheless, we have to keep in mind that the underlying sets of $\mathcal{R}_n$ and $\mathcal{R}_m$ are disjoint for $n \neq m$. In particular, since we are handling homogeneous languages (*i.e.* coverings) of length n instead of subsets of E^n, the empty set of $\mathcal{R}_n$ and the emptyset of $\mathcal{R}_m$ should not be represented by the same mathematical object. A natural trick is to forbid the empty language and identify any word in which the empty set appears with the word all the letters of which are the empty set. Another theoretical reason for this glitch will be explained in Section 9.5 and more specifically in Example 9.5.18 and Remark 9.5.21. In this context, the least element of $\mathbb{A}$ is the empty subset of $\rceil G \lfloor$. Since we think of $\mathbb{A}$ as a collection of subsets of $\rceil G \lfloor$ and the words over it as blocks, it is natural to order them by inclusion. In particular given two such words w and w' of length n, we have $w \subseteq w'$ when $\emptyset$ occurs in w or when $w(i)$ is included in $w'(i)$ for all $i \in \{0, \ldots, n-1\}$. The rest of the section can be seen as an introduction to Section 9.3.

Definition 6.3.2. Because any element of $\rceil G \lfloor^n$ can be identified with a word of length n over $\rceil G \lfloor$, the monoid $\mathcal{D}_h(\rceil G \lfloor)$ is identified with the following disjoint union of powersets so it is ordered by inclusion at each level

$$\mathcal{P} \quad = \quad \bigsqcup_{n\in\mathbb{N}} \mathrm{Pow}(\rceil G \lfloor^n).$$

The **monoid of regions** is the following disjoint union of Boolean algebras equipped with concatenation and ordered by inclusion at each level

$$\mathcal{R} \quad = \quad \bigsqcup_{n\in\mathbb{N}} \mathcal{R}_n \ .$$

The morphisms α_n and the morphisms γ_n from Theorem 6.2.21 can thus be gathered to form mappings α and γ so that we have the following result.

Corollary 6.3.3. The mappings α and γ induce morphisms of (pre)ordered monoids between $\mathcal{D}_{hf}(\mathcal{C})$ and $\mathcal{R}$. They become isomorphisms of ordered monoids if one restricts $\mathcal{D}_{hf}(\mathcal{C})$ to the image of α (*i.e.* the maximal connected block coverings).

Proof. The only nontrivial point to check is that α preserves the product, which derives from the fact that for all blocks w, w', and w'' of dimension n, n', and $n + n'$, one has $w * w' \subseteq w''$ iff $w \subseteq \mathrm{proj}(w'')$ and $w' \subseteq \mathrm{proj}'(w'')$, where proj and proj′ are the projections on coordinates 1 to n and $n + 1$ to $n + n'$. □

Remark 6.3.4. As a by-product of the proof of Corollary 6.3.3, the maximal subblocks of the Cartesian product of isothetic region $A \times B$ have the form $w \times w'$ where w (resp. w') is a maximal subblock of A (resp. B).

Remark 6.3.5. By Proposition 6.2.7 (resp. 6.2.16), Corollary 6.3.3 is still valid for $\mathcal{D}_{hf}(\mathcal{B})$ and $\mathcal{R}$ (resp. $\mathcal{D}_h(C)$ and $\mathcal{P}$) where $\mathcal{B}$ is the Boolean algebra associated with the regional connectology C.

In Chapter 9, the monoids introduced in this section will be turned into free commutative monoids so the prime decomposition corresponds to parallelization of code.

6.4 Directed Topological Regions

Let $\rceil G \lfloor$ be the directed geometric realization of some essentially finite graph G – see Definition 6.1.43. The realization can be indifferently taken in many categories (*cf.* Proposition 6.1.7) yet we chose the one of local pospaces which is more restrictive. From Theorem 6.1.42 we know that the collection of finite unions of connected subsets of $\rceil G \lfloor$ is a Boolean algebra, in other words the connectology associated with the topology of $\rceil G \lfloor$ is fine. In particular Proposition 6.2.7 and Theorem 6.2.21 apply and the elements of the Boolean algebras $\mathcal{R}_n$ are also called the G-**regions** – see Example 6.2.25. Any G-region inherits the direction and the topology from $\rceil G \lfloor$ so it becomes a **directed topological isothetic region**. As we shall see in the remainder of this section, these additional structures can be expressed in terms of G-regions, which makes the notion tractable. Note that when $|G|$ is homeomorphic with a nondegenerate interval of $\mathbb{R}$ or equivalently when G is a connected subgraph of $\mathbb{Z}$ (*cf.* Definition 6.1.27) the G-regions are cubical – see Example 6.2.24.

In the rest of this chapter, the graph G is supposed to be essentially finite (*cf.* Definition 6.1.43).

Interior and Closure Operators

The Boolean algebra $\mathcal{R}_n$ inherits, in a sense to be made precise, from the topological structure of $\rceil G \lfloor$. Let us denote the **interior** of X (*i.e.* the greatest open subset of $\rceil G \lfloor$ contained in X) by int(X) and the **closure** of X (*i.e.* the least closed subset of $\rceil G \lfloor$ containing X) by clo(X). The closure of X is also denoted by $\overline{X}$. Also denote the **boundary** of X (*i.e.* the set difference clo(X) \ int(X)) as bnd(X). It is known, from Kuratowski's axioms, that a topology is entirely characterized by the induced closure (resp. interior) operator. Note that $\mathcal{R}_1$ is actually $\mathcal{R}_{|G|}$ – see Remark 6.2.23.

Lemma 6.4.1. The Boolean algebra $\mathcal{R}_1$ is stable under closure, boundary, and interior operators.

Proof. First note that the Boolean algebra $\mathcal{R}_1$ is stable under the closure operator since the closure of a connected set is connected. The boundary of some $A \in \mathcal{R}_1$ can also be written as

$$\mathrm{bnd}(A) = \mathrm{clo}(A) \cap \mathrm{clo}(\rceil G \lfloor \setminus A)$$

so it also belongs to $\mathcal{R}_1$. As a consequence the interior of A also belongs to $\mathcal{R}_1$ because it can be written as

$$\mathrm{int}(A) = \mathrm{clo}(A) \setminus \mathrm{bnd}(A)$$

□

Proposition 6.4.2. The Boolean algebra $\mathcal{R}_n$ is stable under closure, boundary, and interior operators.

Proof. A basis of the topology of $\rceil G\lfloor^n$ is given by the open connected blocks (*i.e.* those ones whose k^{th} projection is open in $|G|$ for all k). Then remark that for all blocks w we have

$$\mathrm{int}(w) = \mathrm{int}(w_1) \times \cdots \times \mathrm{int}(w_n)$$

and

$$\mathrm{clo}(w) = \mathrm{clo}(w_1) \times \cdots \times \mathrm{clo}(w_n)$$

the latter is a connected block while the former is a disjoint union of (open) connected blocks by Lemma 6.4.1. In particular a point $x \in \rceil G\lfloor$ belongs to $\mathrm{int}(X)$ iff there is some connected open block w such that $x \in w \subseteq X$. Therefore

$$\mathrm{int}(X) = \bigcup_{w \in \alpha_n(X)} \mathrm{int}(w)$$

the union being taken over the collection $\alpha_n(X)$ of the maximal blocks of X – see Proposition 6.2.7. The stability under closure operator immediately derives from the basic fact (of general topology) that for any finite collection C of subsets of a topological space the following equality holds

$$\mathrm{clo}(\bigcup C) = \bigcup_{C \in C} \mathrm{clo}(C)\ .$$

□

Forward and Backward Operators

Until now, we have been concerned about the Boolean and the topological structures of isothetic regions, which are both unambiguously defined. Defining "the" direction of an isothetic region is a bit more subtle because it admits a structure of a local pospace, of a stream, and of a d-space. This structure is indeed inherited from that of $\rceil G\lfloor^n$. Therefore it may depend on the category in which G has been realized. However, by Proposition 6.1.7, and Theorem 4.5.21 and its corollaries, the collection of paths on $|G|$ that induce a directed path on $\rceil G\lfloor$ (*cf.* Definition 4.2.12) does not depend on the category in which G has been realized. Formally, the collection

$$\{U(\gamma) \mid \gamma \in \mathcal{C}([0,r], \rceil G\lfloor_{\mathcal{C}})\}$$

does not depends on the category $\mathcal{C} \in \{\mathbf{LpoTop}, \mathbf{dTop}, \mathbf{Strm}, \mathbf{dTop}_{\mathrm{f}}, \mathbf{Strm}_{\mathrm{d}}\}$ where U is the forgetful functor to **Haus**. The preceding collection is actually the collection of paths on the geometric realization of G such that for all arrows α of G, and for all connected components C of $\gamma^{-1}(\{\alpha\}\times]0,1[)$, the restriction of γ to C is order-preserving.

Proposition 6.4.3. A directed path on a region X is the sense of Definition 4.2.12 is a path γ on UX in the sense of Definition 2.1.7 such that for all $i \in \{1,\ldots,n\}$, the i^{th} projection of γ is a directed path on $\rceil G\lfloor$ in the above sense.

Since the operators we are about to define on the class of isothetic regions only depend on the collection of directed paths on them, these operators are unsensitive for the choice of the underlying category. We arbitrarily decide to work with d-spaces so we write $\rceil G\lfloor$ without subscript and "subobject" without further precision. Then we prove that the Boolean algebra $\mathcal{R}_n$ also inherits from the directed structure of $\rceil G\lfloor^n$.

Definition 6.4.4. For all subsets A and B of a d-space X, the elements of $\mathrm{frw}(A, B)$ are the endpoints of the directed paths of $A \cup B$ starting in A.

$$\mathrm{frw}(A, B) = \{\partial^+\delta \mid \delta \text{ directed path of } X;\ \partial^-\delta \in A;\ \mathrm{img}(\delta) \subseteq A \cup B\}$$

The mapping which send (A, B) to $\mathrm{frw}(A, B)$ is called the **forward** operator over X. The **backward** operator over X (*i.e.* the mapping $(A, B) \mapsto \mathrm{bck}(A, B)$) is defined dually.

$$\mathrm{bck}(A, B) = \{\partial^-\delta \mid \delta \text{ directed path of } X;\ \partial^+\delta \in A;\ \mathrm{img}(\delta) \subseteq A \cup B\}$$

Remark 6.4.5. Both operators are $\subseteq$-increasing in both variables. Also remark that for all subsets A and B, we have $A \subseteq \mathrm{frw}(A, B) \subseteq A \cup B$, and that if A and B are disconnected (*i.e.* neither A nor B meets the closure of the other) then $\mathrm{frw}(A, B) = A$. Note however that the converse is false (*e.g.* $A = [1, 2]$ and $B = [0, 1]$). The same obviously holds for the backward operator. Furthermore if A is (path) connected then so are $\mathrm{frw}(A, B)$ and $\mathrm{bck}(A, B)$. Indeed as any d-space is stable under subpaths, Definition 6.4.4 can be rephrased as follows.

$$\mathrm{frw}(A, B) = \bigcup \left\{\mathrm{img}(\delta) \,\middle|\, \delta \text{ directed path of } X;\ \partial^-\delta \in A;\ \mathrm{img}(\delta) \subseteq A \cup B\right\}$$

Lemma 6.4.6. For all $A_k \subseteq B_k$ with $k \in \{1, \dots, n\}$, we have

$$\mathrm{frw}(A_1 \times \cdots \times A_n, B_1 \times \cdots \times B_n) = \mathrm{frw}(A_1, B_1) \times \cdots \times \mathrm{frw}(A_n, B_n)$$

and

$$\mathrm{bck}(A_1 \times \cdots \times A_n, B_1 \times \cdots \times B_n) = \mathrm{bck}(A_1, B_1) \times \cdots \times \mathrm{bck}(A_n, B_n)$$

Proof. If δ is a dipath on $B_1\times\cdots\times B_n$ with $\partial^-\delta \in A_1\times\cdots\times A_n$ then for all $k \in \{1, \dots, n\}$, $\mathrm{pr}_k \circ \delta$ is a dipath on B_k whose source belong to A_k. Conversely, given some n-tuple of dipaths $(\delta_1, \dots, \delta_n)$ with δ_k dipath on B_k with its source in A_k, the dipath defined by $t \mapsto (\delta_1(t), \dots, \delta_n(t))$ has its source in $A_1 \times \cdots \times A_n$ and its image contained in $B_1 \times \cdots \times B_n$. □

The inclusion assumption cannot be dropped from the statement of Lemma 6.4.6. Indeed, taking the disconnected sets $A = [0, 1[\times[0, 1]$ and $B = [1, 2]\times]1, 2]$ we have $\mathrm{frw}(A, B) = A$ though

$$\mathrm{frw}([0, 1[, [1, 2]) = \mathrm{frw}([0, 1],]1, 2]) = [0, 2]\ .$$

Definition 6.4.7. For a given d-space X we define for all $A \subseteq X$:

- the **future closure** (resp. **past closure**) of A, denoted by $\overline{A}^{\mathrm{f}}$ (resp. $\overline{A}^{\mathrm{p}}$), as $\mathrm{frw}(A, \overline{A})$ (resp. $\mathrm{bck}(A, \overline{A})$),
- the **future cone** (resp. **past cone**) of A, denoted by $\mathrm{cone}^{\mathrm{f}}(A)$ (resp. $\mathrm{cone}^{\mathrm{p}}(A)$) as $\mathrm{frw}(A, X)$ (resp. $\mathrm{bck}(A, X)$),
- the subset A is said to be **future stable** when $\mathrm{cone}^{\mathrm{f}}A = A$. The **past stable** subsets are defined dually.

Remark 6.4.8. With the notation of Definition 6.4.7, $\mathrm{cone}^{\mathrm{f}}A$ (resp. $\mathrm{cone}^{\mathrm{p}}A$) is future (resp. past) stable, and any d-subspace is future stable iff its complement is past stable. The collection of future (resp. past) stable d-subspaces of X forms a sub-complete lattice of the powerset of X.

Remark 6.4.9. As a consequence of Lemma 6.4.6, for all A_k with $k \in \{1, \dots, n\}$ we have

$$\overline{A_1 \times \cdots \times A_n}^{\mathrm{f}} = \overline{A_1}^{\mathrm{f}} \times \cdots \times \overline{A_n}^{\mathrm{f}}$$

and

$$\mathrm{cone}^{\mathrm{f}}(A_1 \times \cdots \times A_n) = \mathrm{cone}^{\mathrm{f}}(A_1) \times \cdots \times \mathrm{cone}^{\mathrm{f}}(A_n)$$

Dually, the same holds for past closure and past cone.

From the operators of Definition 6.4.7 one can define several meaningful new ones.

Definition 6.4.10. Given a subset A of a d-space X, the subset $\mathrm{cone}^{\mathrm{p}} A \cup \mathrm{cone}^{\mathrm{f}} A$ is called the **cone** of A, it is denoted by $\mathrm{cone}_X A$. The reference to the underlying d-space X is often omitted.

Remark 6.4.11. Unlike the future cone and the past cone operators, the cone operator is not idempotent. In fact, the $\subseteq$-increasing sequence $(\mathrm{cone}^n A)_{n \in \mathbb{N}}$ may not be stationary. The union of its terms

$$\mathrm{cone}^{\infty} A \quad = \quad \bigcup_{n \in \mathbb{N}}^{\uparrow} \mathrm{cone}^n A$$

is indeed the set of points that can be reached from some point of A by a zigzag of directed and antidirected paths. In particular, the fundamental category of a d-space is connected iff this d-space is equal to $\mathrm{cone}^{\infty}(\{p\})$ for some point p. Of course, if such a point p exists, then any other point also satisfies the property. For example, if we let X be the infinite ascending staircase on Figure 6.8 (seen as a noncompact subspace of $\mathbb{R}^2$ which is actually dihomeomorphic to $\mathbb{R}_+$), then $\mathrm{cone}_X\{p\} = X$ for every point p. The latter assertion assumes that any increasing paths on $\mathbb{R}^2$ is directed. The preceding d-space should be compared to the infinite descending staircase X (*cf.* Figure 6.8), which is dihomeomorphic to the directed geometric realization of the half infinite zigzag shown below.

$$\begin{tikzcd}
p \bullet \arrow[r] & \bullet & \bullet \arrow[l] \arrow[r] & \bullet & \bullet \arrow[l] \arrow[r] & \bullet & \bullet \arrow[l] \arrow[r] & \bullet & \bullet \arrow[l] \arrow[r] & \bullet & \cdots \arrow[l]
\end{tikzcd}$$

In that case we have $\mathrm{cone}_X^n(\{p\}) \subsetneq \mathrm{cone}_X^{n+1}(\{p\})$ for all $n \in \mathbb{N}$. Yet, the fundamental categories of both infinite staircases are connected.

Example 6.4.12. Given a graph G satisfying the properties of Theorem 6.1.42 (resp. Theorem 6.1.20), there is a finite set of point $F \subseteq \rceil G \lfloor$ and some $n \in \mathbb{N}$ such that $\mathrm{cone}^n_{\rceil G \lfloor} F = \rceil G \lfloor$ (resp. $\mathrm{cone}^{\infty}_{\rceil G \lfloor} F = \rceil G \lfloor$). The converse is false as one can check by considering a star with infinitely many branches.

Conjecture 6.4.13. For all subsets A of an isothetic region X, the sequence $(\mathrm{cone}^n_X A)_{n \in \mathbb{N}}$ is stationary and its limit is the union of all the connected components of X that meet A.

Conjecture 6.4.13 is illustrated by Figure 6.9. We consider the unit square from which some horizontal and vertical segments have been removed. The latter are the "walls" of a "labyrinth".

Remark 6.4.14. Let A be the set of points $\{(n, -n) \mid n \in \mathbb{Z}\} \quad \subseteq \quad \mathbb{R}^2$. Then

$$\mathbb{R}^2 \setminus \mathrm{cone} A \quad = \quad \bigcup_{n \in \mathbb{Z}} [n-1, n] \times [n, n+1] \setminus A$$

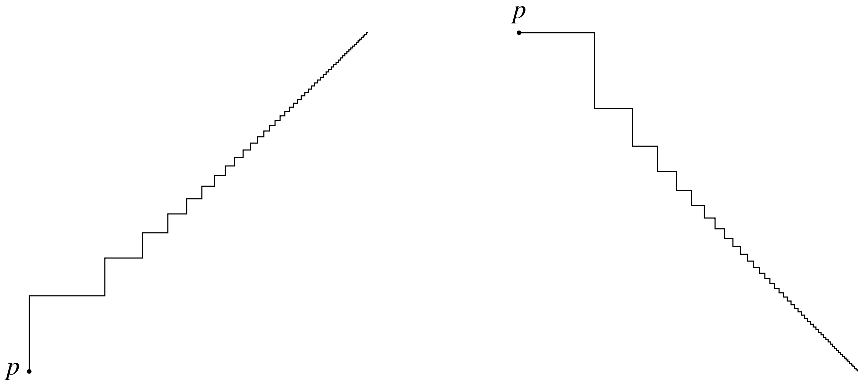


Figure 6.8: Infinite staircases

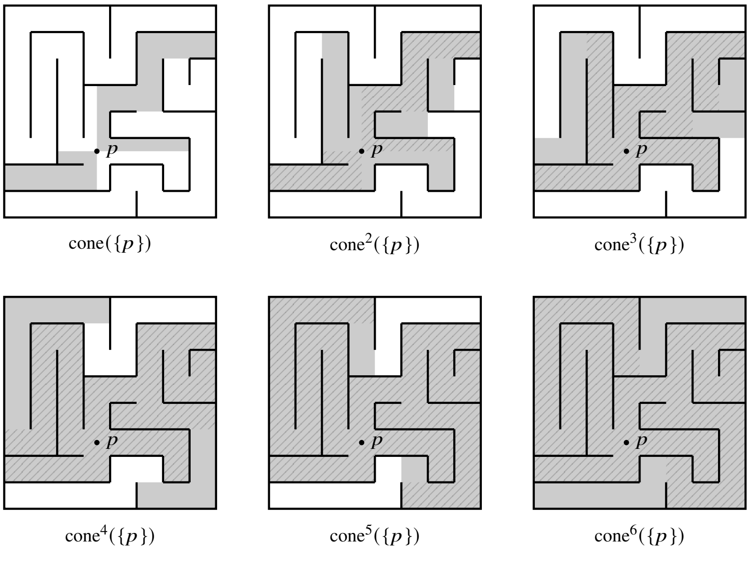


Figure 6.9: Labyrinth

Definition 6.4.15. A d-space X is said to be **zigzag connected** when it contains a point p such that $\mathrm{cone}^{\infty}_{X}(\{p\}) = X$. It is said to be **finitely zigzag connected** when it contains a point p such that $\mathrm{cone}^{n}_{X}(\{p\}) = X$ for some $n \in \mathbb{N}$.

Remark 6.4.16. Given a point p of a zigzag connected d-space X, the least $n \in \mathbb{N} \cup \{\infty\}$ such that $\mathrm{cone}^{n}_{X}(\{p\}) = \mathrm{cone}^{n+1}_{X}(\{p\})$ may depend on p, we denote it by n_p. Then we define the **zigzag diameter** of a zigzag connected d-space X as the least element of the set below.

$$\left\{ n_p \in \mathbb{N} \cup \{\infty\} \,\middle|\, p \in X \right\}$$

Example 6.4.17. If $X = \rceil G \lfloor$ for some finite zigzag G and p is one of the extremities of X, then the least $n \in \mathbb{N}$ such that $\mathrm{cone}^{n}_{X} A = \mathrm{cone}^{n+1}_{X} A$ is the number of arrows of G, which corresponds to the "length" of the zigzag.

The operators we are about to introduce are involved in deadlock detection.

Definition 6.4.18. Given a d-space X we define for all $A \subseteq X$, the **future escape** of A, denoted by $\mathrm{escape}^{\mathrm{f}} A$, as the set of points of X whose future cones avoid A. the **past attractor** of A, denoted by $\mathrm{att}^{\mathrm{p}} A$, is the set of points of $x \in X$ such that any dipath starting at x can be extended to a dipath arriving in A. The **past escape** and the **future attractor** of A, denoted by $\mathrm{escape}^{\mathrm{p}} A$ and $\mathrm{att}^{\mathrm{f}} A$, are defined dually.

Proposition 6.4.19. The future escape and the past attractor are expressed in terms of past cone and complement. Moreover, both are future stable.

$$\mathrm{escape}^{\mathrm{f}} A \;=\; (\mathrm{cone}^{\mathrm{p}} A)^{c} \qquad\qquad \mathrm{att}^{\mathrm{p}} A \;=\; \mathrm{escape}^{\mathrm{f}}(\mathrm{escape}^{\mathrm{f}} A)$$

Lemma 6.4.20. For all subsets $A \subseteq B \subseteq C$ of a d-space, we have

$$\mathrm{frw}(A, C) = \mathrm{frw}(\mathrm{frw}(A, B), C) \quad \text{and} \quad \mathrm{bck}(A, C) = \mathrm{bck}(\mathrm{bck}(A, B), C)$$

Proof. The left member is contained in the right one by Remark 6.4.5 and because $A \subseteq \mathrm{frw}(A, B)$ (no assumption on A, B, nor C is required here). Conversely, let γ be a dipath from $\mathrm{frw}(A, B)$ to C whose image is contained in $\mathrm{frw}(A, B) \cup C$. We have $\mathrm{frw}(A, B) \subseteq B \subseteq C$, hence the image of γ is contained in C. By definition of $\mathrm{frw}(A, B)$, there also exists a dipath δ from A to $\partial^{-}\gamma$ whose image is contained in B, hence in C. The concatenation $\gamma \cdot \delta$ therefore starts in A and has its image contained in C. □

Once again the inclusion assumption cannot be dropped. Taking $A = [0, 1]$, $B = [1, 2]$, and $C = [2, 3]$ provides an obvious counter-example. The following result is the key ingredient for effective computations in the case of isothetic regions.

Proposition 6.4.21. Let A and B be subsets of some d-space X, and suppose that for all dipaths δ of X, $\delta^{-1}(A)$ has finitely many connected components. Then

$$\mathrm{frw}(A, B) = A \cup \mathrm{frw}(\overline{A}^{\mathrm{f}} \cap B, B) \cup \mathrm{frw}(A \cap \overline{B}^{\mathrm{p}}, B)$$

and

$$\mathrm{bck}(A, B) = A \cup \mathrm{bck}(\overline{A}^{\mathrm{p}} \cap B, B) \cup \mathrm{bck}(A \cap \overline{B}^{\mathrm{f}}, B)$$

Proof. Let δ be a dipath on $A \cup B$ starting in A and such that $\partial^{+}\delta \notin A$ (hence $\partial^{+}\delta \in B$). Let C be the connected component of $\delta^{-1}(B)$ that contains 1 and denote by t_0 its greatest lower bound. Then any neighborhood of t_0 contains some $t < t_0$ such that $\delta(t) \notin B$, and therefore $\delta(t) \in A$. By hypothesis on A the last connected component of $\delta^{\text{-}1}(A)$ makes sense. Its least upper bound is t_0. Then $\delta(t_0) \in \overline{A}^{\mathrm{f}}$.

If $\delta(t_0) \in B$ then $\delta|_{[t_0,1]}$ is a dipath whose image is contained in B. It follows that $\partial^{+}\delta \in \mathrm{frw}(\overline{A}^{\mathrm{f}} \cap B, B)$.

If $\delta(t_0) \notin B$ the image of $\delta|_{[t_0,1]}$ is then included in $\overline{B}^{\mathrm{p}}$ and $\delta(t_0) \in A$ (since $\mathrm{img}(\delta) \subseteq A \cup B$). It follows that the image of $\delta|_{[t_0,1]}$ in included in $A \cap \overline{B}^{\mathrm{p}} \cup B$ and therefore $\partial^{+}\delta \in \mathrm{frw}(A \cap \overline{B}^{\mathrm{p}}, B)$.

Conversely, suppose that there exists a dipath δ starting in $\overline{A}^{\mathrm{f}} \cap B$ and whose image is contained in B. Then consider a dipath γ starting in A and such that $\partial^{+}\gamma = \partial^{-}\delta$. The image of the concatenation $\delta \cdot \gamma$ is then contained in $A \cup B$. Therefore $\partial^{+}\delta \in \mathrm{frw}(A, B)$.

Now if δ is a dipath starting in $A \cap \overline{B}^{\mathrm{p}}$ and whose image is contained in $(A \cap \overline{B}^{\mathrm{p}}) \cup B$. Suppose that $\partial^{+}\delta \notin A$, then $\partial^{+}\delta \in B$ and thus let C be the connected component of $\delta^{\text{-}1}(B)$ that contains 1. Let t_1 be the greatest lower bound of C. Suppose that $\delta(t_1) \notin A$, therefore $\delta(t_1) \in B$ and any neighborhood of t_1 contains some $t < t_1$ such that $\delta(t) \notin B$ (and therefore $\delta(t) \in A$). The least upper bound of the last connected component of $\delta^{\text{-}1}(A)$ is thus t_1 and the image of the restriction $\delta|_{[t_1-\varepsilon,t_1]}$, for some $\varepsilon > 0$, is included in A. Then $\mathrm{img}(\delta|_{[t_1-\varepsilon,1]}) \subseteq A \cup B$ and $\partial^{+}\delta \in \mathrm{frw}(A, B)$. By duality, the result also holds for the backward operator. □

Note that the extra hypothesis is only required for the first parameter. Also remark that in the case where $X = \mathbb{R}$ is equipped with the chaotic direction (*i.e.* all continuous maps from $[0, 1]$ to $\mathbb{R}$ are directed) this property fails for all A but $\emptyset$ and $\mathbb{R}$. Given any $t \in \mathbb{R}$ there is indeed a path converging to t that oscillates infinitely many times around t. The isothetic regions behave much better.

Proposition 6.4.22. For all isothetic regions A and all dipaths δ on $\rceil G\lfloor^{n}$, the inverse image $\delta^{\text{-}1}(A)$ has finitely many connected components.

Proof. An immediate consequence of Proposition 6.1.7. □

Definition 6.4.23. To each point p of $(G_1, \ldots, G_n)$ in the sense of Definition 3.3.1 we associate the subblock B_p of $\rceil G_1\lfloor \times \cdots \times \rceil G_n\lfloor$ whose i^{th} component is $\{p_i\}$ when p_i is a vertex, and $\{p_i\}\times]0, 1[$ when it is an arrow. The blocks B_p form the **ginzu partition** of $\rceil G_1\lfloor \times \cdots \times \rceil G_n\lfloor$. By the way, note that defining the dimension of B_p as the number of arrows of p induces a filtration of $\rceil G_1\lfloor \times \cdots \times \rceil G_n\lfloor$ in the sense of (Brown et al., 2011, p.211). A region that can be written as a union of blocks of the canonical partition is said to be a **compatible** with it.

Remark 6.4.24. The ginzu partition is not intrinsically related to the product $\rceil G_1\lfloor \times \cdots \times \rceil G_n\lfloor$ since it partially depends on expandable vertices while the product of the directed realizations does not. In particular, given a finite family F of isothetic subregions of $\rceil G_1\lfloor \times \cdots \times \rceil G_n\lfloor$, there are graphs $G'_1, \ldots, G'_n$ such that $\mathrm{red}(G_i) \cong \mathrm{red}(G'_i)$ for all $i \in \{1, \ldots, n\}$ and the ginzu partition associated with $G'_1, \ldots, G'_n$ is compatible with all the members of the family F. For $i \in \{1, \ldots, n\}$, let G'_i be the graph whose vertices are all the vertices of G_i plus all the points of $\rceil G_i\lfloor$ that belongs to the boundary of $\mathrm{proj}_i(M)$ for some maximal subblock M of some member of the family F. The set V'_i of vertices of G'_i is thus a discrete subspace of $\rceil G_i\lfloor$, the arrows of G'_i are the connected components of $\rceil G_i\lfloor \backslash V'_i$. Each of them is isomorphic, as a pospace, with $]0, 1[$, so the source and the target maps of G'_i are defined accordingly. In particular one has an isomorphism Φ_i from $\rceil G_i\lfloor$ to $\rceil G'_i\lfloor$ whose restriction to V'_i is the identity.

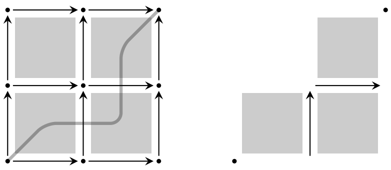


Figure 6.10: A directed path and its discretization.

The product map $\Phi = (\Phi_1, \ldots, \Phi_n)$ is thus an isomorphism from $\rceil G'_1 \lfloor \times \cdots \times \rceil G'_n \lfloor$ to $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$. Due to its specific form Φ also induces an isomorphism between the corresponding Boolean algebras of isothetic regions. In addition the direct image under Φ of every member of the family F is compatible with the ginzu partition of $\rceil G'_1 \lfloor \times \cdots \times \rceil G'_n \lfloor$. The graphs G_i and G'_i readily have isomorphic reduced graphs.

Lemma 6.4.25. If there exists a directed path starting in B_p, ending in $B_{p'}$, and whose image is contained in $B_p \cup B_{p'}$ (*i.e.* $\mathrm{frw}(B_p, B_{p'}) \neq B_p$ or equivalently $\mathrm{bck}(B_{p'}, B_p) \neq B_{p'}$) then we have the following facts:

- for all $i \in \{1, \ldots, n\}$, $p_i = p'_i$ or p_i is the source of the arrow p'_i, or
- for all $i \in \{1, \ldots, n\}$, $p_i = p'_i$ or p'_i is the target of the arrow p_i.

Proof. The inverse image $\gamma^{-1}(B_p)$ and $\gamma^{-1}(B_{p'})$ have finitely many connected components, it is a consequence of Proposition 6.4.22 and the standard equality thereinafter.

$$\gamma^{-1}(B_p) \quad = \quad \bigcap_{i=1}^{n} \gamma_i^{-1}(\mathrm{proj}_i(B_p)) \,.$$

Hence we have a partition of the domain of γ into intervals which are alternatively contained in $\gamma^{-1}(B_p)$ and $\gamma^{-1}(B_{p'})$, the first interval I being contained in the former while the second one J is contained in the latter. In particular we have $\sup I = \inf J$ which either belongs to I or J. The conclusion follows. □

Lemma 6.4.25 has been the motivation for Definition 3.3.2 in retrospect. See also Figure 6.10.

Corollary 6.4.26. Under the hypotheses of Lemma 6.4.25 we have

$$\mathrm{frw}(B_p, B_{p'}) = \mathrm{bck}(B_{p'}, B_p) = B_p \cup B_{p'} \,.$$

Definition 6.4.27. Given a directed path γ on $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$ (*cf.* Proposition 6.4.3) we have, by the same arguments as in the proof of Lemma 6.4.25, a finite partition $I_0 < \cdots < I_K$ of $\mathrm{dom}\,\gamma$ such that for all $k \in \{0, \ldots, K\}$, there exists a (necessarily unique) point $p(k)$ such that $\gamma(I_k) \subseteq B_{p(k)}$. By Lemma 6.4.25 the sequence $\overline{p} = p(0), \ldots, p(K)$ is a path in the sense of Definition 3.3.2. This latter is called the **discretization** of γ and denoted by $D(\gamma)$. Conversely, given a path $\overline{p}$ on $(G_1, \ldots, G_n)$ (*cf.* Definition 3.3.2) it is not difficult to find a directed path γ (*cf.* Definition 4.2.12) on $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$ whose discretization is $\overline{p}$, such a path γ is said to be a **lifting** of $\overline{p}$. The di scretization procedure is illustrated on Figure 6.10.

Theorem 6.4.28.
The Boolean algebra $\mathcal{R}_n$ is stable under forward and backward operators.

Proof. First we assume that all the graphs G_i are finite. Suppose that both A and B are compatible with the canonical partition (*cf.* Definition 6.4.23). By Corollary 6.4.26 any block of the ginzu partition met by some directed path on $A \cup B$ starting in A and ending in B, is entirely contained in $\mathrm{frw}(A, B)$. Since the canonical partition has finitely many blocks, the subspace $\mathrm{frw}(A, B)$ is an isothetic region. By Remark 6.4.24 we can suppose that the ginzu partition is compatible with both A and B. By Theorem 6.1.42 if the graph G_i is essentially finite then its directed geometric realization is canonically embedded in that of $\mathrm{red}(G_i)$ which is finite. We conclude because the product map

$$\upharpoonleft G_1 \downharpoonleft \times \cdots \times \upharpoonleft G_n \downharpoonleft \quad \hookrightarrow \quad \upharpoonleft \mathrm{red}(G_1) \downharpoonleft \times \cdots \times \upharpoonleft \mathrm{red}(G_n) \downharpoonleft$$

preserves and reflects the isothetic regions. □

Corollary 6.4.29.
Given isothetic regions A and B, the sets $\overline{A}^{\mathrm{f}}$, $\overline{A}^{\mathrm{p}}$, $\mathrm{frw}(\overline{A}^{\mathrm{f}} \cap B, B)$, $\mathrm{bck}(\overline{A}^{\mathrm{p}} \cap B, B)$, $\mathrm{frw}(A \cap \overline{B}^{\mathrm{p}}, B)$, and $\mathrm{bck}(A \cap \overline{B}^{\mathrm{f}}, B)$ are isothetic regions.

Corollary 6.4.29 combined with Proposition 6.4.22 provides the algorithm that enables the ALCOOL software to compute forward and backward operators.

Corollary 6.4.30. If $A \subseteq X$ are isothetic regions, then the future attractor and the past attractor (*cf.* Definition 6.4.18) of A (in X) are isothetic regions.

Proof. By Proposition 6.4.19. □

6.5 Metric Properties of Regions

So far we have endowed isothetic regions with topology and direction. These mathematical concepts do not allow more than a qualitative analysis. For example, one can tell whether a program may freeze from the presence of a deadlock in its model, but one cannot tell how likely it is to happen. We are not either able to measure the speed of an execution trace. We provide isothetic regions with extra structures inspired by the standard distance and the Lebesgue measure on $\mathbb{R}$. We do not pretend that the constructions proposed thereafter enable a satisfactory quantitative analysis, yet, they are natural and therefore deserve to be mentioned. Moreover, as we shall see in Chapter 9, any isothetic region (*cf.* Definition 6.2.1) can be written as a Cartesian product of "irreducible" isothetic regions in a unique way (*cf.* Theorem 9.3.2). Knowing that, we wonder if the additional structures are preserved by decomposition.

The length $\ell(\gamma)$ of a path $\gamma : [a, b] \to (X, d)$ on a metric space is defined as the supremum of

$$\sum_{i=1}^{N} d(\gamma(t_{i-1}), \gamma(t_i))$$

for all finite sequences $a = t_0 \leqslant \cdots \leqslant t_N = b$. A **length-metric** is a metric such that the distance between two points is the infimum of the lengths of the paths joining them – see (Bridson and Haefliger, 1999, Chap.I.3), (Epstein, 1992, Sect.3.1), (Papadopoulos, 2013, p.11), or (Gromov, 2007, Chap.1) for a slightly different approach. Length-metrics are actually the significant ones though they raise a technical problem: the metric induced on a subspace of a length metric is generally not a length-metric. However the problem is solved by a standard construction: any metric space (X, d) is associated with a length-metric, the so-called **intrinsic metric**, defined for all $x, y \in X$ by

$$d'(x, y) = \inf\{\ell(\alpha) \mid \alpha \text{ paths joining } x \text{ and } y\}$$

If no such path exists (*i.e.* when the points do not lie in the same path-connected component) the distance between them is conventionally defined as infinite. Even further, a length-metric space is said to be **geodesic** when any couple of its points are connected by a path whose length is the distance between these points – see (Bridson and Haefliger, 1999, p.9). The geometric realization of any graph is known to admit a canonical metric structure assuming that any arrow is of length 1. Such metric spaces are called *metric graphs* – see (Bridson and Haefliger, 1999, Chap.1). Every 1-dimensional isothetic region A is actually a geodesic space by considering the intrinsic metric derived from the metric induced on A by the metric graph structure on $|G|$. This situation is rather exceptional, indeed the intrinsic metric induced on a subset of a geodesic metric space is not geodesic in general.

Example 6.5.1. Observe that the punctured plane (*i.e.* $\mathbb{R}^2 \setminus \{0\}$) equipped with the Euclidean distance (resp. the max distance) is not geodesic tough it is a length-metric.

This section aims at providing any isothetic region with a length-metric. It is well-known that given two metric spaces (X, d_X) and (Y, d_Y) there are many ways to provide the topological space $X \times Y$ with a metric. The Euclidean metric

$$d_{X\times Y} = \sqrt{d_X^2 + d_Y^2}$$

and the max-metric

$$d_{X\times Y} = \max(d_X, d_Y)$$

are two examples of such metrics. One even knows that if both d_X and d_Y are geodesic metrics, then so are their Euclidean product and their max-product. In accord with our intention to measure the speed of a parallel execution trace, we equip every n-dimensional subregion of $\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright$ with the intrinsic metric induced by the max-product, we will assume that each $\upharpoonleft G_k \downharpoonright$ is a metric (control flow) graph, and that “parallel execution trace” actually means “directed path whose interpretation is an execution trace indeed”. If we had been interested in interleaving execution traces, then we would have chosen the following product.

$$d_{\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright} = d_{\upharpoonleft G_1 \downharpoonright} + \cdots + d_{\upharpoonleft G_n \downharpoonright}$$

Remark 6.5.2. Any isothetic region is a Polish space (*i.e.* a complete separable metric space).

Remark 6.5.3. According to Hopf-Rinow Theorem – see (Bridson and Haefliger, 1999, p.35), if the length-metric space we have defined is complete and its underlying topology is locally compact, then it is geodesic regardless of the chosen metric product. Unfortunately the continuous model of a Paml program (*cf.* Definition 7.1.2) generally is neither complete nor geodesic: this will matter in Section 9.6.

Example 6.5.4. The punctured plane (*cf.* Example 6.5.1), which is not geodesic, is the continuous model of the following program.

```
var:  x = 0
proc:  p = x:=0
init:  2p
```

We now focus on measures over isothetic regions. The measure associated with an isothetic region A will be denoted by μ_A or just μ in the sequel. Let us start by considering the case where A is the realization of an essentially finite graph G. We define the measure of a connected subset X of the geometric realization of G by remarking that its interior and its closure have the same measure. Indeed, either X is reduced to a segment that satisfies

$$\{\alpha\}\times]a,b[\ \subseteq\ X\ \subseteq\ \{\alpha\}\times[a,b]$$

for a unique arrow α together with $0 < a \leqslant b < 1$; or for all arrows α of G there exists $0 \leqslant a_\alpha \leqslant b_\alpha \leqslant 1$ such that

$$\{\alpha\}\times(]0,a_\alpha[\cup]b_\alpha,1[)\ \subseteq\ X\cap\{\alpha\}\times]0,1[\ \subseteq\ \{\alpha\}\times(]0,a_\alpha]\cup[b_\alpha,1[)$$

The measure of X is thus defined as $b-a$ in the first case and

$$\mu_G(X) = \sum_{\alpha\in G}(1-b_\alpha+a_\alpha)$$

in the second one. The preceding sum makes sense because the indexing set is countable by Remark 6.1.21. Since any element of $\mathcal{R}_G$ is the finite disjoint union of its connected components, the previous definition extends to a measure μ_G over $\mathcal{R}_G$. Since G is essentially finite, this measure is σ-finite, hence Carathéodory's theorem[5] extends μ_G to a unique measure, still denoted by μ_G, over the complete Borel σ-algebra of $|G|$ (*i.e.* the least σ-algebra containing all the open subsets of $|G|$ as well as any subset of a neglectable[6] subset of $|G|$). Assuming we are given the Lebesgue measure on $]0,1[$ and following the description of the realization of G given in Section 6.1, μ_G can also be defined as the sum

$$\mu_G(X) = \sum_{\alpha\in G}\lambda_{]0,1[}(X\cap\{\alpha\}\times]0,1[)$$

being understood that the set of vertices of G is neglectable. Given an n-tuple of essentially finite graphs, the isothetic region $|G_1|\times\cdots\times|G_n|$ can therefore be equipped with the tensor product of measures (*cf.* (Halmos, 1974, Chap.VII))

$$\mu_{G_1,\dots,G_n} = \mu_{G_1}\otimes\cdots\otimes\mu_{G_n}$$

Remark 6.5.5.
As a distinctive feature, if all the graphs G_k equal $\{\cdot\longrightarrow\cdot\}$ then $\mu(|G_1|\times\cdots\times|G_n|) = 1$. In particular, if all the graphs G_k equal $\mathbb{Z}$ (and therefore $|G_k| = \mathbb{R}$) then $\mu_{G_1,\cdots,G_n}$ boils down to the usual Lebesgue measure over $\mathbb{R}^n$.

Now we would like to provide any subregion A of $|G_1|\times\cdots\times|G_n|$ with a relevant measure. A naive approach consists of defining μ_A as the restriction of $\mu_{G_1,\dots,G_n}$ to measurable subsets of A but this measure is null whenever $\mu_{G_1,\dots,G_n}(A) = 0$. The problem is that Lebesgue measure on $\mathbb{R}^n$, on which $\mu_{G_1,\dots,G_n}$ is based, is not adapted to subsets whose "intrinsic dimension" is strictly lower than n. Formalizing the notion of "intrinsic dimension" and providing the corresponding measure are the first steps to Geometric Measure Theory (Morgan, 2008, chapter 1). There are actually several such measures, all of them being based on the Carathéodory construction (*cf.* (Federer,

[5] See Cohn (1980), Folland (1999), Bogachev (2007), Klenke (2008) or Fremlin (2011).

[6] The neglectable subsets are defined as the ones whose measure is null.

2013, Section 2.10, p.169) or (Krantz and Parks, 1999, Section 3.1, p.57)). The crucial point is that all of them agree on **rectifiable**[7] sets (*cf.* (Federer, 2013, 3.2.14, p.251 and 3.2.26, p.261) or (Krantz and Parks, 1999, 3.2.9-10, p.70-71)). So we can define the **intrinsic measure** of a rectifiable set of intrinsic dimension d as its d-dimensional Hausdorff measure[8]. One readily checks that any product $I_1 \times \cdots \times I_n$ of intervals of $\mathbb{R}$ is rectifiable and its intrinsic dimension d is the cardinal of the family

$$\mathcal{F} = \left\{ k \;\middle|\; I_k \text{ is not a singleton} \right\}$$

Its intrinsic measure is then defined, for all measurable sets X, as the d-dimensional Lebesgue measure of $\mathrm{proj}_{\mathcal{F}}(X)$. By extension we define the intrinsic dimension of a connected block (*cf.* Definition 6.2.12) $C_1 \times \cdots \times C_n \subseteq |G_1| \times \cdots \times |G_n|$ as the cardinal of the family

$$\mathcal{F} = \left\{ k \;\middle|\; C_k \text{ is not a singleton} \right\} = \{ i_1 < \ldots < i_d \} \; .$$

Its intrinsic measure $\mu_{C_1 \times \cdots \times C_n}$ is the tensor product of measures

$$\mu_{C_{i_1}} \otimes \cdots \otimes \mu_{C_{i_d}}$$

where μ_{C_k} is the restriction of $\mu_{|G_k|}$ to measurable subsets of C_k. From the connected ginzu partition $\mathcal{C}$ of A (*cf.* Definition 6.4.23) we define

$$\mu_A(X) \quad := \quad \sum_{C \in \mathcal{C}} \mu_C(C \cap X)$$

where μ_C is the intrinsic measure of the connected block C.

Remark 6.5.6. In general, an isothetic region A is "heterogeneous" in the sense that the elements of its connected ginzu partition do not have the same intrinsic dimension. The measure μ_A is designed to take "locally intrinsic" dimension into account. For example, the connected ginzu partition of the punctured plane (*cf.* Example 6.5.1) has 8 elements, namely $(\mathbb{R}_+ \setminus \{0\})^2$, $(\mathbb{R}_+ \setminus \{0\}) \times \{0\}$ and their images under the $\frac{\pi}{2}$ rotation. According to our construction, the measure of $[-1, 1]^2 \setminus \{(0, 0)\}$ is 8 though its 2-dimensional Lebesgue measure is 4.

[7] Rectifiable subsets of $\mathbb{R}^n$ are characterized in Preiss (1987) and De Lellis (2008).

[8] Hausdorff measure is studied with much attention in the second chapter of Evans and Gariepy (1992).

7

Continuous Semantics

of the Parallel Automata Meta Language

This chapter aims at proving that conservative programs are profitably modelled within the class of isothetic regions (*cf.* Definition 6.2.1). Let $G_1, \dots, G_n$ be the control flow graphs (*cf.* Definition 3.2.1) of the running processes of a program P, we denote the set of vertices of G_i by V_i. In the light of Definition 6.4.27 we partly override Definitions 3.3.1 and 3.3.2 to consider points and directed paths on the local pospace $\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright$. In terms of instruction pointer dynamics, this approach imposes a paradigm shift about the meaning of arrows of control flow graphs. Referring to Definition 3.3.7 we interpret arrows as intermediate positions between instructions. In this context being inbetween instructions is a qualitative statement. After each arrow of a control flow graph has been replaced by a copy of the open segment $]0, 1[$ (by considering the geometric realization of the graph (*cf.* Section 6.1)) the abovementioned statement becomes quantitative. Indeed an intermediate point can be "close to" the preceding instruction (which is carried by the source of the arrow along with which the instruction pointer moves) or "almost on" the next one (which is carried by the target of this arrow). In this chapter we mainly handle isothetic regions. In that context, the term "directed path" (or "dipath" for short) should therefore be understood in the light of Proposition 6.4.3.

7.1 Switching to the continous framework

We adapt Sections 3.3 and 3.4 to the isothetic region context. Roughly speaking, we replace a control flow graph by its directed geometric realization. Leaning on Definition 6.4.27 the sequence of mutli-instructions associated with a directed path γ on $\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright$ is the one associated with its discretization. Hence following Definition 3.3.7, the directed path γ is **admissible** (resp. is an **execution trace**) when so is $D(\gamma)$. If γ is admissible, we define its **action** on (the right of) a state σ as $\sigma \cdot D(\gamma)$.

Remark 7.1.1. Due to the convention that the effect of an instruction is visible at the very moment it is reached, there exists a partition of intervals $I'_0 < \cdots < I'_{K'}$ of $\operatorname{dom} \gamma$, which is coarser than the partition $I_0 < \cdots < I_K$ of Definition 6.4.27, and such that for all $k \in \{0, \dots, K\}'$, the interval I'_k is closed on the left and open on the right, and such that for all $t, t' \in I'_k$, we have the following equality.

$$\delta_0 \cdot \gamma|_{[0,t]} \quad = \quad \delta_0 \cdot \gamma|_{[0,t']}$$

Assuming that the (middle-end representation of the) program is conservative (*cf.* Definition 3.4.1) the potential function F from Definition 3.4.8 is defined for all (p, s) with $s \in \mathcal{S}$ and p point of $(G_1, \ldots, G_n)$ in the sense of Definition 3.3.1. From F we derive another **potential** function by setting $F'(p', s) = F(p, s)$ for all $p' \in B_p$ with B_p defined in Definition 6.4.27.

$$F' : \rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor \times \mathcal{S} \to \mathbb{N}$$

Hence Definition 3.4.10 still makes sense when we consider points of $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$ and the potential function F' instead of F.

Definition 7.1.2. The **continuous model** of the program P, denoted by $[\![P]\!]$, is the complement of the **forbidden** subspace of $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$.

Remark 7.1.3. Referring to the ginzu partition (*cf.* Definition 6.4.23), the continuous model of the program P can also be written as follows

$$\bigcup_{d \in \mathbb{N}} (K_d \setminus F_d) \times]0, 1[^d$$

with K_d (resp. F_d) being the set of d-dimensional (resp. forbidden) points of the tensor product $G_1 \otimes \cdots \otimes G_n$ (*cf.* Definitions 2.4.12 and 3.4.10). This remark is illustrated by Figure 7.1. Analogously, the deadlock subspace can be described as the following union

$$\bigcup_{d \in \mathbb{N}} D_d \times]0, 1[^d$$

where D_d is the set of d-dimensional deadlock points (*cf.* Definition 3.4.10).

Remark 7.1.4. Theorem 3.4.11 is still valid with the terms "directed path", "forbidden point", and "admissible" being understood in the context of $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$. The continuous version of Theorem 3.4.11 casts a glow on the example shown on Figure 3.13: the 'replacement' it proposes might seem cryptic in the discrete setting but, as shown on Figure 7.2, it becomes obvious in the light of the continuous one.

We substitute the discrete model from Definition 3.4.10 with an isothetic region which is compatible with the ginzu partition associated with the control flow graphs (*cf.* Definition 6.4.23).

Proposition 7.1.5. The continuous model, the deadlock subspace, and the deadlock attractor of (the middle-end representation of) a program are isothetic regions. Their common dimension being the number of running processes of the program (*i.e.* the mass of its bootup multiset, see Definition 1.1.7.).

Proof. Referring to Definition 6.4.27, observe that each of the blocks B_p of the ginzu partition of $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$, which is finite, is either contained in the forbidden subspace or disjoint from it, depending on the point p being forbidden or not in the sense of Definition 3.4.10. The same remark applies to the subspace of deadlocks of the program. One deduces from Corollary 6.4.30 that the deadlock attractor is also an isothetic region. □

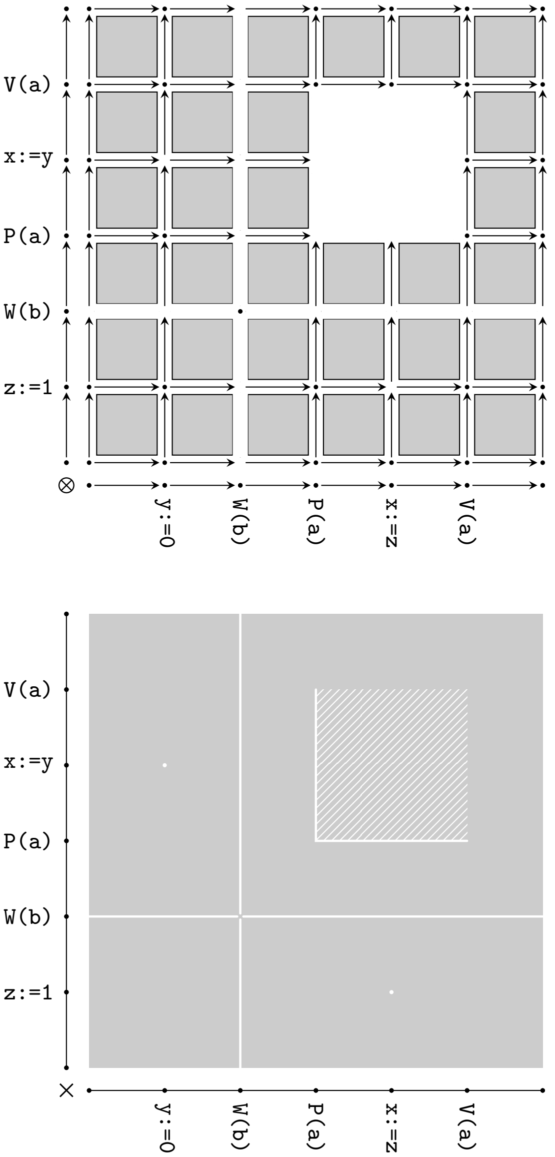


Figure 7.1: Discrete model *vs* continuous model

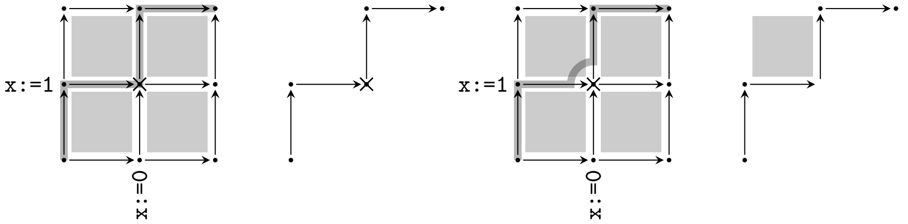


Figure 7.2: Comparing the discrete and the continuous approaches: an admissible directed path that meets a forbidden point and a possible replacement for it.

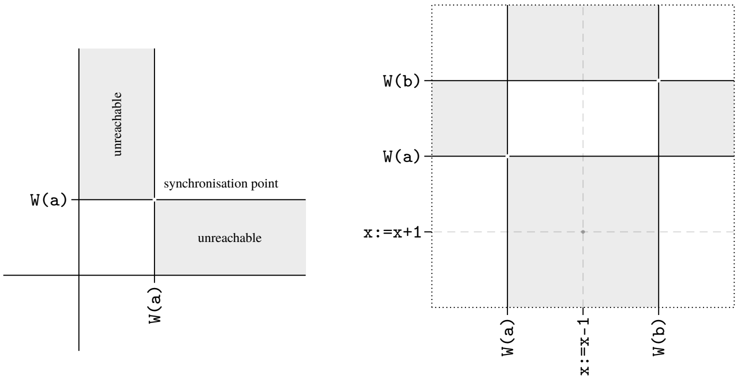


Figure 7.3: Binary synchronisation: producer *vs* consumer on a flat torus (the opposite edges of the dotted frame are identified).

On this occasion we provide several illustrations.

Example 7.1.6. The left-hand model on Figure 7.3 illustrates a binary synchronisation. The right-hand one represents the ‘producer *vs* consumer’ situation introduced in Example 1.1.9. Using synchronisation barriers we ensure that items are made and delivered just-in-time. Note that this example lie on the directed torus, each dotted edge being identified with its opposite. On both models, the grayed out subspace is **unreachable** from the origin of the model.

Example 7.1.7. The models shown on Figure 7.4 are standard examples of deadlocking programs introduced in Example 1.1.10. The “Swiss Cross” model and the “dining philosophers” problem appear in Coffman et al. (1971) and Dijkstra (1971) respectively.

Example 7.1.8. The models shown on Figure 7.5 illustrate how drastically sensitive the model of a program is to the arities of the semaphores it uses. The left hand model is obtained with a semaphore of arity 1 while the right hand one is obtained by setting the arity to 2.

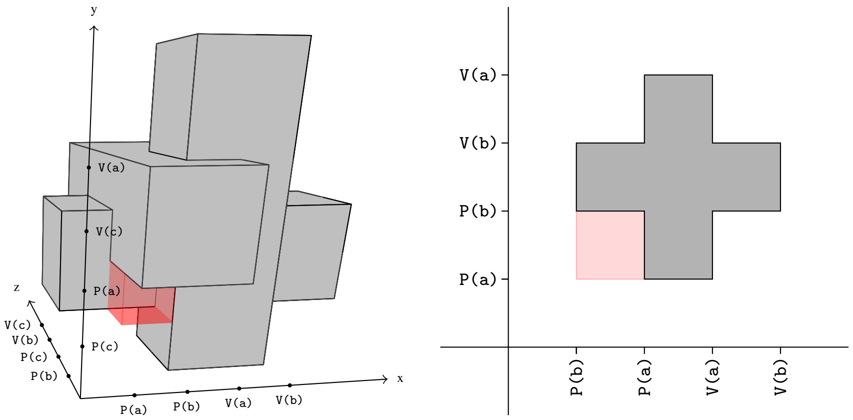


Figure 7.4: The three dining philosophers and the Swiss Cross with their deadlock attractors (in red).

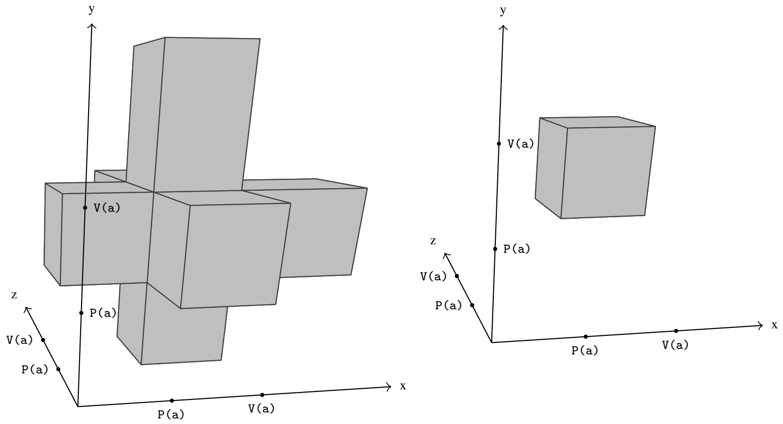
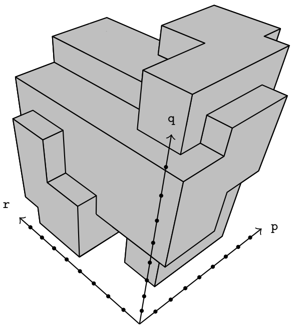


Figure 7.5: The tetrahemihexacron a.k.a. 3D Swiss Cross, and the ‘floating’ cube.



```
sem 1:  u v w x y z
proc:
  p = P(x);P(y);P(z);V(x);P(w);V(z);V(y);V(w)
  q = P(u);P(v);P(x);V(u);P(z);V(v);V(x);V(z)
  r = P(y);P(w);V(y);P(u);V(w);P(v);V(u);V(v)
init:  p q r
```

Figure 7.6: The continuous model of the Lipski algorithm.

Example 7.1.9. The model of the Lipski algorithm, depicted on Figure 7.6, was introduced in Lipski and Papadimitriou (1981) as an example of program without deadlock though its "request graph" has cycles. A careful examination reveals that there is indeed a "tunnel" going through the forbidden subspace.

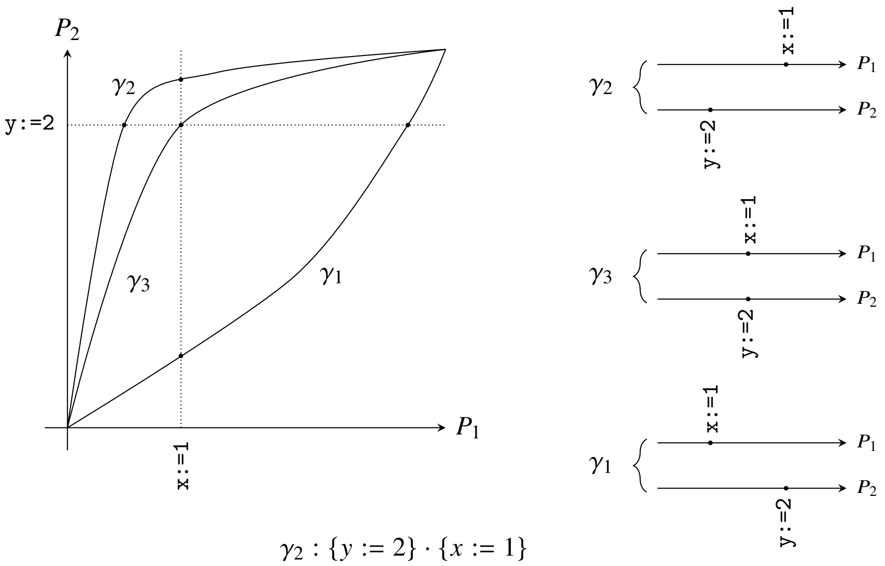
7.2 Justifying the Topological Approach

This section is devoted to a theorem which justifies the use of algebraic topology methods in the study of concurrency. In computer science terms, it says that dihomotopy of directed paths is sound with respect to Paml programs semantics, thus validating the construction described in Definition 7.1.2. In order to state it properly, we relax the notion of alternating dihomotopy (*cf.* Definition 5.2.5).

Definition 7.2.1. A **weak dihomotopy** is a homotopy (*cf.* Definition 5.1.1) whose intermediate paths are directed. In the context of this manuscript, two directed paths are said to be **weakly dihomotopic** when there exists a weak dihomotopy between them. The effect of weakly directed homotopies on the sequences of multi-instructions associated to the intermediate dipaths is illustrated on Figure 7.7.

Example 7.2.2. Any alternating homotopy is a weak dihomotopy but the converse is false as shown by the sphere $\mathbb{S}^2$ directed by its meridians (*i.e.* the Riemann sphere with paths of the form $t \in \mathbb{R}_+ \cup \{\infty\} \mapsto z \cdot t$ with $z \in \mathbb{C} \setminus \{0\}$). Any two meridians are indeed weakly dihomotopic with one another though both of them are dihomotopic iff they have the same image. This counterexample is an instance of directed suspension above a chaotic d-space X: form the product $X \times [0,1]$ then identify all the points $(x, 0)$ (resp. $(x, 1)$) with a single one, being understood that $[0,1]$ is directed. However Fajstrup (2005) proves that the notions coincide for the so-called cubical complexes (*i.e.* realization of geometric precubical sets as local pospaces).

Weak dihomotopies have been introduced with a view to Theorem 7.2.4. However, for theoretical purposes, alternating dihomotopies are easier to work with (*cf.* Remark 5.4.2). In addition, (Fajstrup, 2005, Theorem 5.7, p.203-204), and in a more general setting (Krishnan, 2013, Theorem 8.13, p.208), have proved that the notions coincide on $\upharpoonleft K \downharpoonleft_{\mathbf{LpoTop}}$ and $\upharpoonleft K \downharpoonleft_{\mathbf{Strm}}$ for a large class of precubical and cubical sets. Strictly speaking, isothetic regions do not fall in that class but we are confident that the following conjecture holds.



$$\gamma_2 : \{y := 2\} \cdot \{x := 1\}$$

$$\gamma_3 : \left\{ \begin{array}{c} x := 1 \\ y := 2 \end{array} \right\}$$

$$\gamma_1 : \{x := 1\} \cdot \{y := 2\}$$

Figure 7.7: Timelines and sequences of multi-instructions interpreting weakly dihomotopic directed paths.

Conjecture 7.2.3. Weakly dihomotopic paths on an isothetic region are dihomotopic.

Theorem 7.2.4. Two weakly dihomotopic directed paths on the continuous model of a conservative program induce the same action over the valuations. Moreover, one is an execution trace if and only if the other is so.

Proof. Let X be the continuous model of a conservative program and suppose we are given a weakly directed homotopy $h : [0,r] \times [0,q] \longrightarrow U(X)$. By Definition 7.1.2 the model X is a subobject of $\upharpoonleft G_1 \downharpoonright \times \cdots \times \upharpoonleft G_n \downharpoonright$ with each G_i being the control flow graph of a process. The set of vertices of G_i is denoted by V_i. The labelling on $\upharpoonleft G_i \downharpoonright$ is the mapping $\lambda_i : V_i \longrightarrow \{\text{instructions}\}$ and we define $h_i = \text{proj}_i \circ h$. For all $s \in [0,q]$ the image of the directed path $h(\_, s)$ lies in the continuous model, hence it is admissible by Theorem 3.4.11 which is still valid in the continuous framework (*cf.* Remark 7.1.4). Fix $s \in [0,q]$. In each coordinate $i \in \{1,\dots,n\}$, the subspace $(h_i(\_, s))^{-1}(V_i)$ is the union of the following finite sequence of disconnected compact intervals because $(h_i(\_, s))$ is a directed path (*cf.* Proposition 6.1.7).

$$[a_1^{(i)}(s), b_1^{(i)}(s)] < \cdots < [a_{N_i}^{(i)}(s), b_{N_i}^{(i)}(s)]$$

Define $Y(s) = \{a_k^{(i)}(s) \mid i \in \{1,\dots,n\} \ ; \ k \in \{1,\dots,N_i\}\}$ and let y be some of its elements. We consider $E_y(s)$ the set of $i \in \{1,\dots,n\}$ such that y is the minimum of some connected component of $(h_i(\_, s))^{-1}(V_i)$, or y belongs to this connected component and $\lambda_i(h_i(y,s))$ is a synchronisation $W(\_)$. In other words $E_y(s)$ is the subset of $\{1,\dots,n\}$ whose elements i satisfy the conditions below.

$$\exists k \in \{1,\dots,N_i\};\ y \in [a_k^{(i)}(s), b_k^{(i)}(s)] \quad \wedge \quad y \neq a_k^{(i)}(s) \ \Rightarrow\ \lambda_i(h_i(y,s)) = W(\_)$$

The multi-instruction $\mu_y(s)$ is defined over $E_y(s)$ by $(\mu_y(s))(i) = \lambda_i(h_i(y,s))$ which is the instruction carried by the point $h_i(y,s)$ of $\upharpoonleft G_i \downharpoonright$. Since $Y(s)$ is totally ordered (as a subset of $\mathbb{R}$) so are the multi-instructions $\mu_y(s)$. The latter actually forms the finite sequence associated with the discretization of $h(\_, s)$ (*cf.* Section 7.1). It is denoted by $[\![h(\_, s)]\!]$ in the sequel. Then for each $i \in \{1,\dots,n\}$ choose $t_i > 0$ so that the intervals $[a_k^{(i)}(s) - t_i, b_k^{(i)}(s) + t_i]$ are still pairwise disjoint. For h is continuous, one has $\varepsilon_i > 0$ such that for all $s' \in [0,q]$, if $|s - s'| \leqslant \varepsilon_i$ then $(h_i(\_, s))^{-1}(V_i)$ and $(h_i(\_, s'))^{-1}(V_i)$ have the same number of connected components, viz N_i, and for all $k \in \{1,\dots,N_i\}$, the following inclusions hold.

$$[a_k^{(i)}(s'), b_k^{(i)}(s')] \quad \subseteq \quad [a_k^{(i)}(s) - t_i, b_k^{(i)}(s) + t_i]$$

Define $\varepsilon' = \min\{\varepsilon_1,\dots,\varepsilon_n\}$ and let $s' \in [0,q]$ be satisfying $|s - s'| \leqslant \varepsilon'$. For $Y(s)$ is finite we can actually find $\varepsilon \in]0, \varepsilon']$ such that for all $i, i' \in \{1,\dots,n\}$, for all $k \in \{1,\dots,N_i\}$, for all $k' \in \{1,\dots,N_{i'}\}$, for all $s' \in [0,q]$ such that $|s - s'| \leqslant \varepsilon$, the following statement is satisfied.

$$a_k^{(i)}(s) < a_{k'}^{(i')}(s) \quad \Rightarrow \quad a_k^{(i)}(s') < a_{k'}^{(i')}(s')$$

Enumerating $Y(s) = \{y_1 < \ldots < y_m\}$, it exactly means that if s' is close enough to s, then one can write $[\![h(\_, s')]\!]$ as an appending of finite sequences of multi-instructions $S_1,\dots,S_m$ such that for all $p \in \{1,\dots,m\}$, the domains of definition of the multi-instructions contained in S_p are disjoint and their union is $\mu_{y_p}(s)$ (*i.e.* one recover $[\![h(\_, s)]\!]$ from $[\![h(\_, s')]\!]$ by gathering certain consecutive elements). We write ε_s

instead of ε to stress the dependency on s. The open intervals $]s-\varepsilon_s, s+\varepsilon_s[$ form an open covering of the compact interval $[0,q]$. Let ε be some Lebesgue number of the covering. By Corollary 3.4.14, for all $s, s' \in]0,q[$ such that $|s-s'| \leqslant \varepsilon$ both directed paths $h(\_,s)$ and $h(\_,s')$ induce the same action on valuations, and $h(\_,s)$ is an execution trace iff so is $h(\_,s')$. The conclusion follows. □

Corollary 7.2.5. The fundamental category of the continuous model of a (conservative) Paml program acts on valuations.

7.3 Independence of Conservative Programs

Building on the notion of continuous model (*cf.* Definition 7.1.2) and following the intuition that the Cartesian product of models should represent the parallel composition of the corresponding programs, we introduce another notion of independence that applies to conservative programs.

Definition 7.3.1. The Paml programs $P_1,\ldots,P_N$ are said to be model independent when they make coherent declarations (*cf.* Definition 1.5.1) and the following equality holds.

$$[\![P_1|\ldots|P_N]\!] = [\![P_1]\!] \times \cdots \times [\![P_N]\!]$$

Proposition 7.3.2. Syntactically independent conservative programs are model independent.

Proof. The forbidden region F_ℓ of each of the conservative programs P_ℓ for $\ell \in \{1,\ldots,L\}$ is a subobject of $\rceil G_1^{(\ell)} \lfloor \times \cdots \times \rceil G_{N_\ell}^{(\ell)} \lfloor$ which we denote by Ω_ℓ. The continuous model of the parallel compound $[\![P_1|\cdots|P_L]\!]$ is thus a subregion of the product

$$\underbrace{\rceil G_1^{(1)}\lfloor \times \cdots \times \rceil G_{N_1}^{(1)}\lfloor}_{\Omega_1} \times \cdots \times \underbrace{\rceil G_1^{(L)}\lfloor \times \cdots \times \rceil G_{N_L}^{(L)}\lfloor}_{\Omega_L} \quad .$$

Because the programs P_ℓ are syntactically independent, the forbidden region of the parallel compound $P_1|\cdots|P_L$ generated by P_ℓ has the form

$$\Omega_1 \times \cdots \times \Omega_{\ell-1} \times F_\ell \times \Omega_{\ell+1} \times \cdots \times \Omega_L \quad .$$

So the forbidden region of the parallel compound $P_1|\cdots|P_L$ is the following one

$$\bigcup_{\ell=1}^{L} \Omega_1 \times \cdots \times \Omega_{\ell-1} \times F_\ell \times \Omega_{\ell+1} \times \cdots \times \Omega_L \quad .$$

The model $[\![P_1|\cdots|P_L]\!]$ of the parallel compound is thus the complement of the preceding region.

$$\bigcap_{\ell=1}^{L} \Omega_1 \times \cdots \times \Omega_{\ell-1} \times (\Omega_\ell \setminus F_\ell) \times \Omega_{\ell+1} \times \cdots \times \Omega_L$$

The region above can be rewritten as the following one

$$(\Omega_1 \setminus F_1) \times \cdots \times (\Omega_\ell \setminus F_\ell) \times \cdots \times (\Omega_L \setminus F_L)$$

which is precisely the product of the continuous models of the programs P_ℓ.

$$[\![P_1]\!] \times \cdots \times [\![P_\ell]\!] \times \cdots \times [\![P_L]\!]$$

□

As illustrated by the next example model independent programs might not be syntactically independent. However, such a situation suggests that the source code of the program under consideration can be cleaned up.

Example 7.3.3. Let us denote by P (resp. Q) the left-hand (resp. right-hand) program thereinafter.

```
sem:  1 a                              sem:  1 b
sem:  2 c                              sem:  2 c
proc:  p = P(a);P(c);V(c);V(a)         proc:  q = P(b);P(c);V(c);V(b)
init:  2p                              init:  2q
```

The programs P and Q make coherent declaration (*cf.* Definition 1.5.1) so we can form their parallel composition $P|Q$ yet they are readily not syntactically independent (*cf.* Definition 1.5.3). A direct but rather tedious calculation proves that $[\![P|Q]\!] = [\![P]\!] \times [\![Q]\!]$ in other words that P and Q are model independent (*cf.* Definition 7.3.1). We also observe that due to the mutual exclusion a, the program P cannot hold more that one occurrence of the semaphore c. The same remark can be made about the program Q. Hence no process of the parallel composition $P|Q$ can be hindered by a lack of semaphore c. In other words the parallel composition $P|Q$ never uses more than two occurrences of c. From the above remarks, we deduce that P and Q are observationally independent (*cf.* Definition 1.5.7). We also deduce that the forbidden region generated by c is included in the forbidden region generated by a and b. From the computer science point of view, it means that c has no influence on the executions of the program $P|Q$ so it is harmless to drop it. Denote by P' and Q' the programs obtained by removing the instructions $P(c)$ and $V(c)$ from P and Q, then P' and Q' are syntactically independent and we have $[\![P|Q]\!] \cong [\![P'|Q']\!]$. This example is due to Balabonski and Haucourt (2010).

Theorem 7.3.4. Model independent conservative programs are observationally independent.

Proof. Let $S_1 \sqcup \cdots \sqcup S_N$ be the associated partition of the set of running processes of the parallel composition $P_1|\cdots|P_N$ (*cf.* Definition 1.5.7) with n being the total number of running processes. Let δ be an execution trace (*cf.* Definition 2.1.7) and denote by $(\mu_0, \ldots, \mu_q)$ its associated sequence of multi-instructions. First, we treat the case where the permutation is a rolling $(0 \cdots q)$ compatible with δ (*cf.* Definitions 1.5.7 and 1.5.10). The idea is to consider a lifting of δ and to deform it (by a weakly directed homotopy) so it becomes a directed path whose discretization is associated with the sequence of multi-instructions $(\mu'_0, \ldots, \mu'_q)$ with $\mu'_k = \mu_{\rho^{-1}(k)}$. If we succeed, then we have by Theorem 7.2.4 that both sequences of multi-instructions $(\mu_0, \ldots, \mu_q)$ and $(\mu'_0, \ldots, \mu'_q)$ come from execution traces and induce the same action over valuations. The general case will follow from the rolling decomposition of the compatible permutation to treat (*cf.* Lemma 1.5.12). As ρ is compatible with δ we have $J \cap J' = \emptyset$ with J and J' befined thereinunder.

$$J \quad = \quad \big\{ j \in \{1, \ldots, N\} \;\big|\; \operatorname{dom} \mu_q \cap S_j \neq \emptyset \big\}$$

$$J' \quad = \quad \big\{ j \in \{1,\ldots,N\} \;\big|\; \operatorname{dom}\mu_k \cap S_j \neq \emptyset \text{ for some } k \in \{0,\ldots,q-1\} \big\}$$

The projections proj_J and $\mathrm{proj}_{J'}$ send a point of $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$ to the extracted tuples of components whose indices respectively belong to the following sets.

$$\bigcup_{j \in J} S_j \qquad\qquad \bigcup_{j' \in J'} S_{j'}$$

Therefore we have a lifting γ of δ (*cf.* Definition 6.4.27) that can be written as a concatenation $\gamma = \gamma_{J'} \cdot \gamma_J$ such that $\mathrm{proj}_J \circ \gamma_{J'}$ and $\mathrm{proj}_{J'} \circ \gamma_J$ are constant. As an instance of Godement exchange law between composition and concatenation we have the directed paths γ_1 and γ_2 defined below.

$$\gamma_1 = \mathrm{proj}_{J'} \circ \gamma = \gamma_{J'} \cdot \mathrm{cst} \qquad\qquad \gamma_2 = \mathrm{proj}_J \circ \gamma = \mathrm{cst} \cdot \gamma_J$$

So we have $\gamma = (\gamma_1, \gamma_2)$ with the convention that we omit all the 'harmless' components, that is to say the ones in following set.

$$\{1,\ldots,n\} \quad \backslash \bigcup_{j \in J \cup J'} S_j \ .$$

Since $P_1, \ldots, P_N$ are model independent, the continuous model $[\![P_1|\cdots|P_N]\!]$ is the local pospace product $[\![P_1]\!] \times \cdots \times [\![P_N]\!]$. Therefore, assuming that both γ_1 and γ_2 are defined over $[0,1]$, the mapping thereinafter induces a local pospace morphism from $[0,1]^2$ to the continuous model of the parallel composition.

$$(x, y) \mapsto (\gamma_1(x), \gamma_2(y))$$

Precomposing with a weakly directed homotopy from $[0,1] \times \{0\} \ \cup \ \{1\} \times [0,1]$ to $\{0\} \times [0,1] \ \cup \ [0,1] \times \{1\}$ we obtain a weakly directed homotopy from $\gamma = \gamma_{J'} \cdot \gamma_J$ to $\gamma' = \gamma'_J \cdot \gamma'_{J'}$ with $\gamma_{J'}$, γ_J, γ'_J, and $\gamma'_{J'}$ being characterized by the relations below:

- $\mathrm{proj}_J \circ \gamma'_J = \mathrm{proj}_J \circ \gamma_J$ and $\mathrm{proj}_{J'} \circ \gamma'_{J'} = \mathrm{proj}_{J'} \circ \gamma_{J'}$, and
- $\mathrm{proj}_J \circ \gamma'_{J'} = \mathrm{cst}$ and $\mathrm{proj}_{J'} \circ \gamma'_J = \mathrm{cst}$.

The sequence of multi-instruction associated with γ' is $(\mu'_0, \ldots, \mu'_q)$ and we are done. □

Remark 7.3.5. Remark 1.5.8 also proves that observationally independent programs might not be model independent. Gathering the results obtained so far, we have proven the following chain of strict implications.

syntactic independence ⇒ model independence ⇒ observational independence

Relevancy of model independence goes beyond the theoretical aspect since it comes with a unique decomposition theorem for cubical regions together with a factorization algorithm Balabonski and Haucourt (2010). In addition Haucourt and Ninin (2014) proves that the Boolean algebra of n-dimensional cubical regions is the n-fold tensor product of the Boolean algebra of 1-dimensional cubical regions. Both results should be extendible to all isothetic regions and gathered in a more general formulation, which is the purpose of Chapter 9.

8

Categories of Components

In good cases (*e.g.* when X is the continuous model of a Paml program, see Definition 7.1.2) the homsets of the fundamental category of X are finitely generated, and even finite when X is a pospace. However $\overrightarrow{\pi_1}X$ still has infinitely many objects while, in some sense, all information about it would be known from a well chosen full subcategory with finitely many objects. If $\overrightarrow{\pi_1}X$ were a groupoid then its skeleton would be the disjoint union of the fundamental groups of its connected components, thus providing the expected reduction. Unfortunately, for any relevant X the only isomorphisms of $\overrightarrow{\pi_1}X$ are its identities, and therefore $\overrightarrow{\pi_1}X$ is its own skeleton. The category of components, introduced in Fajstrup et al. (2004), is intended to address this issue. The basic idea is to define a generic class of morphisms that strictly contains the collection of isomorphisms and shares enough of its properties. This lead to the notion of systems of weak isomorphisms. Note that we have adopted another terminology than the one used in Fajstrup et al. (2004): Yoneda morphisms and Yoneda systems are now called potential weak isomorphisms and systems of weak isomorphisms. A practical application of the categories of components can be found in Goubault and Haucourt (2005). The functoriality of the construction is broached in Goubault and Haucourt (2007). Goubault et al. (2010) changes the original approach to break the symmetry between past and future. The notion of components of a category is actually far from being completely understood except in the loop-free case – see Haucourt (2006). A concept that would relevantly apply to all small categories is almost surely a chimere – at least for now. Our aim is thus to find a class of categories within which a "reasonable" notion of component can be defined. Loop-free categories and one-way categories, which were respectively introduced in Haefliger (1992) and McLarty (2006) in mathematical contexts very far from ours, provide two examples. Localizations – see Borceux (1994a) and **generalized congruences** – see Bednarczyk et al. (1999), are everywhere in this chapter. The latter allow us to identify morphisms even when they do not share their extremities, and therefore also identify objects. When R is the left (resp. right) adjoint to an inclusion functor $\mathcal{A} \subseteq \mathcal{B}$, $R(x)$ is called the left (resp. right) **reflect** of x in $\mathcal{A}$. By extension the functor R is also called a **reflection** – see Borceux (1994a).

In Section 8.1 we describe a class of small categories in which the notion of components is fully understood, all the subsequent sections (but the last one) only deal with such categories. In Section 8.2, we formalize the idea of a class of morphisms that extends the class of isomorphisms but shares a great deal of its properties. The category of components $\overrightarrow{\pi_0}C$ of a one-way category C is defined in Section 8.3. By the way, an extensive description of it is provided. In particular it comes with a quotient functor

$q : C \to \overrightarrow{\pi_0}C$ whose right inverses are discussed in Section 8.4. The existence of such embeddings is guaranteed by the finiteness of the set of components. The general case is related to the Axiom of Choice. How the category of components construction behaves with respect to Cartesian product is studied in Section 8.5. In Section 8.6, we attempt to provide the category of components with a homotopical interpretation. Indeed both model categories (*cf.* Quillen (1967)) and components are based on the idea that the class of isomorphisms of certain categories could be extended soundly. Some conjectures about the components of regions are stated in Section 8.7. The overall idea is that the components of a region should be regions.

8.1 Loop-Free Categories vs One-Way Categories

The notion of one-way category slightly extends the notion of loop-free category. Most of the ideas implemented in this context come from Haucourt (2006).

Definition 8.1.1. A category C is said to be **one-way** when its endomorphisms are its identities. A one-way category C is said to be **loop-free** when $C[x, y] \neq \emptyset$ and $C[y, x] \neq \emptyset$ implies that $x = y$.

We denote by **Ow** (resp. **Lf**) the full subcategory of **Cat** (resp. **Ow**) whose objects are one-way (resp. loop-free).

Example 8.1.2. The fundamental category of a pospace is loop-free because any directed loop on a pospace is constant.

Definition 8.1.3 (Borceux (1994a), p.120)**.** A full subcategory $\mathcal{A}$ of $\mathcal{B}$ is said to be **reflective** when $\mathcal{A}$ contains any object of $\mathcal{B}$ which is isomorphic to some object of $\mathcal{A}$, and the inclusion functor $\mathcal{A} \hookrightarrow \mathcal{B}$ has a left adjoint. It is said to be **epireflective** when it is reflective and for all objects B of $\mathcal{B}$, the unit of the adjunction at B is an epimorphism.

Lemma 8.1.4. The category **Ow** (resp. **Lf**) is epireflective in **Cat** (resp. **Ow**).

Proof. Given a small category C consider the congruence $\sim$ that identifies any endomorphism $e \in C(x, x)$ with id_x. This construction induces a functor from **Cat** to **Ow** which is left adjoint to the inclusion **Ow** $\subseteq$ **Cat**. The collection of quotient functors $C \to C/\sim$, which are epimorphisms of **Cat**, indeed provides the unit of the adjunction.

Given a one-way category C consider the generalized congruence $\sim$ that identifies any isomorphism $f \in C(x, y)$ with id_x (and therefore also with id_y). This construction induces a functor from **Ow** to **Lf** which is left adjoint to the inclusion **Lf** $\subseteq$ **Ow**. The collection of quotient functors $C \to C/\sim$, which are epimorphisms of **Ow**, indeed provides the unit of the adjunction. □

The skeleton of a category can be defined as any of its full subcategories whose collection of objects meets every isomorphism class exactly once. Such a full subcategory is a representative of the skeleton.

Lemma 8.1.5. A category is one-way iff its skeleton if loop-free. Moreover the reflect of a one-way category in **Lf** is its skeleton.

Proof. Let C be a one-way category and let f, g be morphisms of the skeleton of C such that $\partial^- f = \partial^+ g$ and $\partial^+ f = \partial^- g$. Then f is an isomorphism: its inverse is necessarily

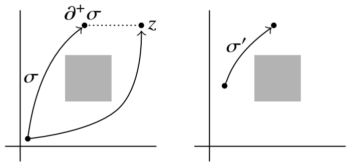


Figure 8.1: Potential weak isomorphism

g because C is one-way. Since the skeleton meets every isomorphism class exactly once we have $\partial^- f = \partial^+ f$ hence f is an identity. Conversely every homset $C(x, x)$ is in bijection with some homset $\mathcal{S}(y, y)$ of the skeleton of C, which is loop-free.

Suppose that C is one-way and let $\mathcal{S}$ be a representative of its skeleton. Then denote by $\phi(x)$, for every object $x \in C$, the distinguished object of the isomorphism class of x. Since C is one-way, there is a unique morphism $\tau_x : x \longrightarrow \phi(x)$. In particular τ_x is an isomorphism. Then one can extend ϕ to a functor from C to $\mathcal{S}$ sending $f \in C(x, y)$ to $\tau_y \circ f \circ \tau_x^{-1} \in \mathcal{S}(\phi(x), \phi(y))$. Then one checks that ϕ sends any isomorphism to an identity, and that it is universal among those functors enjoying this property. □

Corollary 8.1.6. A category is one-way iff it is equivalent to its loop-free reflect.

The construction we are about to describe is actually a generalization of the previous one.

8.2 Systems of Weak Isomorphisms

Given a small category C, we define a class of morphisms that contains the class of isomorphisms and that enjoys many of its properties. For any morphism σ and any object z, we define the σ, z-precomposition as the mapping

$$\gamma \in C(\partial^+\sigma, z) \mapsto \gamma \circ \sigma \in C(\partial^-\sigma, z)$$

and dually the z, σ-post-composition by

$$\delta \in C(z, \partial^-\sigma) \mapsto \sigma \circ \delta \in C(z, \partial^+\sigma)$$

Note that $C(\partial^+\sigma, z)$ and $C(z, \partial^-\sigma)$ may be empty. Also remark that σ is an isomorphism iff for all z both σ, z-precomposition and z, σ-post-composition are bijective. In order to capture a wider class, the latter condition has to be weakened.

Definition 8.2.1. A morphism σ is said to preserve the **future cones** (resp. **past cones**) when for all z, if $C(\partial^+\sigma, z) \neq \emptyset$ (resp. $C(z, \partial^-\sigma) \neq \emptyset$) then the precomposition (resp. post-composition) is bijective. The morphism σ is called a **potential weak isomorphism** when it preserves both the future and the past cones. Potential weak isomorphisms readily compose.

Example 8.2.2. Assume that we are working in the fundamental category of the isothetic region depicted on Figure 8.1. Due to the lower dipath, the σ, z-precomposition is not bijective. On the contrary, σ' is a potential weak isomorphism.

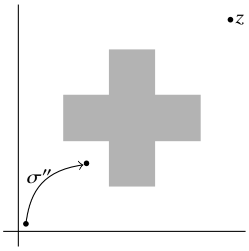


Figure 8.2: An unwanted potential weak isomorphism

Example 8.2.3. The collection of potential weak isomorphisms might contain unwanted members. On Figure 8.2, the morphism σ'' is indeed a potential weak isomorphism though there exists a morphism from $\partial^- \sigma''$ to z but none from $\partial^+ \sigma''$ to z.

A remarkable feature of the collection of isomorphisms of a category is that it is stable under pushouts. That is if σ is an isomorphism and γ is any morphism such that $\partial^- \sigma = \partial^- \gamma$ then there exists a pushout of σ along γ and all of them are still isomorphisms. Dually, the collection of isomorphisms is stable under pullback. Hence

Definition 8.2.4. A family Σ is said to be a **system of weak isomorphisms** when
i) $\{$ isomorphisms $\} \subseteq \Sigma \subseteq \{$ potential weak isomorphisms $\}$
ii) Σ is stable under pushouts and pullbacks

Remark 8.2.5. Let σ be an element of some system of weak isomorphisms Σ, and ϕ be an isomorphism such that $\partial^- \sigma = \partial^+ \phi$. The pushout of σ along ϕ^{-1}, namely $\sigma \circ \phi$, thus belongs to Σ. We have proved that Σ is stable under pre-composition by an isomorphism. Dually, we prove that it is stable under post-composition by an isomorphism.

We denote by $< \Sigma >$ the closure under composition of any collection Σ of morphisms of C.

Lemma 8.2.6. If Σ is a system of weak isomorphisms then so is $< \Sigma >$.

Proof. Potential weak isomorphisms compose and two pullbacks (resp. pushouts) put side by side give rise to another pullback (resp. pushout) – (Borceux, 1994a, Prop.2.5.9 p.54). □

Denote the collections of systems of weak isomorphisms over C by $\mathrm{SWI}(C)$, and let $\mathrm{SWIC}(C)$ be the subcollection of $\mathrm{SWI}(C)$ whose elements are closed under composition. We denote the corresponding inclusion by i and the mapping that returns the closure under composition by r. They provide a Galois connection i.e. $r \circ i = \mathrm{id}$ and $\mathrm{id} \subseteq i \circ r$

$$\begin{tikzcd}\mathrm{SWIC}(C) \arrow[r, shift left, "i"] & \mathrm{SWI}(C) \arrow[l, shift left, "r"]\end{tikzcd}$$

Remark 8.2.7. A **right calculus of fractions** of C is a collection Σ of morphisms of C satisfying the following properties (Borceux, 1994a, Definition 5.2.3, p.183):

– $\{$ identities $\} \subseteq \Sigma$,

– Σ is stable under composition,

- for all morphisms $f : x \longrightarrow z$ of C and all $\sigma : y \longrightarrow z$ with $\sigma \in \Sigma$, there exist a morphism $f' : z' \longrightarrow y$ of C and $\sigma' : z' \longrightarrow x$ in Σ such that $\sigma \circ f' = f \circ \sigma'$, and
- for all morphisms $f, g : x \longrightarrow y$ of C and all $\sigma : z \longrightarrow x$ in Σ such that $\sigma \circ f = \sigma \circ g$, there exists $\sigma' : y \longrightarrow z'$ such that $f \circ \sigma' = g \circ \sigma'$.

Every $\Sigma \in \mathrm{SWIC}(C)$ readily satisfies the first three properties. Then remark that if $\sigma \circ f = \sigma \circ g$, then $f = g$ because σ is a potential weak isomorphism. Therefore $\sigma' = \mathrm{id}_y$ is an element of Σ that satisfies $f \circ \sigma' = g \circ \sigma'$. Dually, Σ is also a **left calculus of fractions**. This remark will be of importance in Section 8.3.

From now on C is assumed to be one-way.

Definition 8.2.8. A collection Σ of morphisms is said to be **pure** when both morphisms γ and δ belong to Σ when their composite does.

The following Lemma is of prime importance in the sequel.

Lemma 8.2.9. Any system of weak isomorphisms of a one-way category is pure.

Proof. Consider $\sigma \in \Sigma$ and two morphisms δ and γ such that $\sigma = \gamma \circ \delta$. As Σ is stable under pushout, we have $\sigma' \in \Sigma$ and a morphism δ' of C which form a pushout square. We also have a unique morphism ξ making the following diagram commute. Since C is one-way both morphisms δ' and ξ are isomorphisms. In particular $\gamma = \xi \circ \sigma'$ belongs to Σ by Remark 8.2.5. We prove the same way, referring to the stability under pullback, that δ belongs to Σ.

$$\begin{tikzcd}
 & {} & \\
 & {} \arrow[u, "\xi"] & \\
{} \arrow[ur, dotted, "\delta'"] \arrow[uur, bend left, "\mathrm{id}"] & \scriptstyle\text{push out} & {} \arrow[ul, dotted, "\sigma'"'] \arrow[uul, bend right, "\gamma"'] \\
 & {} \arrow[ul, "\sigma"] \arrow[ur, "\delta"'] &
\end{tikzcd}$$

□

Lemma 8.2.10. The inverse image (resp. the direct image) of a system of weak isomorphisms by an equivalence of categories is a system of weak isomorphisms. If the system is stable under composition, then so is its inverse (resp. direct) image.

Proof. As an equivalence of category, the functor $E : C \longrightarrow \mathcal{D}$ is such that
1) all the homset mappings $\gamma \in C(x, y) \mapsto E(\gamma) \in \mathcal{D}(Ex, Ey)$ are bijections,
2) any object of $\mathcal{D}$ is isomorphic to some $E(x)$ with x object of C,
3) the pushout squares and the pullback squares are preserved by E.
The remaining details are routine verification. □

A **locale** is a **complete lattice** whose binary meet distributes over arbitrary join. Intuitively it mimics the properties of the lattice of open subsets of a topological space – see (Borceux, 1994c, Chap.1) or (Pedicchio et al., 2003, Chap.2).

Lemma 8.2.11. The collection $\mathrm{SWI}(C)$ is stable under union and intersection, and $\mathrm{SWIC}(C)$ forms a locale. They share their least and greatest elements.

Proof. The collection SWI(C) is obviously stable under union. Suppose that σ belongs to the intersection of a nonempty family Σ_i of elements of SWI(C). Let γ be a morphism with $\partial^-\gamma = \partial^-\sigma$. Given an index i we let σ' be a representative of the pushout of σ along γ. Then σ' belongs to Σ_i. By Lemma 8.2.6 the least upper bound of a *nonempty* family of elements of SWIC(C) is the closure under composition of its union. As a consequence the union of all systems of weak isomorphisms of C is stable under composition. Let Σ' belong to SWIC(C) and Σ_i be a nonempty family of elements of SWIC(C). We would like to prove that

$$\bigvee_i (\Sigma' \cap \Sigma_i) \;=\; \Sigma' \cap \bigvee_i \Sigma_i$$

From general facts about posets the left-hand term is included in the right-hand one as soon as both exist. Conversely an element of the right hand term can be written as $\sigma = \sigma_n \circ \cdots \circ \sigma_1$ with $\sigma_k \in \Sigma_{i_k}$. By Lemma 8.2.9 each σ_k belongs to Σ' which is closed under composition. □

In particular the greatest system of weak isomorphisms of C, which is intended to extend the class of isomorphisms, can also be obtained as the following decreasing intersection

$$\bigcap_{n\in\mathbb{N}} \Phi^n(\{\text{potential weak isomorphisms of } C\})$$

with $\Phi(S)$ defined as the collection of morphisms of $\sigma \in S$ such that for all morphisms f:

– if $\partial^-\sigma = \partial^- f$ then the pushout of σ along f exists and belongs to S, and
– if $\partial^+\sigma = \partial^+ f$ then the pullback of σ along f exists and belongs to S.

Lemma 8.2.12. The greatest system of weak isomorphisms of C is stable under the action of the group of functorial permutations of the category C (*i.e.* the functors from C to itself which admits an inverse)).

Proof. Let $A : C \to C$ be an automorphism. Then note that A induces a permutation of the set of potential weak isomorphisms of C and that the mapping on morphisms induced by A commutes with Φ. □

Remark 8.2.13. Lemma 8.2.12 may fail for systems of weak isomorphisms which are not maximum. Consider for example the totally ordered real line $(\mathbb{R}, \leqslant)$ as a loop-free category. Its greatest system of weak equivalences contains all the morphisms (*i.e.* the 2-tuple (x, y) with $x \leqslant y$). The partition

$$\mathbb{R} \quad = \quad]-\infty, 0[\;\cup\; [0, +\infty[$$

induces the following system of weak isomorphisms.

$$\Sigma \quad = \quad \big\{(x, y) \,\big|\, x \leqslant y;\ y < 0 \text{ or } 0 \leqslant x\big\}$$

Then any nonidentity translation induces an automorphism of $(\mathbb{R}, \leqslant)$ which does not preserve Σ.

From now on all the systems of weak isomorphisms we consider are supposed to be closed under composition.

Definition 8.2.14. The elements of the greatest system of weak isomorphisms are called the **weak isomorphisms** of the category.

$$\begin{tikzcd}
y \arrow[r] & {} \\
{} \arrow[u] \arrow[r] & x \arrow[u] \arrow[ul, dashed]
\end{tikzcd}$$

Figure 8.3: Filling square property

Example 8.2.15. We go back over Example 4.2.5. Since the fundamental category of any pospace is loop-free (*cf.* Example 8.1.2), we can consider the category of pairs (X, Σ) where X is a pospace and Σ is a system of weak isomorphisms of $\overrightarrow{\pi_1}X$. The morphisms from (X, Σ) to (X', Σ') are the pospace morphisms f from X to X' such that $\overrightarrow{\pi_1}f(\Sigma) \subseteq \Sigma'$. Then define $A = \mathbb{R}_- \setminus \{0\} \times \mathbb{R}$, $B = \mathbb{R}^2 \setminus (0, 0)$, and $C = \mathbb{R}^2$. The greatest systems of weak isomorphisms of A and C contain all the morphisms, while the components corresponding to the greatest system of weak isomorphisms of B are

$$(\mathbb{R}_+ \setminus \{0\})^2,\ (\mathbb{R}_- \setminus \{0\})^2,\ \mathbb{R}_+ \times \mathbb{R}_- \setminus (0,0),\ \text{and } \mathbb{R}_- \times \mathbb{R}_+ \setminus (0,0).$$

According to the fourth axiom of Definition 4.2.2, A is a subspace of C provided that both of them are equipped with their greatest systems of weak isomorphisms. However the subspace structure induced on A by B is made of the two components $(\mathbb{R}_- \setminus \{0\})^2$ and $\mathbb{R}_- \setminus \{0\} \times \mathbb{R}_+$, therefore it is not a subspace of C (although it is a subobject of it), see Example 4.2.5.

Before embarking in the next section, which defines the category of components, we prove that, under an additional hypothesis, the axioms of Definition 8.2.14 can be relaxed.

Definition 8.2.16. A category is said to be **square filling** when for all commutative squares which are both pushout and pullback, if there is at least one morphism from x to y then there exists a morphism from x to y that makes both triangles on Figure 8.3 commute.

Proposition 8.2.17. In a filling square one-way category, any collection of morphisms that is stable under pushout and pullback is actually a system of weak isomorphisms.

Proof. Let Σ be such a collection of such a category C. Given $\sigma \in \Sigma(x, y)$ and $\gamma \in C(y, z)$ we have, by hypothesis, a pushout square formed by the morphisms σ, γ, $\gamma \circ \sigma$, and $\sigma' \in \Sigma$. In particular there is a unique morphism ξ such that $\xi \circ \sigma' = \text{id}$ and $\gamma = \xi \circ \gamma'$. Then ξ is an isomorphism because C is one-way, and the outer shape of the following diagram is actually a pushout.

$$\begin{tikzcd}
& & {} \\
{} \arrow[r, "\sigma'"] \arrow[urr, bend left, "\text{id}"] & {} \arrow[ur, "\xi"] & \\
{} \arrow[u, "\gamma\circ\sigma"] \arrow[r, "\sigma"'] & {} \arrow[u, "\gamma'"'] \arrow[uur, bend right, "\gamma"'] &
\end{tikzcd}$$

If γ'' is a morphism such that $\sigma \circ \gamma = \sigma \circ \gamma''$ then we have a pushout square formed by σ, γ'', $\gamma \circ \sigma$, and id. Then $\gamma = \gamma''$ since the only endomorphisms of C are its identities. Therefore σ is an epimorphism. By duality it is also a monomorphism.

Now let $\delta \in C(x, z)$ with $C(y, z) \neq \emptyset$. By hypothesis on Σ we have a pushout square formed by σ, δ, δ', and $\sigma' \in \Sigma$; and then a pullback square formed by σ', δ', δ'', and $\sigma'' \in \Sigma$. Therefore we have a morphism ζ such that $\delta = \delta'' \circ \zeta$ and $\sigma = \sigma'' \circ \zeta$. Then let u and v be such that $u \circ \delta'' = v \circ \sigma''$. By precomposition with ζ we obtain $u \circ \delta = v \circ \sigma$ so there is a unique ζ' such that $\zeta' \circ \sigma' = u$ and $\zeta' \circ \delta' = v$. Hence the pullback square is also a pushout. By the filling square property there is thus a morphism γ such that $\delta'' = \gamma \circ \sigma''$ from which we deduce that $\delta = \gamma \circ \sigma$.

$$\begin{tikzcd}
 & & & {} \\
 & z \arrow[r, "\sigma'"] \arrow[urr, bend left, "u"] & {} \arrow[ur, "\zeta'"] & \\
 & {} \arrow[u, "\delta''"] \arrow[r, "\sigma''"'] & y \arrow[u, "\delta'"'] \arrow[ul, "\gamma"] \arrow[uur, bend right, "v"'] & \\
x \arrow[ur, "\zeta"] \arrow[uur, bend left, "\delta"] \arrow[urr, bend right, "\sigma"'] & & &
\end{tikzcd}$$

□

Proposition 8.2.17 might have a concrete application. Indeed we speculate that the fundamental category of any isothetic region without directed loops actually satisfies a stronger property: for all commutative squares with at least one dipath from x to y – see Figure 8.3, there exists a morphism that makes both triangles of Figure 8.3 commute. Hence the algorithm that computes the category of components of a cubical region can actually skip the initialization phase which consists of finding all the potential weak isomorphisms.

Remark 8.2.18. The fundamental category of the directed circle (*cf.* Remark 5.2.17) does not satisfy the filling square property.

8.3 Categories of Components

Given a collection of morphisms Σ the **quotient** functor Q_Σ is defined by the following universal property: for all categories $\mathcal{D}$ and all functors $F : C \longrightarrow \mathcal{D}$ such that

$$F(\Sigma) \subseteq \{\text{identities of } \mathcal{D}\}$$

there exists a unique functor G such that $F = G \circ Q_\Sigma$. The existence of Q_Σ is a consequence of a general result explained in Bednarczyk et al. (1999). The **localization** functor I_Σ is defined similarly with the condition

$$F(\Sigma) \subseteq \{\text{isomorphisms of } \mathcal{D}\} .$$

Remark 8.3.1. For any collection Σ of morphisms of C we have $Q_\Sigma = Q_{<\Sigma>}$ and $I_\Sigma = I_{<\Sigma>}$. It immediately derives from the fact that functors preserve identities, isomorphisms, and composition.

The existence of I_Σ relies on a general construction which is detailed in (Borceux, 1994a, Chap.5). While the former construction is rather restricted the latter one is widely spread in model category and homological algebra. We describe Q_Σ and I_Σ

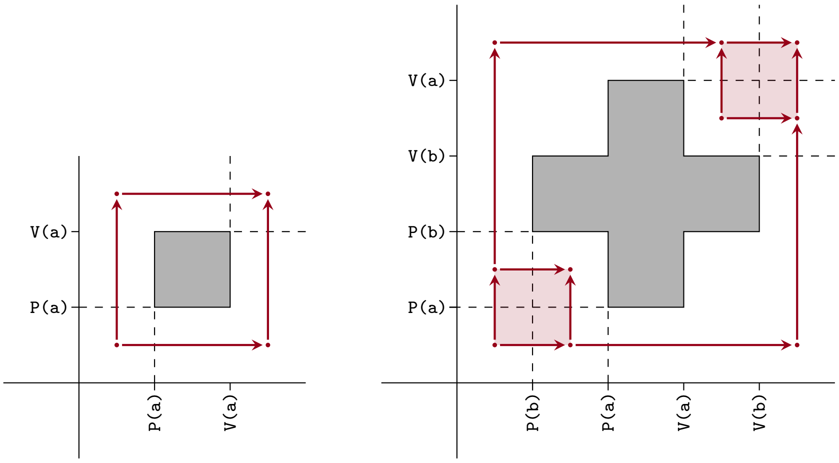


Figure 8.4: The categories of components of the square and the Swiss Cross

when Σ is a system of weak isomorphisms of a one-way category $\mathcal{C}$ that is closed under composition. We know from Remark 8.2.7 that Σ is both a right and a left calculus of fractions. Then we prove that the codomains of Q_Σ and I_Σ, respectively the **quotient category** $\mathcal{C}/\Sigma$ and the **category of fractions** $\mathcal{C}[\Sigma^{-1}]$, are equivalent.

Definition 8.3.2. The **category of components** of $\mathcal{C}$, denoted by $\overrightarrow{\pi_0}\mathcal{C}$, is the quotient of $\mathcal{C}$ by its greatest system of weak isomorphisms (*cf.* Lemma 8.2.11).

Example 8.3.3. The fundamental categories of the continuous models depicted on Figure 8.1 and Figure 8.2 are loop-free, their categories of components are shown (in red) on Figure 8.4. The red arrows generate the category while the light red filled squares indicate commutative diagrams. These categories of components can be seen as precubical sets, in particular the category of components of the square is isomorphic to $\partial\square_2^+$ (*i.e.* the boundary of the standard 2-cube $\square_2^+$ (*cf.* Definition 2.4.8)).

Example 8.3.4. The category of components of the continuous model on Figure 7.1 is more intriguing. It is indeed isomorphic to the disjoint union of 3 copies of the hollow square. The correspondence between components is shown on Figure 8.6. Two components appearing on the same line are related by an isomorphism between the categories of components.

Example 8.3.5. The components of the hollow cube (*i.e.* the boundary of $[0,1]^3$ as a subpospace of $\mathbb{R}^3$) are the subspaces of the form $A \times B \times C$ with $A, B, C \in \big\{\{0\},\{1\},]0,1[\big\}$ and at least one of them not $]0,1[$ (*i.e.* the 8 vertices, the 12 edges, and the 6 faces). The category of components is isomorphic to the fundamental category of the 3-fold tensor product of the graph $0 \longrightarrow 1 \longrightarrow 2$ from which the middle point $(1,1,1)$ has been removed (*cf.* Definition 2.4.8). It is also the category of components of the "floating cube" – see Figure 7.5.

Generally speaking the fundamental category of a precubical set is loop-free iff so is the fundamental category of "its" directed realization. When it is the case, one may

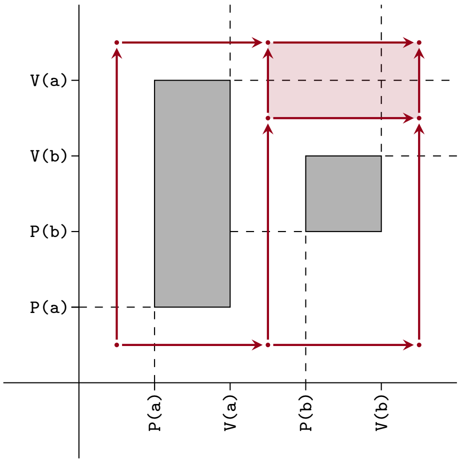


Figure 8.5: The category of components of a square and a rectangle

ask whether

$$\overrightarrow{\pi_0}\overrightarrow{\pi_1}K \cong \overrightarrow{\pi_0}\overrightarrow{\pi_1}\rceil K \rfloor$$

Taking $K = \partial\square_3^+$ actually provides a counterexample. Indeed $\overrightarrow{\pi_1}K \cong \{0 < 1\}^3$ therefore it has a single components, while $\rceil K \rfloor$ is the hollow cube.

Example 8.3.6. The category of components of the tetrahemihexacron (or "3D Swiss Cross" see Figure 8.7) is freely generated by the graph on Figure 8.7.

Lemma 8.3.7. Any homset containing a potential weak isomorphism is a singleton.

Proof. If $\sigma \in C(x, y)$ is a weak isomorphism then the x, σ-post-composition is a bijection from $C[x, x]$ to $C[x, y]$, and the former is a singleton since C is one-way. □

As a consequence of Lemma 8.3.7 any element of a system of weak isomorphisms is entirely defined by its extremities. In particular any arrow labelled by a system of weak isomorphisms Σ is the unique element of its homset and it belongs to Σ.

Definition 8.3.8. Two objects x and y are **Σ-connected** when there exists a finite sequence $x_0, \ldots, x_n$ such that $x_0 = x$, $x_n = y$, and $\Sigma[x_i, x_{i-1}] \neq \emptyset$ or $\Sigma[x_{i-1}, x_i] \neq \emptyset$ for all $i \in \{1, \ldots, n\}$. We can suppose that $x_i \neq x_j$ for $i \neq j$ and also that $\Sigma[x_i, x_{i-1}] = \emptyset$ or $\Sigma[x_{i-1}, x_i] = \emptyset$ because C is one-way and Σ is stable under composition.

Therefore by Lemma 8.3.7 the sequence $x_0, \ldots, x_n$ fully characterizes a zigzag of elements of Σ. The **Σ-connectedness** defines an equivalence relation over the objects of C whose classes are called the **Σ-components** of C.

Lemma 8.3.9. The following are equivalent:

1) The objects x and y are Σ-connected.

2) There exists an object z such that $\Sigma[z, x] \times \Sigma[z, y]$ is not empty.

3) There exists an object z such that $\Sigma[x, z] \times \Sigma[y, z]$ is not empty.

Proof. Statements 2) and 3) are equivalent because Σ is stable under pushouts and pullbacks. Now if there exists a Σ-zigzag of length greater or equal than 3, then a strictly shorter one can be found applying the fact that Σ is stable under pushout and composition. □

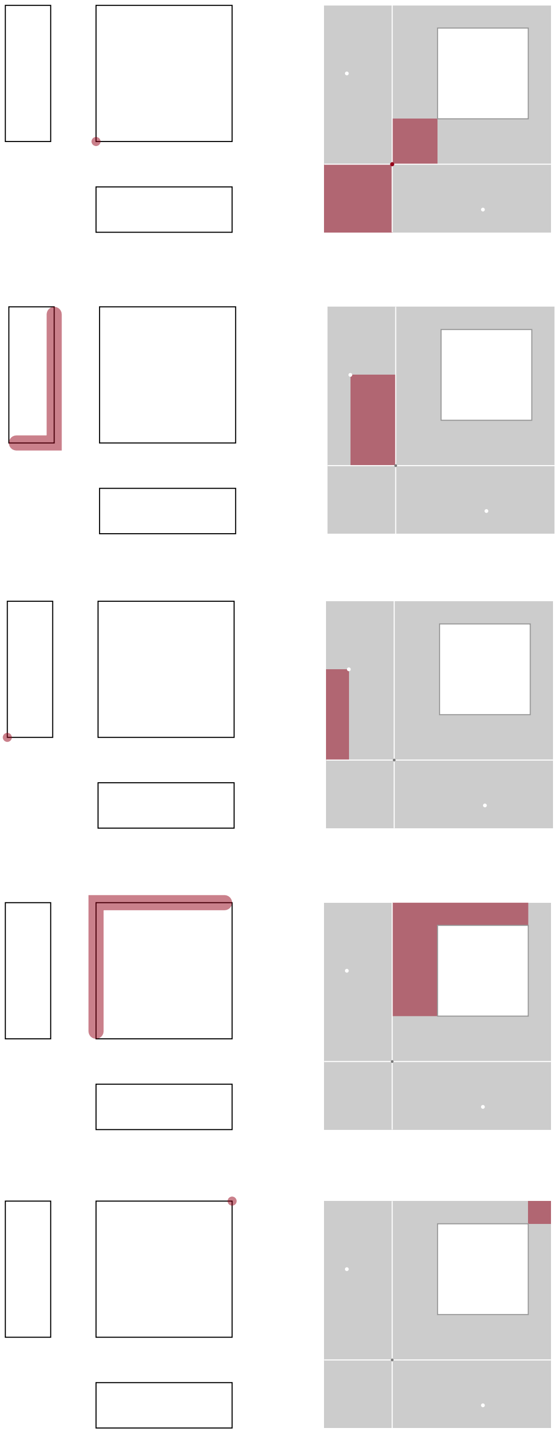


Figure 8.6: Corresponding components.

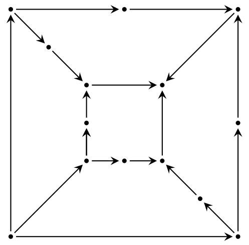


Figure 8.7: Components of the tetrahemihexacron a.k.a. 3D Swiss Cross

$$\begin{tikzcd}
& u & \\
x \arrow[ur] & & y \arrow[ul] \\
& d \arrow[ul] \arrow[ur] \arrow[uu, phantom, very near start, "\wedge"] \arrow[uu, phantom, very near end, "\vee"] &
\end{tikzcd}$$

Diagram 1

$$\begin{tikzcd}
x & & y \\
& d \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Diagram 2

$$\begin{tikzcd}
& u & \\
x \arrow[ur] & & y \arrow[ul]
\end{tikzcd}$$

Diagram 3

Figure 8.8: Structure of the Σ-components

A preordered set in which any pair of elements has both a least upper bound and a greatest lower bound is called a **prelattice**. A lattice can thus be seen as an antisymmetric prelattice. According to that definition, a lattice may not have a least (resp. greatest) element. This fact will be of importance in Subsection 9.5, moreover it is adapted to the next statement.

Theorem 8.3.10 (Structure of a Σ-component)**.** Let K be a Σ-component of C, and $\mathcal{K}$ be the full subcategory of C whose objects are the elements of K. The following properties are satisfied :

1. The category $\mathcal{K}$ is isomorphic to the preorder $(K, \preccurlyeq)$ where $x \preccurlyeq y$ stands for $C[x, y] \neq \emptyset$. In particular, every diagram in $\mathcal{K}$ commutes.
2. The preordered set $(K, \preccurlyeq)$ is a prelattice.
3. If d and u are respectively a greatest lower bound and a least upper bound of the pair $\{x, y\}$, then Diagram 1 on Figure 8.8 is both a pullback and a pushout in C, and all the arrows appearing on the diagram belong to Σ.
4. $C = \mathcal{K}$ iff C is a prelattice, and Σ is the greatest system of weak isomorphisms of C.

Proof. Let $\alpha \in \mathcal{K}(x, y)$. Since the objects x and y lie in the same Σ-component, there are, by Lemma 8.3.9, four morphisms $\sigma_1, \sigma_2, \sigma_3$ and σ_4 in Σ which form a commutative square as in Diagram 1 on Figure 8.9. The morphism $\sigma_1 \circ \sigma_3$ belongs to Σ which is stable under composition. By Lemma 8.3.7 it follows that $\sigma_2 \circ \alpha \circ \sigma_3 = \sigma_1 \circ \sigma_3$. Then $\alpha \in \Sigma$ by Lemma 8.2.9, and $\mathcal{K}(x, y)$ is a singleton by Lemma 8.3.7. On the way we have proved that any morphism of C between two objects of the same Σ-component is in Σ. Let x and y be two elements K. Lemma 8.3.9 provides Diagram 2 on Figure 8.9 which admits a pullback with $\sigma_1', \sigma_2' \in \Sigma$, as shown on Diagram 3 on Figure 8.9. The object d clearly belongs to K. If d' is a lower bound of $\{x, y\}$ then $C[d', x]$ and $C[d', y]$ are two singletons whose respective elements γ and δ belong to Σ. So Diagram 4 on Figure 8.9 commutes by Lemma 8.3.7. The universal property of pullbacks implies that

$$\begin{tikzcd}
& {} & \\
x \arrow[ur, "\sigma_1"] \arrow[rr, "\alpha"] & & y \arrow[ul, "\sigma_2"'] \\
& {} \arrow[ul, "\sigma_3"] \arrow[ur, "\sigma_4"'] &
\end{tikzcd}$$

Diagram 1

$$\begin{tikzcd}
& u & \\
x \arrow[ur, "\sigma_1"] & & y \arrow[ul, "\sigma_2"']
\end{tikzcd}$$

Diagram 2

$$\begin{tikzcd}
& u & \\
x \arrow[ur, "\sigma_1"] & \text{pullback} & y \arrow[ul, "\sigma_2"'] \\
& d \arrow[ul, "\sigma'_2"] \arrow[ur, "\sigma'_1"'] &
\end{tikzcd}$$

Diagram 3

$$\begin{tikzcd}
& u & \\
x \arrow[ur, "\sigma_1"] & & y \arrow[ul, "\sigma_2"'] \\
& d' \arrow[ul, "\gamma"] \arrow[ur, "\delta"'] &
\end{tikzcd}$$

Diagram 4

Figure 8.9: Illustrating the proof of Theorem 8.3.10

$$\begin{tikzcd}
& & x \arrow[r, "\delta"] & y \arrow[dr, "\Sigma"] \arrow[drr, "\Sigma"] & & \\
d \arrow[urr, "\Sigma"] \arrow[drr, "\Sigma"'] & x \wedge x' \arrow[l, "\cong"'] \arrow[ur, "\Sigma"'] \arrow[dr, "\Sigma"] & & & y \vee y' \arrow[r, "\cong"] & u \\
& & x' \arrow[r, "\delta'"'] & y' \arrow[ur, "\Sigma"] \arrow[urr, "\Sigma"'] & &
\end{tikzcd}$$

Figure 8.10: Soundness of the equivalence of morphisms

$\mathcal{K}[d', d] \neq \emptyset$ i.e. $d' \preccurlyeq d$. We prove analogously the existence of the least upper bound of $\{x, y\}$. The third and fourth assertion follow. □

We provide a simple description of the quotient category C/Σ. Its objects are the Σ-connected components of C, in particular the one containing an object x is denoted by $[x]$. Then given $\delta \in C(x, y)$ and $\delta' \in C(x', y')$ with $x \sim x'$ and $y \sim y'$ we write $\delta \sim \delta'$ when the inner hexagon on Figure 8.10 commutes. There is however a slight ambiguity here since the greatest lower bound and least upper bound given by Theorem 8.3.10 are only defined up to isomorphisms. Then let u and d be other representatives of $x \wedge x'$ and $y \vee y'$. On Figure 8.10 the arrows labelled with Σ belongs to it while the ones labelled with $\cong$ are isomorphisms. Then it follows by Lemma 8.3.7 that the four triangles on Figure 8.10 commute, so an easy diagram chase proves that the outer hexagon commutes iff the inner one does. Therefore we will write without ambiguity $x \vee y$ and $x \wedge y$ to denote a greatest lower bound or a least upper bound of x and y.

Lemma 8.3.11. The relation $\sim$ is an equivalence.

Proof. Transitivity is given by Lemma 8.3.9, Lemma 8.3.7 and the fact that Σ is stable under composition. □

We denote by $[\gamma]$ the $\sim$-equivalence class of γ.

Lemma 8.3.12. Let γ, γ', δ and δ' be respectively picked from $C[y, z]$, $C[y', z']$, $C[x, y]$ and $C[x', y']$. If $\gamma \sim \gamma'$ and $\delta \sim \delta'$ then $\gamma \circ \delta \sim \gamma' \circ \delta'$.

Proof. Since Diagrams 1 and 2 on Figure 8.11 commute and the central square on Figure 8.12 is both pushout and a pullback (*cf.* Theorem 8.3.10), there exist a unique morphism δ'' in $C[x \wedge x', y \wedge y']$ and a unique morphism γ'' in $C[y \vee y', z \vee z']$ making the diagram on Figure 8.12 commute. □

$$\begin{tikzcd}
x \arrow[r, "\delta"] & y \arrow[r, "\Sigma"] & y \vee y' \\
x \wedge x' \arrow[u, "\Sigma"] \arrow[r, "\Sigma"'] & x' \arrow[r, "\delta'"'] & y' \arrow[u, "\Sigma"']
\end{tikzcd}$$

Diagram 1

$$\begin{tikzcd}
y \arrow[r, "\gamma"] & z \arrow[r, "\Sigma"] & z \vee z' \\
y \wedge y' \arrow[u, "\Sigma"] \arrow[r, "\Sigma"'] & y' \arrow[r, "\gamma'"'] & z' \arrow[u, "\Sigma"']
\end{tikzcd}$$

Diagram 2

$$\begin{tikzcd}
x \arrow[r, "\gamma \circ \delta"] & z \arrow[r, "\Sigma"] & z \vee z' \\
x \wedge x' \arrow[u, "\Sigma"] \arrow[r, "\Sigma"'] & x' \arrow[r, "\gamma' \circ \delta'"'] & z' \arrow[u, "\Sigma"']
\end{tikzcd}$$

Diagram 3

Figure 8.11: Equivalent morphisms

$$\begin{tikzcd}
 & x \arrow[rr, "\delta"] & & y \arrow[rr, "\gamma"] \arrow[dr, "\Sigma"] & & z \arrow[dr, "\Sigma"] & \\
x \wedge x' \arrow[ur, "\Sigma"] \arrow[dr, "\Sigma"'] \arrow[rr, "\delta''"] & & y \wedge y' \arrow[ur, "\Sigma"] \arrow[dr, "\Sigma"'] & & y \vee y' \arrow[rr, "\gamma''"] & & z \vee z' \\
 & x' \arrow[rr, "\delta'"'] & & y' \arrow[rr, "\gamma'"'] \arrow[ur, "\Sigma"'] & & z' \arrow[ur, "\Sigma"'] &
\end{tikzcd}$$

Figure 8.12: Equivalence fits with composition

We now describe a category $\overline{C}$ whose objects and morphisms are the Σ-components and the $\sim$-equivalence classes of morphisms of C. Given a morphism γ of C the source and the target of $[\gamma]$ are defined as $[\partial^-(\gamma)]$ and $[\partial^+(\gamma)]$. Indeed none of these equivalence classes depend on the choice of the representative of $[\gamma]$. Given another morphism δ of C such that $[\partial^-(\gamma)] = [\partial^+(\delta)]$ there exist two morphisms γ' and δ' of C such that $[\gamma] = [\gamma']$, $[\delta] = [\delta']$ and $\partial^-(\gamma') = \partial^+(\delta')$: it suffices to invoke the stability of Σ under pushout as suggested by Figure 8.13. Suppose that γ'' and δ'' are two other such morphisms, then we have $[\gamma'] = [\gamma'']$ and $[\delta'] = [\delta'']$. Applying Lemma 8.3.12 we have $[\gamma' \circ \delta'] = [\gamma'' \circ \delta'']$. Then we can define $[\gamma] \circ [\delta]$ as $[\gamma \circ \delta]$ without ambiguity, the identities are given by the $\sim$-equivalence classes of the identities of C.

Lemma 8.3.13. For any morphism δ of C, t.f.a.e.

1) $\delta \in \Sigma$
2) $[\delta] \subseteq \Sigma$
3) $[\delta]$ is an identity of $\overline{C}$

Proof. By Lemma 8.2.9 the collection Σ is pure therefore the morphism δ on Diagram 1 of Figure 8.11 belongs to Σ is and only if so does δ', statements 1) and 2) are thus equivalent. It is easy to see that $[\sigma] = [\mathrm{id}_{\partial^-(\sigma)}] = [\mathrm{id}_{\partial^+(\sigma)}]$ for all σ in Σ, so 1) implies 3). Now if $[\delta]$ is an identity of $\overline{C}$, then $\delta \sim \mathrm{id}_x$ for some object x of C. Since id_x is an element of Σ so does δ (see the argument at the beginning of this proof). □

The category $\overline{C}$ is thus constructed and we define a functor $\overline{Q}$ from C to $\overline{C}$, surjective

$$\begin{tikzcd}
 & {} \arrow[r, "\Sigma"] & {} \arrow[r, "\gamma'"] & {} \\
{} \arrow[ur, "\delta"] \arrow[urr, "\delta'"'] & & {} \arrow[u, "\Sigma"] \arrow[r, "\gamma"'] \arrow[ur, phantom, "\llcorner", very near end] & {} \arrow[u, "\Sigma"']
\end{tikzcd}$$

Figure 8.13: Composition in the quotient category

on objects and morphisms, by setting $\overline{Q}(x) := [x]$ for each object x of C and $\overline{Q}(\gamma) := [\gamma]$ for each morphism γ of C. We conclude :

Proposition 8.3.14. The category $\overline{C}$ and the functor $\overline{Q}$ are the quotient category C/Σ and the quotient functor Q_Σ.

Proof. Let $F : C \to \mathcal{D}$ be a functor such that $F(\Sigma) \subseteq \{\text{identities of } \mathcal{D}\}$. For every object x and every morphism γ of C, we set $G([x]) := F(x)$ and $G([\delta]) := F(\delta)$. These definitions are sound since $F(x) = F(x')$ as soon as x and x' are Σ-connected and $F(\delta) = F(\delta')$ when Diagram 1 on Figure 8.11 commutes. One easily checks that G is a functor. Since the functor $\overline{Q}$ is surjective on objects and morphisms, we have $G = G'$ as soon as $G \circ \overline{Q} = G' \circ \overline{Q}$. Thus $\overline{Q}$ satisfies the universal property that characterizes the quotient functor Q_Σ. □

Corollary 8.3.15. The functor Q_Σ is surjective on morphisms.

Corollary 8.3.16. The category C/Σ is loop-free.

Proof. Consider the diagram on Figure 8.13 in which $\partial^- \delta \sim \partial^+ \gamma$, then we have $\partial^- \delta \sim \partial^+ \gamma'$ from which one deduces, from Lemma 8.2.9, that γ and δ belongs to Σ. We conclude by Lemma 8.3.13. □

The next result actually claims that Q_Σ is not far from being an equivalence of categories.

Theorem 8.3.17. If $C(x, y)$ is nonempty then the following map is a bijection.

$$\delta \in C(x, y) \;\mapsto\; Q_\Sigma(\delta) \in C/\Sigma\big(Q_\Sigma(x), Q_\Sigma(y)\big)$$

Proof. Any element of $C/\Sigma(Q_\Sigma(x), Q_\Sigma(y))$ is the equivalence class of some δ' whose source x' and target y' are equivalent to x and y. Then $x \wedge x'$ and $y \vee y'$ are provided by Theorem 8.3.10. Since the arrows $y \to y \vee y'$ and $x \wedge x' \to x$ are weak isomorphisms there is a unique $\delta \in C[x, y]$ such that Diagram 1 on Figure 8.11 commutes. □

Corollary 8.3.18. If $C/\Sigma\big(Q_\Sigma(x), Q_\Sigma(y)\big)$ is nonempty then there exist x' and y' such that $\Sigma(x', x)$, $\Sigma(y, y')$, $C(x', y)$, and $C(x, y')$ are nonempty.

Proof. By Corollary 8.3.15 we have some morphism $\alpha : a \to b$ whose image by Q_Σ belongs to $C/\Sigma(Q_\Sigma(x), Q_\Sigma(y))$. Then Theorem 8.3.10 provides $x \wedge a$ and for Σ is a system of weak isomorphisms we have the pushout square on Figure 8.14. Then Theorem 8.3.10 provides $y \vee c$ so $C(x, y \vee c)$ is nonempty and we put $y' = y \vee c$. Then by Theorem 8.3.10 we have $b \wedge y$ so we can consider the pullback on Figure 8.14 thus providing x'.

In an analogous way we find an object x' such that both $\Sigma(x', x)$ and $C(x', y)$ are nonempty. □

Corollary 8.3.19. The quotient C/Σ is a poset iff C is a preorder.

Proof. By Corollary 8.3.18 all pairs of objects of C/Σ can be obtained as $[x], [y]$ with $C(x, y)$ nonempty. It follows from Theorem 8.3.17 that C/Σ is a preorder iff C is a preorder. We conclude by Corollary 8.3.16. □

$$\begin{tikzcd}
x \arrow[rr] & & c \arrow[r, "\Sigma"] \arrow[dll, phantom, "\llcorner", very near start] & y' = y \vee c \\
x \wedge a \arrow[u, "\Sigma"] \arrow[r, "\Sigma"'] & a \arrow[r, "\alpha"'] & b \arrow[u, "\Sigma"'] & \\
x' \arrow[u, "\Sigma"] \arrow[rr] \arrow[urr, phantom, "\urcorner", very near start] & & b \wedge y \arrow[u, "\Sigma"'] \arrow[r, "\Sigma"'] & y \arrow[uu, "\Sigma"']
\end{tikzcd}$$

Figure 8.14: Bringing extremities into line

Then combining Theorem 8.3.17 and Corollary 8.3.18 we see that for any morphism $\delta \in C/\Sigma(Q_\Sigma(x), Q_\Sigma(y))$ there is a unique α in $C(x, y')$, resp. in $C(x', y)$, such that $Q_\Sigma(\alpha) = \delta$. In other words when we need α such that $Q_\Sigma(\alpha) = \delta$ we can always choose one of its extremities in the suitable Σ-component: this is the **lifting property** of Q_Σ. As a consequence Q_Σ preserves potential weak isomorphisms.

Corollary 8.3.20. The functor Q_Σ reflects and preserves potential weak isomorphisms.

Proof. Let $\alpha \in C(x, y)$ such that $Q_\Sigma(\alpha)$ is a potential weak isomorphism and f a morphism of C from x to z such that $C(y, z)$ is not empty. There is a unique β' from $Q_\Sigma(y)$ to $Q_\Sigma(z)$ such that $Q_\Sigma(f) = \beta' \circ Q_\Sigma(\alpha)$. We have a unique morphism β in $C(y, z)$ such that $Q_\Sigma(\beta) = \beta'$ and it is the only one in $C(y, z)$ such that $f = \beta \circ \alpha$ by Theorem 8.3.17. We have proved that α preserves the future cone and we prove in the same way that it preserves the past one, so Q_Σ reflects weak isomorphisms.
Now suppose that $\alpha \in C(x, y)$ is a potential weak isomorphism and let $f' \in C/\Sigma([x], z')$ such that $C/\Sigma([y], z') \neq \emptyset$. By Corollary 8.3.18 there exists z such that $[z] = z'$ and $C(y, z) \neq \emptyset$. Then by Theorem 8.3.17 there is a unique $f \in C(x, z)$ such that $[f] = f'$ and for α is a potential weak isomorphism there is a unique $g \in C(y, z)$ such that $f = g \circ \alpha$. So $g' = [g]$ satisfies $f' = g' \circ [\alpha]$ and it is the only one in $C/\Sigma([y], [z])$ by Theorem 8.3.17. So Q_Σ preserves potential weak isomorphisms. □

Corollary 8.3.21 (Finiteness)**.** If C is finite then so is the quotient C/Σ

Proof. The functor Q_Σ is surjective on morphisms by Corollary 8.3.15. □

By the universal property of the functor I_Σ there exists a unique functor P_Σ such that $Q_\Sigma = P_\Sigma \circ I_\Sigma$. We will prove that P_Σ is an equivalence of categories. A **fraction** is a pair of morphisms (γ, σ) with $\sigma \in \Sigma$, $\partial^+(\sigma) = x$, $\partial^-(\sigma) = \partial^-(\gamma)$, and $\partial^+(\gamma) = y$. Two fractions (γ, σ) and (γ', σ') are $\sim_{x,y}$-related when there exists two morphisms $\tau, \tau' \in \Sigma$ such that Diagram 1 on Figure 8.15 commutes. It is proved in Borceux (1994a) that if Σ is a right calculus of fractions, then the morphisms of $C[\Sigma^{-1}][x, y]$ are the $\sim_{x,y}$-equivalence classes of fractions. The composition is defined as suggested by Diagram 2 on Figure 8.15. In particular $(\gamma, \sigma) \sim_{x,y} (\gamma', \sigma')$ implies that $\gamma \sim \gamma'$ i.e. $[\gamma] = [\gamma']$. Thus, if κ is the $\sim_{x,y}$-equivalence class of (γ, σ) then $\kappa = I_\Sigma(\gamma) \circ (I_\Sigma(\sigma))^{-1}$ and $P_\Sigma(\kappa) = [\gamma]$.

Lemma 8.3.22. If $C(x, y) \neq \emptyset$ then I_Σ induces a bijection from $C(x, y)$ to $C[\Sigma^{-1}](x, y)$.

Proof. Let the fraction (γ, σ) be a representative of an element of $C[\Sigma^{-1}](x, y)$. Since σ is a weak isomorphism there is a unique $\delta \in C(x, y)$ such that $\gamma = \delta \circ \sigma$. Hence $(\delta, \mathrm{id}_x) \sim_{x,y} (\gamma, \sigma)$. If $(\delta_1, \mathrm{id}_x) \sim_{x,y} (\delta_2, \mathrm{id}_x)$ then we have v and $\tau_1, \tau_2 \in \Sigma(v, x)$ such that $\delta_1 \circ \tau_1 = \delta_2 \circ \tau_2$. It follows from Lemma 8.3.7 that $\tau_1 = \tau_2$ and since τ_1 is a weak isomorphism we get $\delta_1 = \delta_2$. □

$$\begin{tikzcd}
 & y & \\
u \arrow[ur, "\gamma"] \arrow[dr, "\sigma"'] & v \arrow[l, dotted, "\tau"'] \arrow[r, dotted, "\tau'"] & u' \arrow[ul, "\gamma'"'] \arrow[dl, "\sigma'"] \\
 & x &
\end{tikzcd}$$

Diagram 1

$$\begin{tikzcd}
 & v \arrow[r, "\delta"] \arrow[dr, "\tau\in\Sigma"] & z \\
w \arrow[ur, dotted, "\gamma'"] \arrow[dr, dotted, "\tau'\in\Sigma"'] & & y \\
 & u \arrow[ur, "\gamma"] \arrow[r, "\sigma\in\Sigma"'] & x
\end{tikzcd}$$

Diagram 2

Figure 8.15: Equivalent fractions and composition of fractions

Theorem 8.3.23 (Haucourt (2006))**.** The functor P_Σ is an equivalence of categories.

Proof. The functor Q_Σ is surjective on objects by construction hence so is P_Σ. Let x and y be two objects of C and $f \in C/\Sigma(Q_\Sigma(x), Q_\Sigma(y))$. By Corollary 8.3.18 there exists $\sigma \in \Sigma(y, y')$ such that $C[x, y'] \neq \emptyset$. By Theorem 8.3.17 there is a unique $\delta \in C(x, y')$ such that $Q_\Sigma(\delta) = f$. Then $\xi := (I_\Sigma(\sigma))^{-1} \circ I_\Sigma(\delta)$ belongs to $C[\Sigma^{-1}][x, y]$ and satisfies $P_\Sigma(\xi) = f$. Suppose that ξ' is another such morphism. Then $I_\Sigma(\sigma) \circ \xi' \in C[\Sigma^{-1}](x, y')$ and by Lemma 8.3.22 there is a unique $\delta' \in C(x, y')$ such that $I_\Sigma(\delta') = I_\Sigma(\sigma) \circ \xi'$. Moreover

$$Q_\Sigma(\delta') = P_\Sigma(I_\Sigma(\delta')) = P_\Sigma(I_\Sigma(\sigma) \circ \xi') = P_\Sigma(\xi') = f$$

Hence $\delta' = \delta$ from which comes $I_\Sigma(\sigma) \circ \xi' = I_\Sigma(\sigma) \circ \xi$, and thus $\xi' = \xi$. □

Corollary 8.3.24. The skeleton of $C[\Sigma^{-1}]$ is C/Σ and $C[\Sigma^{-1}]$ is one-way.

Proof. The quotient C/Σ is skeletal by Corollary 8.3.16 and it is equivalent to $C[\Sigma^{-1}]$ by Theorem 8.3.23. Hence $C[\Sigma^{-1}]$ is one-way by Lemma 8.1.5. □

From a theoretical point of view, Theorem 8.3.23 implies that it is unimportant whether one defines the category of components by means of fractions or quotients. Yet the latter might have a finite presentation even if the former has infinitely many objects. A similar result was proved by Clerc and Mimram (2015) in the context of rewriting theory.

8.4 Sections of the Quotient Functor

We define the category of components in an alternative way. This approach turns out to be fruitful when trying to extend the notion of component to categories with loops (*e.g.* the fundamental category of the directed circle). However it poses a technical problem related to the Axiom of Choice. In any category when the composite $r \circ s$ is an identity one says that r is a **retract** of s and that s is a **section** of r. An **embedding** is a full and faithful functor that is one-to-one on objects. We will study the sections of Q_Σ and prove that:

1. any section of Q_Σ is an embedding,

2. the sections of Q_Σ form a preordered set with binary l.u.b. and g.l.b., and

3. the functor Q_Σ admits a section when C/Σ is finite.

Proposition 8.4.1. Any section of Q_Σ is an embedding.

Proof. Any section of any functor is faithful and one-to-one on objects. Let $f \in \mathcal{C}(S(a),S(b))$, we have $Q_\Sigma \circ S \circ Q_\Sigma(f) = Q_\Sigma(f)$ hence $Q_\Sigma(f) \in \mathcal{C}/\Sigma[a,b]$ and $S \circ Q_\Sigma(f) = f$ for Q_Σ is faithful by Theorem 8.3.17. □

Corollary 8.4.2. Every section of Q_Σ reflects identities, isomorphisms, potential weak isomorphisms, pushout and pullback squares.

Proof. It is an immediate consequence of Proposition 8.4.1 and the fact that if $\mathcal{A}$ is a full subcategory of $\mathcal{B}$, then every morphism (resp. diagram) of $\mathcal{A}$ which is a potential weak isomorphism (resp. a pushout/pullback square) of $\mathcal{B}$ is also a potential weak isomorphism (respectively pushout/pullback square) of $\mathcal{A}$. □

Proposition 8.4.3. Every section of Q_Σ preserves the potential weak isomorphisms.

Proof. Let σ' be a potential weak isomorphism of $\mathcal{C}/\Sigma$. Then $S(\sigma') \in \mathcal{C}(x,y)$ and let z be such that $\mathcal{C}(y,z) \neq \emptyset$, and $\delta \in \mathcal{C}(x,z)$. We have a unique γ' such that $Q_\Sigma(\delta) = \gamma' \circ \sigma'$. Then by Theorem 8.3.17 we have a unique $\gamma \in \mathcal{C}(y,z)$ such that $\delta = \gamma \circ S(\sigma')$, it is indeed given by the unique γ such that $Q_\Sigma(\gamma) = \gamma'$. □

The collection of objects of a category is preordered putting x less than y when the homset from x to y is nonempty.

Proposition 8.4.4. Any morphism of preordered sets $\phi : Obj(\mathcal{C}/\Sigma) \to Obj(\mathcal{C})$ such that $\phi(K) \in K$ for all $K \in Obj(\mathcal{C}/\Sigma)$ extends to a section of Q_Σ in a unique way.

Proof. The object part of the section S is given by ϕ and by Theorem 8.3.17 we define $S(\delta)$, for $\delta \in \mathcal{C}/\Sigma(K,K')$, as the unique γ in $\mathcal{C}(S(K),S(K'))$ such that $Q_\Sigma(\gamma) = \delta$. □

The collection of sections of Q_Σ is preordered as follows: $S \preccurlyeq S'$ when $S(K) \preccurlyeq S'(K)$ for all $K \in \mathrm{Obj}(\mathcal{C}/\Sigma)$. Then as an immediate consequence of Theorem 8.3.10 we have

Corollary 8.4.5. The preordered set of sections of Q_Σ is a prelattice.

Still by Theorem 8.3.10 we know that each Σ-component of $\mathcal{C}$ is a poset with binary l.u.b.'s and g.l.b.'s from which it follows that there is an inclusion from the collection of sections of Q_Σ to the Cartesian product of all the Σ-components of $\mathcal{C}$. This inclusion preserves l.u.b.'s and g.l.b.'s and it is an isomorphism iff the following property is satisfied for all Σ-components K and K':

$$(\exists x \in K \exists x' \in K', \mathcal{C}[x,x'] \neq \emptyset) \Rightarrow (\forall x \in K \forall x' \in K', \mathcal{C}[x,x'] \neq \emptyset)$$

which is the case when $\mathcal{C}$ derives from an equivalence relation (*i.e.* when $\mathcal{C}$ is a groupoid each homset of which admits at most one element) such a category is also called a **contractible groupoid**. In this case the unique system of weak isomorphisms is the collection of isomorphisms of $\mathcal{C}$, the Σ-components are just the connected components, and each Σ-component inherits the chaotic preorder of $\mathcal{C}$, and thus a section of Q_Σ is just a function of choice. In particular if we suppose that the quotient functor Q_Σ admits a section for all systems of weak isomorphisms of all one-way categories, then the Axiom of Choice is satisfied. The converse, (*i.e.* the existence of such sections within Zermelo-Fraenkel set theory with Axiom of Choice) is currently an open problem. Nevertheless we have

Proposition 8.4.6. For all systems of weak isomorphisms Σ of all one-way categories C, and for all finite subset $\mathcal{K}$ of $\mathrm{Obj}(C/\Sigma)$, there exists a morphism of preordered sets S from $\mathcal{K}$ to $\mathrm{Obj}(C)$ such that $S(K) \in K$ for all $K \in \mathcal{K}$.

Proof. By induction on the cardinality n of $\mathcal{K}$. If $n = 0$ the empty map fits. Suppose that we have the statement for some $n \in \mathbb{N}$ and let $\mathcal{K}$ be of cardinality $n + 1$. Then let K_m be a maximal element of $\mathcal{K}$ with respect to the preorder inherited from C/Σ. For all $K \neq K_m$, we have, by Corollary 8.3.18, some x_K in K_m such that $S(k) \preccurlyeq x_K$. Then define

$$S(K_m) = \bigvee_{K \neq K_m} x_K$$

which is the least upper bound of a finite family in a preordered set which admits binary least upper bounds – see Theorem 8.3.10. □

Corollary 8.4.7. If C/Σ has finitely many objects then Q_Σ admits a section.

If Σ is the greatest system of weak isomorphisms of C then it seems rather natural to imagine that the greatest system of weak isomorphisms of C/Σ is trivial (*i.e.* only contains isomorphisms which are actually identities since C/Σ is loop-free by Corollary 8.3.16). It would mean, in particular, that the category of components construction is idempotent. Up to now however, I have been unable to prove it. Yet we have

Proposition 8.4.8. Denote by Σ and Σ' the greatest systems of weak isomorphisms of $C$and C/Σ, t.f.a.e.
1 – Σ' is trivial
2 – C/Σ has a unique system of weak isomorphisms
3 – $Q_\Sigma^{-1}(\Sigma')$ is a system of weak isomorphisms
moreover if S is a section of Q_Σ we also have a logical equivalence with
4 – $S(\Sigma') \subseteq \Sigma$
5 – $S^{-1}(\Sigma) = \Sigma'$

Proof. Assertions 1) and 2) are equivalent because the set of identities of C/Σ is the least system of weak isomorphisms of a loop-free category and C/Σ is loop-free by Corollary 8.3.16. Suppose that assertion 3) is satisfied, then $Q_\Sigma^{-1}(\Sigma') \subseteq \Sigma$. Since Q_Σ is surjective on morphisms – by Corollary 8.3.15, we have $Q_\Sigma(Q_\Sigma^{-1}(\Sigma')) = \Sigma'$ hence $\Sigma' \subseteq Q_\Sigma(\Sigma) = \{\text{identities}\}$. The converse is obvious. The fourth assertion implies the first one since

$$\Sigma' = Q_\Sigma(S(\Sigma')) \subseteq Q_\Sigma(\Sigma) = \{\text{identities}\}$$

The converse is obvious. Let $\sigma' \in S^{-1}(\Sigma)$ be, then $\sigma' = Q_\Sigma(S(\sigma')) = \mathrm{id}$. Hence the first assertion implies the fifth one which obviously implies the fourth one. □

8.5 Components of a Product

We study the systems of weak isomorphisms and the categories of components of (co)products of one-way categories. Consider to this end a family C_i one-way categories with i ranging through some indexing set $\mathcal{I}$. The product of this family will be denoted by $\mathcal{P}$, and the projections are denoted by $p_i : \mathcal{P} \to C_i$.

Lemma 8.5.1. A morphism σ of $\mathcal{P}$ is a potential weak isomorphism iff for all $i \in \mathcal{I}$, $p_i(\sigma)$ is a potential weak isomorphism of C_i.

Proof. Let x and y be the source and target of σ. Fix $i \in \mathcal{I}$ and $\delta_i \in C_i(p_i(x), z_i)$ be such that $C_i(p_i(y), z_i) \neq \emptyset$. Then define δ by $p_i(\delta) = \delta_i$ and $p_j(\delta) = p_j(\sigma)$ for $i \neq j$. Because σ is supposed to be a potential weak isomorphism, there is a unique γ such that $\gamma \circ \sigma = \delta$. Hence $p_i(\gamma) \circ p_i(\sigma) = \delta_i$. The uniqueness of $p_i(\gamma)$ derives from the one of γ. So σ is a potential weak isomorphism of C_i. The converse is obvious. □

The same pattern of proof applies to pull-backs and push-outs.

Lemma 8.5.2. A commutative square of $\mathcal{P}$ is a pushout (resp. pullback) iff for all $i \in \mathcal{I}$, its image by p_i is a pushout (resp. pullback) of C_i.

Corollary 8.5.3. Given a family $\Sigma_i \subseteq C_i$ with $i \in \mathcal{I}$, $\prod_{i \in \mathcal{I}} \Sigma_i$ is a system of weak isomorphisms iff all the Σ_i are systems of weak isomorphisms.

The preceding corollary does not mean that all systems of weak isomorphisms of a product is a corresponding product of systems of weak isomorphisms. One can remark indeed that the projection of a system of weak isomorphisms may not be stable under composition. Consider for example the product

$$\{0,1\} \times (\mathbb{R}, \leqslant)$$

The collection of morphisms $(\varepsilon, (x \leqslant y))$ such that:
$y \leqslant 0$ or $0 < x$ if $\varepsilon = 0$, and
$y < 0$ or $0 \leqslant x$ if $\varepsilon = 1$
forms a system of weak isomorphisms that is stable under composition though its projection on $(\mathbb{R}, \leqslant)$ is not.

Lemma 8.5.4. For all $i \in \mathcal{I}$, we have a morphism of poset

$$\Sigma \in \mathrm{SWIC}(\mathcal{P}) \quad \mapsto \quad < p_i(\Sigma) > \in \ \mathrm{SWIC}(C_i)$$

Proof. All the elements of $p_i(\Sigma)$ are potential weak isomorphisms by Lemma 8.5.1. Given $\sigma \in \Sigma$ we write σ_i for $p_i(\sigma)$. If γ_i is a morphism of C_i with $\partial^- \gamma_i = \partial^- \sigma_i$, define the morphism $\gamma \in \mathcal{P}$ by $p_i(\gamma) = \gamma_i$, and $p_j(\gamma) = \mathrm{id}_{\partial^- p_j \sigma}$ for $j \neq i$. The pushout of σ along γ exists and belongs to Σ (for the latter is a system of weak isomorphisms) and its image by p_i is, by Lemma 8.5.2, the pushout of σ_i along γ_i. The closure under composition is still a system of weak isomorphisms by Lemma 8.2.6. The remaining facts hold by Corollary 8.5.3. □

Corollary 8.5.5. The mapping P that sends an $\mathcal{I}$-family $\Sigma_i \in \mathrm{SWIC}(C_i)$ to its product, and the mapping S that sends $\Sigma \in \mathrm{SWIC}(\mathcal{P})$ to the $\mathcal{I}$-family $< p_i(\Sigma) >$, satisfy

$$S \circ P = \mathrm{id} \quad \text{and} \quad \mathrm{id} \subseteq P \circ S$$

Lemma 8.5.6. Let σ be an element of a system of weak isomorphisms Σ of $\mathcal{P}$, and let δ be such that $\partial^- \delta = \partial^- \sigma$ or $\partial^- \delta = \partial^+ \sigma$. Then $\sigma' \in \Sigma$ with σ' defined, for some $J \subseteq \mathcal{I}$, by $p_i(\sigma') = p_i(\sigma)$ if $i \in J$, and $p_i(\sigma') = \mathrm{id}_{\partial^+ \delta_i}$ otherwise.

Proof. Given a morphism $\gamma \in \mathcal{P}$, and $J \subseteq \mathcal{I}$ we define γ_J^- and γ_J^+ by:
$p_i(\gamma_J^-) = p_i(\gamma_J^+) = p_i(\gamma)$ when $i \in J$, and
$p_i(\gamma_J^-) = \mathrm{id}_{\partial^- \gamma_i}$ and $p_i(\gamma_J^+) = \mathrm{id}_{\partial^+ \gamma_i}$ when $i \notin J$.
The commutative square $\gamma^+_{\mathcal{I} \backslash J} \circ \gamma^-_J = \gamma^+_J \circ \gamma^-_{\mathcal{I} \backslash J}$ is both a pushout and a pullback. For all $\sigma \in \Sigma$, a system of weak isomorphisms of $\mathcal{P}$, σ_J^+ and σ_J^- belong to Σ by Lemma 8.2.9. Suppose that $\partial^- \delta = \partial^- \sigma$ and define γ by:

$p_i(\gamma) = p_i(\sigma)$ for $i \in J$, and
$p_i(\gamma) = p_i(\delta)$ for $i \notin J$.
Then note that $\gamma^-_J = \sigma^-_J$ hence $\sigma' := \gamma^+_J$ belongs to Σ which is stable under pushout. In the case where $\partial^- \delta = \partial^+ \sigma$, just replace δ by $\delta \circ \sigma$ in the previous proof. □

Remark 8.5.7. A category is said to be **connected** when any two objects of it are related by a zigzag of morphisms. If $\mathcal{P}$ is connected then so are all the categories C_i. The converse may however be false when $\mathcal{I}$ is infinite. Consider indeed the product of the family W_n, for $n \in \mathbb{N}$, with W_n a zigzag of length n. Each W_n is connected though the product is not.

Corollary 8.5.8. Assume that $\mathcal{P}$ is connected. If σ belongs to a system of weak isomorphisms, then so does the morphism obtained by changing, for all i in a given set of index J, the i-coordinate of σ into some identity of C_i.

Proof. Consider an object of $\mathcal{P}$ whose i-coordinate, for $i \notin J$, is $\partial^- p_i(\sigma)$. Then $\partial^- \sigma$ is related to this object by a zigzag of morphisms such that for all $i \notin J$ the i-coordinate of all morphisms of the zigzag is $\mathrm{id}_{\partial^-(p_i(\sigma))}$. Then apply Lemma 8.5.6 along the zigzag. □

Proposition 8.5.9. If $\mathcal{P}$ is connected and Σ is stable under composition, then each $p_i(\Sigma)$ is stable under composition. Moreover if $\mathcal{I}$ is finite then $P \circ S = \mathrm{id}$.

Proof. If $\sigma, \sigma' \in \Sigma$, and $p_i(\sigma)$ and $p_i(\sigma')$ compose, then by Corollary 8.5.8 we can change all the j-coordinates of σ and σ', for $j \neq i$, so that σ and σ' compose and still belong to Σ. It follows that $p_i(\Sigma)$ is stable under composition. Assume $\mathcal{I} = \{1, \ldots, n\}$ is finite and let Σ be a system of weak isomorphisms of $\mathcal{P}$. Let $\sigma \in \mathcal{P}$ be such that for all $i \in \mathcal{I}$, $p_i(\sigma) \in p_i(\Sigma)$. For $k \in \{1, \ldots, n\}$ define $\tau^{(k)}$ as

$$(\mathrm{id}_{\partial^+ p_1(\sigma)}, \ldots, \mathrm{id}_{\partial^+ p_{k-1}(\sigma)}, p_k(\sigma), \mathrm{id}_{\partial^- p_{k-1}(\sigma)}, \ldots, \mathrm{id}_{\partial^- p_n(\sigma)})$$

Each morphism $\tau^{(k)}$ belongs to Σ by Corollary 8.5.8, hence so does the composite namely σ. □

Proposition 8.5.10. Let Σ_i be a system of weak isomorphisms of C_i, for $i \in \mathcal{I}$, then

$$\prod_{i \in \mathcal{I}} (C_i/\Sigma_i) \quad \cong \quad \Big(\prod_{i \in \mathcal{I}} C_i\Big) \Big/ \Big(\prod_{i \in \mathcal{I}} \Sigma_i\Big)$$

Proof. By Corollary 8.5.3 the product of the family of systems Σ_i, denoted by Σ, is a system of weak isomorphisms of the product of categories $\mathcal{P}$. Thus both terms of the relation make sense. From Proposition 8.3.14 we deduce that two objects (resp. morphisms) of $\mathcal{P}$ are Σ-equivalent iff their projections on i are Σ_i-equivalent for all $i \in \mathcal{I}$. □

Proposition 8.5.11. For all $i \in \mathcal{I}$, let Σ_i be a system of weak isomorphisms of C_i, then

$$\prod_{i \in \mathcal{I}} (C_i[\Sigma_i^{-1}]) \quad \cong \quad \Big(\prod_{i \in \mathcal{I}} C_i\Big)\Big[\Big(\prod_{i \in \mathcal{I}} \Sigma_i\Big)^{-1}\Big]$$

Proof. As a consequence of Remark 8.2.7, the category $C[\Sigma^{-1}]$ admits a nice description (Borceux, 1994a, Proposition 5.2.4, p.183-184) from which one easily deduces the result. □

8.6 A Homotopical Perspective on Categories of Components

As explained in Section 8.2 the systems of weak isomorphisms are intended to extend the collection of isomorphisms of one-way categories. Following this idea it would be nice that they are preserved by functors, however it is obviously not the case: the unique non trivial morphism of the category $\{\cdot \to \cdot\}$ might indeed be sent to an arrow that is not even a potential weak isomorphism. Then it is natural to consider $\mathbf{Ow}_{\mathrm{h}}$ the category of one-way categories with functors preserving the greatest systems of weak isomorphisms. As a consequence of Corollary 8.3.16 and the universal property of the quotient category, the category of components construction induces a functor

$$\overrightarrow{\pi_0} \ : \ \mathbf{Ow}_{\mathrm{h}} \to \mathbf{Lf}$$

Following Dwyer et al. (2004), there is actually a homotopic interpretation of this. A category is said to be **homotopical** when it comes with a class $\mathcal{W}$ of distinguished morphisms, whose elements are called the **weak equivalences**, satisfying the following properties:

1) all the identities belong to $\mathcal{W}$, and

2) **2 out of 6**: when $\gamma \circ \beta$ and $\beta \circ \alpha$ (exist and) belong to $\mathcal{W}$, then α, β, γ, and $\gamma \circ \beta \circ \alpha$ also belong to $\mathcal{W}$.

Alternatively the second condition is equivalent to the conjunction of the following two ones:

2') **weak invertibility**: any morphism β for which there are α and γ such that $\gamma \circ \beta \in \mathcal{W}$ and $\beta \circ \alpha \in \mathcal{W}$, belongs to $\mathcal{W}$, and

2") **2 out of 3**: for every two morphisms α and β for which $\partial^{-}\beta = \partial^{+}\alpha$ and two of α, β, and $\beta \circ \alpha$ belong to $\mathcal{W}$, so is the third.

The **homotopy category** of a homotopical category C with its class of weak equivalences $\mathcal{W}$ is defined as the localization (*i.e.* $C[\mathcal{W}^{-1}]$). A functor between homotopical categories is then said to be **homotopical** when it preserves the weak equivalences. By Lemma 8.2.9 any one-way category that comes with a system of weak isomorphisms is a homotopical category. In particular the objects of $\mathbf{Ow}_{\mathrm{h}}$ are homotopical categories and its morphisms are the homotopical functors between them.

Proposition 8.6.1. If Σ is a system of weak isomorphisms of a one-way category C then $C[\Sigma^{-1}]$ is one-way.

Proof. The categories C/Σ and $C[\Sigma^{-1}]$ are equivalent by Theorem 8.3.23. The quotient C/Σ is loop-free by Corollary 8.3.16, and therefore it is skeletal. Hence $C[\Sigma^{-1}]$ is one-way by Lemma 8.1.5. □

As a consequence of Proposition 8.6.1 and the universal property of localization there is a functor sending each homotopical category to its homotopy category.

$$\mathrm{Ho} \ : \ \mathbf{Ow}_{\mathrm{h}} \to \mathbf{Ow}$$

The relation to homotopy actually goes further: the notion of homotopical category has been designed to generalize the concept of **model category** which was identified as an abstract framework for homotopy theory by Quillen (1967), see also Hovey (1999). Such a structure is made of three distinguished classes of morphisms, namely the fibrations, the cofibrations, and the weak equivalences, that are related by a series of axioms. In

$$\begin{tikzcd}
\mathcal{C}[\Sigma^{-1}] \arrow[rrr, "h"] \arrow[dd, "P_\Sigma"'] & & & \mathcal{D}[\Sigma'^{-1}] \arrow[dd, "P_{\Sigma'}"] \\
 & \mathcal{C} \arrow[ul, "I_\Sigma"'] \arrow[r, "f"] \arrow[dl, "Q_\Sigma"] & \mathcal{D} \arrow[ur, "I_{\Sigma'}"] \arrow[dr, "Q_{\Sigma'}"'] & \\
\mathcal{C}/\Sigma \arrow[rrr, "g"'] & & & \mathcal{D}/\Sigma'
\end{tikzcd}$$

Figure 8.16: Characterizing the weak equivalences of $\mathbf{Ow}_{\mathrm{h}}$

particular the last class satisfies the 2 out of 3 property. It is a well-known fact by model category theorists that there is a unique model category structure over **Cat** whose weak equivalences are precisely the usual equivalences of categories. One refers to it as the canonical model structure on **Cat**. In regard with the fourth point of Theorem 8.3.10, a satisfactory class of weak equivalences should at least contains all the functors from a prelattice to $\{*\}$, the terminal object of **Cat**. The restriction of the canonical model category to **Ow** is therefore not very interesting since no nontrivial lattice – seen as a category, is equivalent to $\{*\}$. However the category $\mathbf{Ow}_{\mathrm{h}}$ itself can be equipped with a homotopical category structure by defining $\mathcal{W}$ as the class of morphisms of $\mathbf{Ow}_{\mathrm{h}}$ whose image under $\overrightarrow{\pi_0}$ is an isomorphism. The homotopical category structure of $\mathbf{Ow}_{\mathrm{h}}$ we have defined is thus related to the canonical model category over **Cat**, as we show now.

Proposition 8.6.2. Given a functor $f \in \mathbf{Ow}_{\mathrm{h}}(\mathcal{C}, \mathcal{D})$ with Σ and Σ' the systems of weak isomorphisms of $\mathcal{C}$ and $\mathcal{D}$, the unique functor $g : \mathcal{C}/\Sigma \to \mathcal{D}/\Sigma'$ such that $g \circ Q_\Sigma = Q_{\Sigma'} \circ f$ is an isomorphism iff the unique functor $h : \mathcal{C}[\Sigma^{-1}] \to \mathcal{D}[\Sigma'^{-1}]$ such that $h \circ I_\Sigma = I_{\Sigma'} \circ f$ is an equivalence of categories.

Proof. Observe the commutative diagram on Figure 8.16 keeping in mind that, by Corollary 8.3.16, both $\mathcal{C}/\Sigma$ and $\mathcal{D}/\Sigma'$ are loop-free. Hence any equivalence between them is actually an isomorphism. Moreover, both P_Σ and $P_{\Sigma'}$ are equivalences of categories by Theorem 8.3.10. The conclusion follows from the fact that the collection of equivalences of categories satisfy the 2 out of 3 property. □

Theorem 8.6.3. All equivalences of categories between one-way categories are weak equivalences.

Proof. By Lemma 8.2.10 any equivalence of category belongs to $\mathbf{Ow}_{\mathrm{h}}$. Let E be an equivalence of categories and F be its quasi inverse. We denote by E' and F' the unique functors such that

$$Q_{\Sigma_{\mathcal{D}}} \circ E = E' \circ Q_{\Sigma_{\mathcal{C}}} \text{ and } Q_{\Sigma_{\mathcal{C}}} \circ F = F' \circ Q_{\Sigma_{\mathcal{D}}}$$

There are natural isomorphisms $\alpha : E \circ F \to \mathrm{id}_{\mathcal{D}}$ and $\beta : F \circ E \to \mathrm{id}_{\mathcal{C}}$. It follows that $Q_{\Sigma_{\mathcal{D}}} * \alpha$ and $Q_{\Sigma_{\mathcal{C}}} * \beta$ are identities in a category of functors. Therefore we have

$$Q_{\Sigma_{\mathcal{D}}} = Q_{\Sigma_{\mathcal{D}}} \circ E \circ F = E' \circ F' \circ Q_{\Sigma_{\mathcal{D}}} \text{ and } Q_{\Sigma_{\mathcal{C}}} = Q_{\Sigma_{\mathcal{C}}} \circ F \circ E = F' \circ E' \circ Q_{\Sigma_{\mathcal{C}}}$$

By the universal properties of $Q_{\Sigma_{\mathcal{C}}}$ and $Q_{\Sigma_{\mathcal{D}}}$ we have

$$E' \circ F' = \mathrm{id}_{\mathcal{D}} \text{ and } F' \circ E' = \mathrm{id}_{\mathcal{C}}$$

□

A **natural weak equivalence** is a natural transformation which sends objects to weak equivalences therefore we have:

Corollary 8.6.4. The collection of functors $P_\Sigma : \mathcal{C}[\Sigma^{-1}] \to \mathcal{C}/\Sigma$ with $\mathcal{C}$ ranging through the collection of (small) one-way categories and Σ being the greatest system of weak isomorphisms of $\mathcal{C}$, forms a natural weak equivalence from Ho to $\overrightarrow{\pi_0}$.

One may ask whether I_Σ and Q_Σ are weak equivalences, the answer is actually related to the problem of idempotency of the functor $\overrightarrow{\pi_0}$.

Corollary 8.6.5. Let Σ be the greatest system of weak isomorphisms of the one-way category $\mathcal{C}$, t.f.a.e.

1) $Q_\Sigma : \mathcal{C} \to \mathcal{C}/\Sigma$ is a weak equivalence,

2) $I_\Sigma : \mathcal{C} \to \mathcal{C}[\Sigma^{-1}]$ is a weak equivalence, and

3) the greatest system of weak isomorphisms of $\mathcal{C}/\Sigma$ only contains identities.

Proof. We have $Q_\Sigma = P_\Sigma \circ I_\Sigma$ and P_Σ is a weak equivalence by Theorem 8.3.23 and Theorem 8.6.3, hence the assertions 1) and 2) are equivalent by the 2 out of 3 property. Denote by Σ' the greatest system of weak isomorphisms of $\mathcal{C}/\Sigma$ and F be the unique functor such that $F \circ Q_\Sigma = Q_{\Sigma'} \circ Q_\Sigma$. The universal property of Q_Σ implies that $F = Q'_\Sigma$. Then F is an isomorphism iff Σ' is the collection of identities of $\mathcal{C}/\Sigma$. □

Through the functor $\overrightarrow{\pi_0}$ we have endowed $\mathbf{Ow}_\mathrm{h}$ with a structure of homotopical category in which the class of weak equivalences strictly contains the class of equivalences of categories. To finish this section let us remark that the pushout in $\mathbf{Ow}_\mathrm{h}$ of two copies of $\{0 < 1\}$ over $\{0\}$ is the commutative square (*i.e.* $\{0 < 1\}^2$).

8.7 Components of Regions

An isothetic region is said to be **loop-free** when its fundamental category is so. Note that cubical regions (*cf.* Example 6.2.24) are special cases of loop-free regions. Loop-free regions thus provide a convenient directed topological playground for applying the results of the preceding sections. The following conjectures illustrate how the information encoded in the fundamental category of a (loop-free) region can be drastically reduced.

Conjecture 8.7.1. The components of (the fundamental category of) a loop-free region are loop-free regions.

In fact the dihomotopy class of a dipath is characterized by its image.

Conjecture 8.7.2. For all loop-free regions X, there exists a finite family $\mathcal{K}$ of loop-free subregions of X such that for all directed paths γ and δ on X sharing their sources and their targets, γ and δ are dihomotopic iff

$$\forall K \in \mathcal{K},\ img(\gamma) \subseteq K \Leftrightarrow img(\delta) \subseteq K$$

Figure 8.17 provides a simple illustration of Conjecture 8.7.2. The elements of $\mathcal{K}$ are called the dihomotopy **classifiers** and they may not cover the whole space – see

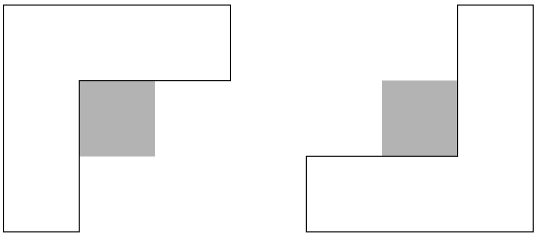


Figure 8.17: The dihomotopy classifiers of the complemented square

Figure 8.18. A candidate for $\mathcal{K}$ is obtained as follows. First define for all dipaths γ the collection $[\gamma]$ of all points of X visited by a dipath δ that is dihomotopic with γ:

$$[\gamma] := \{p \in X \mid \text{there exists } \delta \text{ dihomotopic with } \gamma \text{ such that } \delta \text{ covers } p\}$$

Then our next conjecture is the following.

Conjecture 8.7.3. For all dipath γ over a loop-free region X, the set $[\gamma]$ is a loop-free subregion of X.

Then consider the collection $\mathcal{K}'$ of all $[\gamma]$ such that $\overrightarrow{\pi_1}X[\partial^-\gamma, \partial^+\gamma]$ is not a singleton. The collection $\mathcal{K}$ would then be the set of $\subseteq$-maximal elements of $\mathcal{K}'$. Moreover, each element of $\mathcal{K}$ should be a finite union of Σ-components of $\overrightarrow{\pi_1}X$ where Σ is the greatest system of weak isomorphisms of $\overrightarrow{\pi_1}X$ (*cf.* Definition 8.3.8 and Lemma 8.2.11). It is worth noticing that Conjecture 8.7.2 is obviously wrong for regions in general: the image of a dipath covering the directed circle does not depend on its winding number. The notion of components remains puzzling in the presence of loops, still, we state some conjectures to extend the concept beyond the loop-free case.

Let G be a graph together with a binary relation ρ over the collection of paths on G such that two related paths share their sources and their targets. Then (G, ρ) is said to be a **presentation** of the category $F(G)/\rho^*$ where $F(G)$ is the category freely generated by G and ρ^* is the least congruence on it containing ρ, see (Mac Lane, 1998, p.52). Any category C is presented by the graph whose set of arrows consists of all the morphisms of C, two paths of which being related when their composites match. This presentation comes from the pair of adjoint functors initiated by the forgetful functor **Cat** → **Grph**, and it is actually the most expansive one in the sense that it has more redundancy that any other reasonable one. We will refer to this presentation as the **standard** one. Our purpose is to provide certain fundamental categories $\overrightarrow{\pi_1}X$ with a much cheaper presentation taking advantage of some extra hypothesis made on X. Such presentations are intended to approximate the category of components of $\overrightarrow{\pi_1}X$.

We first provide a presentation of $\overrightarrow{\pi_1}(\rceil G\lfloor)$ for a graph G. The vertices of the underlying graph of the presentation are the points of $\rceil G\lfloor$. Its arrows are the triples $t \cdot \alpha \cdot t'$ with α arrow of G and $0 \leqslant t < t' \leqslant 1$. We define the source of $t \cdot \alpha \cdot t'$ as $\partial^-\alpha$ if $t = 0$, and (α, t) otherwise. The target is defined accordingly. For all arrows α and all triples $t < t' < t''$ we have the relation

$$(t' \cdot \alpha \cdot t'') \circ (t \cdot \alpha \cdot t') = t \cdot \alpha \cdot t''$$

The previous presentation of $\overrightarrow{\pi_1}\rceil G\lfloor$ is clearly smaller than the generic one hence the category of components of $\overrightarrow{\pi_1}\rceil G\lfloor$ should at least remove all expandable vertices of G

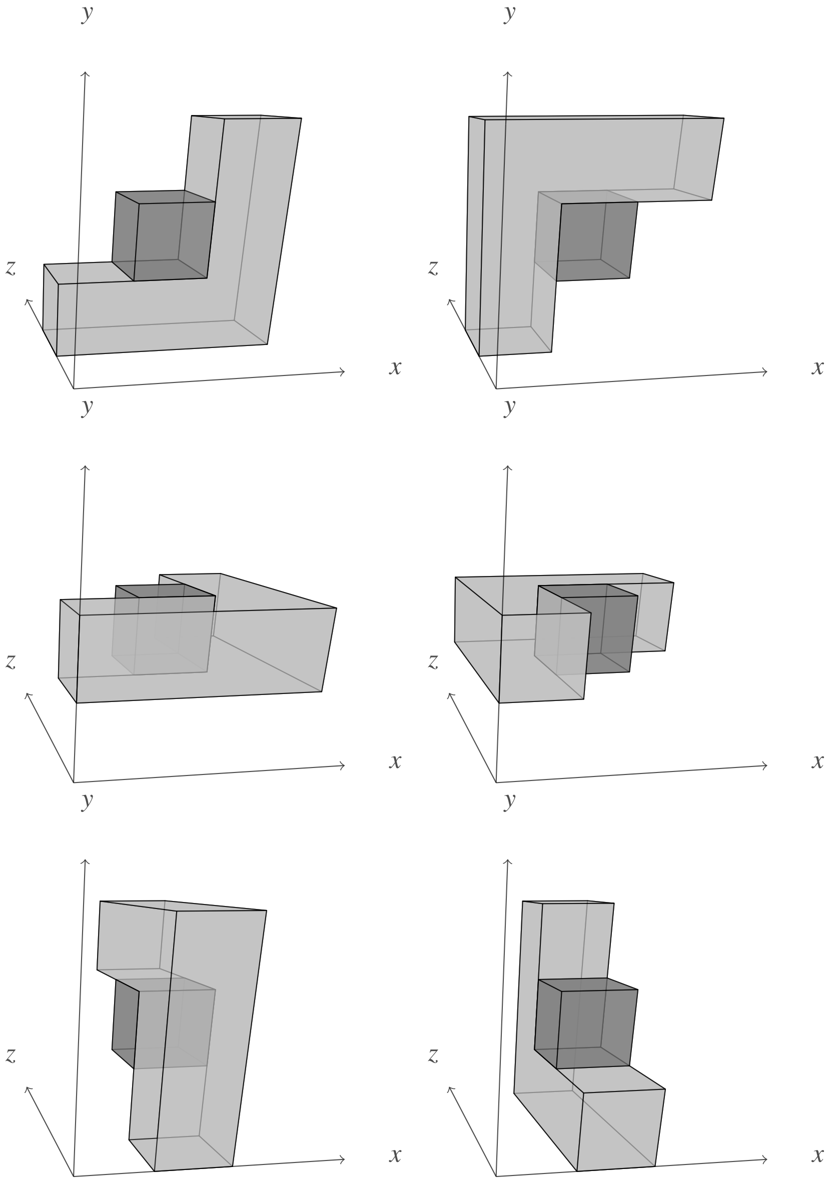


Figure 8.18: The dihomotopy classifiers of the complemented cube

(*cf.* Definition 6.1.26).

The higher dimensional case is much more intricate. As we have seen Conjecture 8.7.2 fails in the presence of loops, yet it can be weakened so it applies to all regions.

Lemma 8.7.4. Suppose that all the elements of a finite partition $\mathcal{P}$ of a region A are subregions of A, and let γ be a dipath on A. Then we have a finite ordered partition $\{I_1 < \ldots < I_N\}$ of $dom(\gamma)$ whose elements are intervals together with a mapping $P : \{1, \ldots, N\} \to \mathcal{P}$ such that for all $k \in \{1, \ldots, N\}$, $\gamma(I_k) \subseteq P_k$ and $P_k \neq P_{k+1}$.

Proof. The result derives from Proposition 6.1.7 and the fact that $\mathcal{B}_{\mathrm{dom}(\gamma)}$ is a Boolean algebra. Then if one considers the components of γ, say $(\gamma_1, \ldots, \gamma_n)$, and a subblock $B_1 \times \cdots \times B_n$ of A then

$$\gamma^{\text{-1}}(B_1 \times \cdots \times B_n) = \gamma_1{}^{\text{-1}}(B_1) \cap \cdots \cap \gamma_n{}^{\text{-1}}(B_n)$$

□

The finite sequence $P_1, \ldots, P_N$ is called the **trace** of γ on the partition $\mathcal{P}$. Conjecture 8.7.2 is then adapted to isothetic regions as follows:

Conjecture 8.7.5. For all regions A, there exist a partition of subregions $\mathcal{P}$ such that for all dipath γ over A the collection

$$\{\ \mathrm{trace}(\delta) \mid \delta \text{ is dihomotopic with } \gamma\ \}$$

is finite and the collection

$$W = \{\ \mathrm{trace}(\gamma) \mid \gamma \text{ is dipath on } A\ \}$$

is a regular language[1] over $\mathcal{P}$. Moreover there exists a finite collection $\rho \subseteq W \times W$ such that two dipaths sharing their sources and their targets are dihomotopic iff there traces are equivalent up to the congruence on W generated by ρ.

With the notation of Conjecture 8.7.5 the collection W may not be stable under concatenation. However if ω_1, ω_1', ω_2, ω_2', $\omega_2 \cdot \omega_1$ belong to W, and $\omega_1 \rho^* \omega_1'$, and $\omega_2 \rho^* \omega_2'$, then $\omega_2' \cdot \omega_1' \in W$ and $\omega_2 \cdot \omega_1 \rho^* \omega_2' \cdot \omega_1'$. Furthermore W is pure in the sense that if $\omega_2 \cdot \omega_1$ belongs to W then so ω_2 and ω_1 do. Beyond the mere statement of Conjecture 8.7.5, the basic ginzu partition of A (*cf.* Definition 6.4.23) is a natural candidate for $\mathcal{P}$, this question was the subject of the master thesis of Quentin Plazar (2015). Yet there may be admissible partitions $\mathcal{P}$ that do not derive from a ginzu partition (*e.g.* Figure 8.4 and Figure 8.5). Conjecture 8.7.5 should also be related to the notion of **automatic groups** (*i.e.* a set S that generates the group together with an automaton which recognizes exactly the words over S whose composite is the neutral element of the group – see Epstein (1992)).

Example 8.7.6. The punctured torus is the local pospace $\mathbb{S}^1 \times \mathbb{S}^1 \setminus \{(1, 1)\}$ or

$$\left\{(z_1, z_2) \in \mathbb{C} \times \mathbb{C} \,\middle|\, |z_1| = |z_2| = 1\right\} \setminus \{(1, 1)\}$$

[1] See the first chapter of Epstein (1992), Lawson (2004), Wang (2012) or any textbook on the subject.

using the complex number notation (*cf.* Example 5.4.3). Its components are expected to be $A = \{1\}\times\mathbb{S}^1\setminus\{(1,1)\}$, $B = \mathbb{S}^1\times\{1\}\setminus\{(1,1)\}$, and $C = \mathbb{S}^1\times\mathbb{S}^1\setminus(\{1\}\times\mathbb{S}^1\cup\mathbb{S}^1\times\{1\})$, while its category of components is expected to be freely generated by the graph

$$\begin{tikzcd} A \arrow[r, shift left] & C \arrow[l, shift left] \arrow[r, shift left] & B \arrow[l, shift left] \end{tikzcd}$$

This should be compared to Section 5.5 and the fundamental group of the underlying space (*i.e.* $\mathbb{Z} \oplus \mathbb{Z}$ in the category of groups since the punctured torus is homotopy equivalent to two circles sharing a single point.)

9

Unique Decomposition Theorems

It was known by the ancient Greeks that any integer can be written as a product of prime numbers in a unique way. In modern algebra a **unique factorization domain** or **UFD** for short – Hungerford (2003), is a ring whose elements satisfy this property. Such a ring is also said to be **factorial** – Lang (2002). On the computer science side, a **process algebra** is a set of terms built over an infinite set of variables and the following binary operators:

- sequential composition $P; Q$ which mean that Q is performed once P is over,
- branching $P + Q$ which means that either P or Q is executed, but which one cannot be predicted a priori, and
- parallel composition $P \| Q$ which means that both P and Q run simultaneously.

These terms are subjected to a set of rewriting rules which provide each operator with a behaviour that fits with the intuition. For example both branching and parallel composition are supposed to be commutative while the sequential one should not. Also sequential composition distributes over branching but not over parallel composition.

Examples of such process algebras are the Calculus of Communicating Systems or CCS (*cf.* Milner (1989)), the π-Calculus (*cf.* Milner (1999)), the Communicating Sequential Processes or CSP (*cf.* Hoare (1978, 1985)), the Join-calculus (*cf.* Fournet and Gonthier (2000)) which is based on the chemical abstract machine – Berry and Boudol (1990, 1992) and actually implemented in the JoCaml compiler). An element of a process algebra is just called a **process**. A process that can be written without the parallel composition operator is called a **sequential process**.

The ring of polynomials over some factorial ring is factorial. This well known fact provides a theoretical result of existence while numerous algorithms actually performing the decomposition are known, each of them taking advantage of any extra information about the coefficient ring. In the process algebra setting, the decomposition of a program is always understood with respect to parallel composition. Remark indeed that the decomposition of polynomials is to root finding as the decomposition of a program is to parallel computing optimization. The interest in parallel decomposition is thus beyond the theoretic concern, and it is not surprising that it has already been the subject of several publications – Milner and Moller (1993); Luttik (2003); Luttik and Oostrom

(2005); Fröschle and Lasota (2009); Dreier et al. (2013). The continuous model of a program satisfies the following property: if the set of resources occurring in each of the processes $P_1, \ldots, P_n$ are pairwise disjoint, then

$$[\![P_1|\cdots|P_n]\!] \cong [\![P_1]\!] \times \cdots \times [\![P_n]\!]$$

If we are able to decompose any mathematical object arising as the geometric model of some program, then we could try to recover the decomposition of the program from it (at least partial information about it). Results about decomposition of mathematical structures as a direct product of simpler ones abound. One of the most famous facts of this kind is about finitely generated abelian groups – Hungerford (2003): any such group G can be written as a direct sum

$$G \cong \mathbb{Z}_{p_1}^{s_1} \oplus \cdots \oplus \mathbb{Z}_{p_k}^{s_k} \oplus \mathbb{Z}^n$$

with $k \in \mathbb{N}$, $n \in \mathbb{N}$, and $(p_1, s_1), \ldots, (p_k, s_k)$ a sequence of (not necessarily distinct) ordered pairs of integers whose first term is prime and the second one is positive. The decomposition being unique up to reordering. The Remak-Krull-Schmidt theorem is a broad generalization – see Rotman (1994); Grillet (2007): any (not necessarily abelian) group G whose $\subseteq$-sequences of normal subgroups $(H_n)_{n\in\mathbb{Z}}$ are constant beyond some rank (*i.e.* there exists an integer n such that $H_k \cong H_n$ and $H_{-k} \cong H_{-n}$ for all $k \geqslant n$) can be written as an internal finite product of indecomposable subgroups. Moreover given two such decompositions $G_1 \times \cdots \times G_s$ and $H_1 \times \cdots \times H_t$ one has $s = t$ and there is a reordering such that $G_i \cong H_i$ for all i, and for each $r \leqslant t$

$$G \cong G_1 \times \cdots \times G_r \times H_{r+1} \times \cdots \times H_t$$

The story goes even further. The Ore-Kuroš theorem – see Ore (1936); Grätzer (2003); Blyth (2005), states that in a **modular** lattice L if $a = x_1 \vee \cdots \vee x_n$ and $a = y_1 \vee \cdots \vee y_m$ are irredundant joins of join-irreducible elements then $n = m$ and for all x_i there exists y_j such that

$$a = x_0 \vee \cdots \vee x_{i-1} \vee y_j \vee x_{i+1} \vee \cdots \vee x_{n-1}$$

The Remak-Krull-Schmidt theorem can then be deduced from the Ore-Kuroš one as the lattice of normal subgroups of a group is known to be modular (Kuroš, 1956, p.92).

The standard notions of prime element and irreducible element in a commutative monoid are recalled in Section 9.1. They are used to characterize the free commutative monoids (*cf.* Proposition 9.1.5 and Corollary 9.1.16). Homogeneous monoids are described in Section 9.2. They are based on the action of symmetric groups on homogeneous languages (*cf.* Definition 6.3.1) and proven to be free (*cf.* Theorem 9.2.5). In Section 9.3, we prove that the commutative monoid of regions (*cf.* Definition 9.3.1) is isomorphic to a pure submonoid of some homogeneous monoid, therefore it is free. A very efficient factoring algorithm, discovered by Nicolas Ninin, is explained (*cf.* Theorem 9.3.4). In Section 9.4, we examine how the category of components construction could relate the free commutative monoid of regions to that of nonempty connected loop-free categories (*cf.* Conjecture 9.4.13). In Section 9.5, we introduce the tensor product of Boolean algebras in the category of semilattices with zero (*cf.* Proposition 9.5.25). Then we prove that the map sending any region to the Boolean algebra of its subregions turns a product into a tensor product. The content of Section 9.6 is exploratory, it is concerned with prime decomposition of regions endowed with a metric space structure (*cf.* Section 6.5).

9.1 Prime vs Irreducible

The existence of unique decompositions is related to the subtle distinction between prime and irreducible elements in a commutative monoid. In this section we denote the neutral element of a monoid by ε. More details can be found in the first chapter of Geroldinger and Halter-Koch (2006).

Definition 9.1.1. A **unit** of a commutative monoid is an element u to which corresponds an element u' such that $uu' = \varepsilon$. One says that d **divides** x when there exists x' such that $x = dx'$, this situation being denoted by $d|x$. The elements x and y are said to be **equivalent** when $y = ux$ for some unit u.

Definition 9.1.2. A nonunit element is said to be **irreducible** when it can only be divided, up to equivalence, by ε and itself. A nonunit element is said to be **prime** when it divides a or b as soon as it divides their product. Denote by $I(M)$ and $P(M)$ the set of irreducible elements and the set of prime elements of a commutative monoid M.

Example 9.1.3. In the monoid $(\mathbb{N} - \{0\}, \times, 1)$ it is well known that an integer is prime iff it is irreducible.

Example 9.1.4. Define the **support** of a mapping from X to $\mathbb{N}$ as the subset of X on which it is nonzero. The collection of all the mappings with finite support is denoted by $F(X)$. It becomes a commutative monoid when endowed with the pointwise addition, the null mapping being the neutral element. This construction extends to a functor F : **Set** $\longrightarrow$ **CMon** which is left adjoint to the forgetful one. A commutative monoid is said to be **free** when it is isomorphic to $F(X)$ for some set X. We say that $G \subseteq M$ generates M when any element of M is, up to equivalence, a product of elements of G. For example $F(X)$ is generated by the mappings $g_x : X \longrightarrow \mathbb{N}$ defined by $g_x(y) = 1$ if $x = y$; 0 otherwise.

Proposition 9.1.5.
A commutative monoid M is free iff $P(M) = I(M)$ and generates M.

Example 9.1.6. The monoids $(\mathbb{N}, +, 0)$ and $(\mathbb{N}\backslash\{0\}, \times, 1)$ are freely commutative.

Example 9.1.7.
The commutative monoid $(\mathbb{R}_+, +, 0)$ has neither prime nor irreducible element.

Example 9.1.8. Due to idempotency, a semilattice has no irreducible element. In particular any nonbottom element of a lower bounded chain is a reducible prime.

Example 9.1.9. In $\mathbb{Z}_6$ one can remark that 2 is a reducible prime since $2 = 2 \cdot 4$ (mod 6) and neither 2 nor 4 are unit as they are zero divisors – see (Hungerford, 2003, p.136).

Example 9.1.10.
In the subring $\{a + b\sqrt{10} \mid a, b \in \mathbb{Z}\}$ of $\mathbb{R}$ the elements 2, 3 and $4 \pm \sqrt{10}$ are nonprime irreducible – see (Hungerford, 2003, p.140) or (Bressoud, 1989, p.1). This example is the prototype of a situation which is thoroughly explained in (Weintraub (2008)).

Example 9.1.11. In the semiring $\mathbb{N}[X]$ of polynomials with coefficient in $\mathbb{N}$, one sees that Hashimoto's polynomial $X^5 + X^4 + X^3 + X^2 + X + 1$ has two noncompatible decompositions

$$(X + 1)(X^4 + X^2 + 1) = (X^3 + 1)(X^2 + X + 1)$$

while the following ones hold in $\mathbb{Z}[X]$

$$X^3 + 1 = (X + 1)(X^2 - X + 1) \qquad X^4 + X^2 + 1 = (X^2 + X + 1)(X^2 - X + 1) \, .$$

The polynomials $X^3 + 1$ and $X^4 + X^2 + 1$ are therefore nonprime irreducible – see Nakayama and Hashimoto (1950).

Example 9.1.12. In differential geometry, the compact, connected, smooth oriented n-dimensional manifolds without boundary equipped with the connected sum # (tom Dieck, 2008, p.390) form a commutative monoid $\mathcal{M}_n$ whose neutral element is the n-sphere. It is well-known that $\mathcal{M}_2$ is generated by torus T^2 (Massey, 1991, Chap.1). A more involved result is that $\mathcal{M}_3$ is freely generated by countably many elements – see (Hempel, 1976, Chap.3) or (Jaco, 1980, Chap.2). The existence of the decomposition is due to Kneser (1929) and its uniqueness to Milnor (1962).

Definition 9.1.13. A commutative monoid M is said to be **graded** when there exists a morphism of monoid $d : M \to \mathbb{N}$ such that

$$d^{-1}(\{0\}) = \{\text{units of } M\}$$

The morphism d is called a **length function** in (Anderson, 1997, p.8).

Remark 9.1.14. A free commutative monoid is graded since each of its elements can be associated with the number of terms of its prime decomposition. The graded monoids are actually not far from being free.

Proposition 9.1.15. Given a graded monoid M, $I(M)$ generates M and contains $P(M)$.

Proof. Given $x_1, \ldots, x_n$ nonunit elements of M we have

$$d(x_1 \cdots x_n) = d(x_1) + \cdots + d(x_n) \geqslant n$$

because M is graded. It is therefore generated by its irreducible elements. Suppose that $p = a \cdot b$ is prime. So we can suppose p divides a, and then

$$d(a) + d(b) = d(a \cdot b) = d(p) \leqslant d(a)$$

Therefore $d(b) = 0$ from which we deduce that b is a unit element of M which is graded. □

Corollary 9.1.16. A commutative monoid M is free iff it is graded and $I(M) \subseteq P(M)$.

Note that in Definition 9.1.13 we can actually replace $\mathbb{N}$ by any free commutative monoid. Yet another characterization of free commutative monoids can be found in the introduction of Luttik and Oostrom (2005).

Any submonoid of a graded monoid is graded. Yet a submonoid of a free commutative monoid might not be free e.g. define $\alpha = x + 2y$, $\beta = 2x + y$, and $\gamma = x + y$ so the submonoid of $\mathbb{N}^{\{x,y\}}$ generated by α, β, and γ satisfies $\alpha + \beta = 3\gamma$.

Definition 9.1.17. A submonoid P of M is said to be **pure** when for all $x, y \in M$, if $x \cdot y \in P$ then both x and y belongs to P.

Definition 9.1.17 is actually a special case of Definition 8.2.8, in the current context it is motivated by the following result:

Lemma 9.1.18. Any pure submonoid of a free commutative monoid is free.

Proof. Let M be a free commutative monoid and P be one of its submonoids. Then P is graded. Moreover if p is irreducible in P and p divides $x \cdot y$ with $x, y \in P$ then we can suppose that $x = p \cdot x'$ with $x' \in M$. Since P is pure, x' actually belongs to P. We conclude by Corollary 9.1.16. □

9.2 Action of the Symmetric Groups on the Homogeneous Languages

This section provides a nontrivial example of free commutative monoid which is based on homogeneous languages (*cf.* Definition 6.3.1) and can be seen as the sequel of Section 6.3.

Definition 9.2.1. A **subword** of a word w is a word of the form $w \circ \phi$ where ϕ is a strictly increasing map $\{1, \ldots, n\} \to \{1, \ldots, \ell(w)\}$. Such a map ϕ is entirely characterized by its image A, hence it makes sense to write $w|_A$ (instead of $w \circ \phi$) for any $A \subseteq \{1, \ldots, \ell(w)\}$. By extension we define the **subword language** of a homogeneous language D applying the precomposition $\_ \circ \phi$ to all its words, and denote it by $D|_A$.

It comes with a factorization algorithm whose complexity is exponential in the length of its elements (*cf.* Definition 6.3.1). In practice, if the prime factors of some element have "small" lengths, then the decomposition algorithm quickly finds them. This is to be compared with the naive decomposition algorithm of natural numbers, which is fast provided the prime factors are "small".

The n^{th} **symmetric group** $\mathfrak{S}_n$, whose elements are the permutations of the set $\{1, ..., n\}$, acts on the words of length n by composing on the right, that is for all $\sigma \in \mathfrak{S}_n$ and all words w of length n we have

$$\sigma \cdot w := w \circ \sigma = (w_{\sigma(1)} \cdots w_{\sigma(n)})$$

By extension $\mathfrak{S}_n$ also acts on $\mathcal{D}_n(\mathbb{A})$ (*cf.* Definition 6.3.1) by applying the same permutation to all the words of a homogeneous language D of length n:

$$\sigma \cdot D := \{\sigma \cdot w \mid w \in D\}$$

Two homogeneous languages are said to be equivalent, denoted by $D \sim D'$, when $\ell(D) = \ell(D') = n$ and there exists $\sigma \in \mathfrak{S}_n$ such that $D' = \sigma \cdot D$. The **juxtaposition** $\sigma \otimes \sigma' \in \mathfrak{S}_{n+n'}$ of two permutations $\sigma \in \mathfrak{S}_n$ and $\sigma' \in \mathfrak{S}_{n'}$ is defined as:

$$\sigma \otimes \sigma'(k) := \left\{ \begin{array}{ccc} \sigma(k) & \text{if} & 1 \leqslant k \leqslant n \\ \big(\sigma'(k-n)\big) + n' & \text{if} & n+1 \leqslant k \leqslant n+n' \end{array} \right.$$

A Godement-like exchange law is satisfied, which ensures that $\sim$ is actually a congruence over $\mathcal{D}_h(\mathbb{A})$:

$$(\sigma \cdot S) * (\sigma' \cdot S') = (\sigma \otimes \sigma') \cdot (S * S')$$

Definition 9.2.2. The **homogeneous monoid** over $\mathbb{A}$, denoted by $\mathcal{H}(\mathbb{A})$, is the quotient $\mathcal{D}_h(\mathbb{A})/\!\sim$ from which the absorbing element has been removed. Moreover the homogeneous monoid is commutative and its only unit is the singleton $\{\varepsilon\}$.

Remark 9.2.3. If the alphabet $\mathbb{A}$ is a singleton (resp. the empty set) then the homogeneous monoid $\mathcal{H}(\mathbb{A})$ is isomorphic to $(\mathbb{N}, +, 0)$ (resp. the null monoid).

Remark 9.2.4. If $\mathbb{A}$ contains at least two elements, then $\mathcal{H}(\mathbb{A})$ is not isomorphic to the abelianization of $\mathcal{D}_h(\mathbb{A})$. Indeed $\{aaa, aab, baa\}$ is irreducible in the latter, not in the former since it is identified with $\{aaa, aba, baa\} = \{aa, ab, ba\} * \{a\}$.

As they are $\sim$-equivalence classes, the elements of $\mathcal{H}(\mathbb{A})$ are subsets of $\mathcal{D}_h(\mathbb{A})$. The notion of length extends to all $H \in \mathcal{H}(\mathbb{A})$ defining $\ell(H) := \ell(D)$ for any $D \in H$. Hence $\mathcal{H}(\mathbb{A})$ is graded.

Theorem 9.2.5 (Balabonski and Haucourt (2010))**.**
The commutative monoid $\mathcal{H}(\mathbb{A})$ is free.

Proof. Suppose that H is an element of $\mathcal{H}(\mathbb{A})$ which divides $H_1 * H_2$ and pick D, D_1 and D_2 respectively from the equivalence classes H, H_1 and H_2. Define $n = \ell(D)$, $n_1 = \ell(D_1)$ and $n_2 = \ell(D_2)$, and remark that $n \leqslant n_1 + n_2$. There exists $\sigma \in \mathfrak{S}_n$ and some D_3 such that $\sigma \cdot (D_1 * D_2) = D * D_3$ in $\mathcal{D}_h(\mathbb{A})$. Suppose in addition that H does not divide H_1 nor H_2 (therefore H is not prime), then we have $A_1 \subseteq \{1, ..., n_1\}$ and $A_2 \subseteq \{1, ..., n_2\}$ such that $A_1 \neq \emptyset$, $A_2 \neq \emptyset$, and $\sigma(A_1 \cup A'_2) = \{1, ..., n\}$ where $A'_2 := \{a + n_1 \mid a \in A_2\}$. Then we have a nontrivial factorization $D = D'_1 * D'_2$ where D'_i is the subword language $D|_{A_i}$ for $i \in \{1, 2\}$. Thus H is not irreducible. We conclude by Corollary 9.1.16. □

By definition an element of $\mathcal{H}(\mathbb{A})$ is an equivalence class whose elements are sets of the same cardinal. Therefore we can define the cardinal of an element of $\mathcal{H}(\mathbb{A})$ as the cardinal of any of its elements. In particular an element of $\mathcal{H}(\mathbb{A})$ is said to be finite when so is its cardinal. Therefore we denote by $\mathcal{H}_f(\mathbb{A})$ the collection of finite elements of $\mathcal{H}(\mathbb{A})$.

Lemma 9.2.6. The collection $\mathcal{H}_f(\mathbb{A})$ is a pure submonoid of $\mathcal{H}(\mathbb{A})$.

Proof. The nonempty languages S and S' are finite iff $S \cdot S'$ is so. □

Corollary 9.2.7. The commutative monoid $\mathcal{H}_f(\mathbb{A})$ is free.

Proof. Readily comes from Theorem 9.2.5, Lemma 9.2.6 and Lemma 9.1.18. □

Remark 9.2.8. Finding a factor of $H \in \mathcal{H}_n(\mathbb{A})$ amounts to finding a representative of H that can be written as $D_0 \cdot D_1$ in $\mathcal{D}_h(\mathbb{A})$ (*cf.* Definition 6.3.1). In other words, we can fix some representative D of H and seek for a permutation σ of $\{1, \ldots, n\}$ such that $\sigma \cdot D$ factors in $\mathcal{D}_{nf}(\mathbb{A})$. In particular there is a subset $A \subseteq \{1, \ldots, n\}$ of cardinal $n' = \ell(D_0)$ such that σ sends A to $\{1, \ldots, n'\}$ and the complement of A to $\{n'+1, \ldots, n\}$. Actually we can even suppose that the restrictions of σ to A and to its complement are order-preserving in order to make it entirely defined by A. The factoring algorithm thus requires to test $\lceil n/2 \rceil$ subsets of $\{1, \ldots, n\}$[1].

We shall see that the preorder relation introduced in Definition 6.2.1 can actually be transferred to both $\mathcal{H}(\mathbb{A})$ and $\mathcal{H}_f(\mathbb{A})$ giving rise to an isomorphism analogous to that of Corollary 6.3.3.

Definition 9.2.9. A binary relation $\diamond$ over $\mathcal{D}_h(\mathbb{A})$ is said to be **admissible** when it is compatible with the product (of $\mathcal{D}_h(\mathbb{A})$) and satisfies

$$\forall D, D' \in \mathcal{D}_h(\mathbb{A}) \, \big(D \diamond D' \;\Rightarrow\; \ell(D) = \ell(D') \text{ and } \forall \sigma \in \mathfrak{S}_{\ell(D)} \, (\sigma \cdot D) \diamond (\sigma \cdot D')\big)$$

[1] $\lceil n/2 \rceil$ denotes the least integer greater or equal than $n/2$.

A useful feature of admissible relations is given by

Lemma 9.2.10. Any admissible relation over $\mathcal{D}_h(\mathbb{A})$ can be transferred to a relation on $\mathcal{H}(\mathbb{A})$ which is still compatible with the product.

Proof. It suffices to set $H \diamond H'$ when $\ell(H) = \ell(H')$ and there exists $D \in H$ and $D' \in H'$ such that $(\sigma \cdot D) \diamond (\sigma \cdot D')$ hold for all $\sigma \in \mathfrak{S}_{\ell(H)}$. □

The quotient map $\mathcal{D}_h(\mathbb{A}) \to \mathcal{H}(\mathbb{A})$ is then compatible with $\diamond$ and its extension. The next result provides a natural example of such a relation.

Lemma 9.2.11. Let $\preccurlyeq_{\mathbb{A}}$ be a preorder on the alphabet $\mathbb{A}$ with a least element $\bot_{\mathbb{A}}$. The preorder $\preccurlyeq_h$ defined over $\mathcal{D}_h(\mathbb{A})$ by

$$D \preccurlyeq_h D' \;\equiv\; \forall w \in D \,(\exists i \in \{1, \ldots, \ell(D)\},\; w(i) \sim_{\mathbb{A}} \bot_{\mathbb{A}} \;\text{ or }\; \exists w' \in D',\; w \preccurlyeq_{\mathbb{A}}^{\ell(D)} w')$$

is admissible. Moreover the relation $\preccurlyeq$ induced on $\mathcal{H}(\mathbb{A})$ is also a preorder with a least element. In particular, $H, H' \in \mathcal{H}(\mathbb{A})$ are equivalent iff there exist $D \in H$ and $D' \in H'$ such that for all $\sigma \in \mathfrak{S}_{\ell(H)}$, $\sigma \cdot D$ and $\sigma \cdot D'$ are equivalent.

Proof. The relation $\preccurlyeq_h$ is admissible by construction and the relation $\preccurlyeq$ is obviously reflexive. If $H_1 \preccurlyeq H_2 \preccurlyeq H_3$ then we have $D_1 \in H_1$, $D_2, D_2' \in H_2$, and $D_3' \in H_3$ such that

$$D_1 \preccurlyeq_h D_2 \;\text{ and }\; D_2' \preccurlyeq_h D_3'$$

then we have some permutation τ such that $\tau \cdot D_2' = D_2$. Therefore setting $D_3 = \tau \cdot D_3'$ we have for all $\sigma \in \mathfrak{S}_{d(H)}$

$$\sigma \cdot D_1 \preccurlyeq_h \sigma \cdot D_2 \preccurlyeq_h \sigma \cdot D_3$$

In particular given $w \in D_1$ no term of which is (equivalent to) $\bot_{\mathcal{A}}$, we have $w' \in D_2$, and $w'' \in D_3$ satisfying $w \preccurlyeq_{\mathbb{A}} w' \preccurlyeq_{\mathbb{A}} w''$, and thus $H_1 \preccurlyeq H_3$. Now suppose that $H \preccurlyeq H'$ and $H' \preccurlyeq H$. Then we have $D \in H$, $D' \in H'$ and some permutation τ such that $D \preccurlyeq_h D' \preccurlyeq_h \tau \cdot D$. So we obtain inductively for $m \in \mathbb{N}$

$$D \preccurlyeq_h \tau \cdot D \preccurlyeq_h \cdots \preccurlyeq_h \tau^m \cdot D \preccurlyeq_h \cdots$$

By finiteness of the group $\mathfrak{S}_{\ell(H)}$ we have $\tau^m = \mathrm{id}$ for some $m \in \mathbb{N} \setminus \{0\}$ so D and D' are equivalent. □

Corollary 9.2.12. The free commutative monoids $\mathcal{H}(\mathbb{A})$ and $\mathcal{H}_f(\mathbb{A})$ inherits their preorders from the preorder on the alphabet $\mathbb{A}$.

Remark 9.2.13. Defining $\preccurlyeq_h$ by

$$D \preccurlyeq_h D' \;\equiv\; \forall w \in D \,\exists w' \in D',\; w \preccurlyeq_{\mathbb{A}}^{\ell(D)} w'$$

would have also provided an admissible relation. In the former case however, two words containing $\bot_{\mathbb{A}}$ are equivalent though they may not be in the latter. The definition given in Lemma 9.2.11 is thus better fitted with the situation where $\mathbb{A}$ is a (fine) connectology over $\mathcal{X}$ ordered by inclusion, for it sees any word w containing $\bot_{\mathbb{A}}$ as a representative of the empty subset of $\mathcal{X}^{\ell(w)}$.

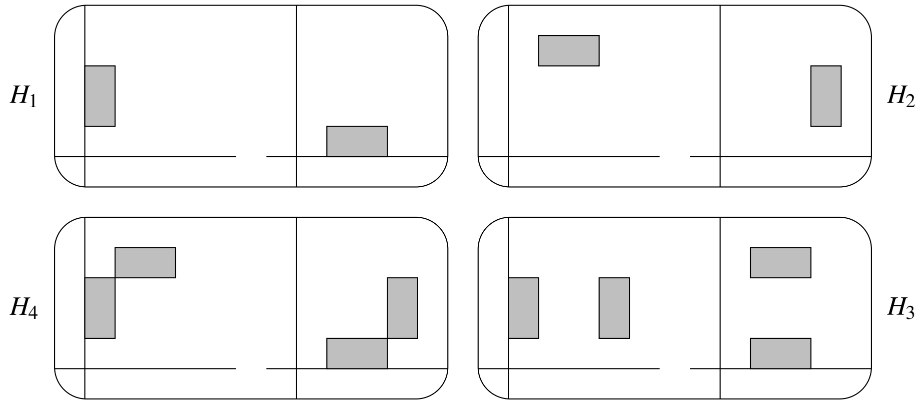


Figure 9.1: Two elements of $\mathcal{H}_f(\mathbb{A})$ with two non comparable minimal upper bounds

Remark 9.2.14. The preorder induced on $\mathcal{H}_f(\mathbb{A})$ by Lemma 9.2.11 does not, in general, inherits the properties from $\preccurlyeq_{\mathbb{A}}$. For example let $\mathbb{A}$ be the collection of finite unions of subintervals of $\mathbb{R}$ ordered by inclusion (*cf.* Example 6.2.20). The frames on Figure 9.1 are extensive descriptions of elements of $\mathcal{H}_f(\mathbb{A})$, which are denoted by H_1, H_2, H_3 and H_4. Formally we have $H_1 = \{[0,1] \times [1,3], [1,3] \times [0,1]\}$ and $H_2 = \{[1,3] \times [3,4], [3,4] \times [1,3]\}$. Then observe that H_3 and H_4 are non comparable minimal upper bounds of H_1 and H_2:

$$H_3 = \{[0,1] \times [1,3] \cup [3,4] \times [1,3], [1,3] \times [0,1] \cup [1,3] \times [3,4]\}$$

and

$$H_4 = \{[0,1] \times [1,3] \cup [1,3] \times [3,4], [1,3] \times [0,1] \cup [3,4] \times [1,3]\}$$

The resulting preorder on $\mathcal{H}_f(\mathbb{A})$ is not even a $\vee$-lattice. The structures provided by Theorem 6.2.21 are thus lost.

As a more degenerated case, let $\mathbb{A}$ be $\{a, b, a', b'\}$ with the discrete order. Then consider $D = \{ab\}$ and $D' = \{a'b'\}$, and let $[D]$ and $[D']$ be their corresponding equivalence classes under the action of the symmetric group. The classes of $\{ab, a'b'\}$ and $\{ab, b'a'\}$ are minimal upper bounds of $\{[D], [D']\}$ which are not comparable. Once again, the structure carried by $\mathcal{H}(\mathbb{A})$ is thus not even a $\vee$-lattice.

9.3 Isothetic Regions

We have introduced three notions of independence for Paml programs: syntactic independence (*cf.* Definition 1.5.3), observational independence (*cf.* Definition 1.5.7), and model independence (*cf.* Definition 7.3.1). They are related by the following chain of implications, each of which being strict (*cf.* Proposition 7.3.2, Theorem 7.3.4, and Example 7.3.3).

$$\text{syntactically indep.} \quad \Rightarrow \quad \text{model indep.} \quad \Rightarrow \quad \text{observationally indep.}$$

On one hand the leftmost notion is too strong because it is not even able to detect obviously artificial dependencies. On the other hand the rightmost one is so weak that it lets certain Paml programs be independent from themselves, plus it cannot be decided at compile time. The remaining one is related to the existence of decompositions of the continuous model as Cartesian product of continuous models: the Paml programs $P_1, \ldots, P_n$ are model independent when

$$[\![P_1|\cdots|P_n]\!] = [\![P_1]\!] \times \cdots \times [\![P_n]\!]$$

Since the continuous models of Paml programs are isothetic regions (*cf.* Proposition 7.1.5), they are the prime motivation for studying homogeneous monoids (*cf.* Definition 9.2.2). In this particular case, $\mathbb{A}$ is a (fine) connectology over $\mathcal{X}$ (*cf.* Definition 6.2.8, Definition 6.2.19, and Example 6.2.25) ordered by inclusion. We turn the monoid of regions $\mathcal{R}$ (*cf.* Definition 6.3.2) into a commutative one.

Definition 9.3.1. The **commutative monoid of regions** is the quotient of $\mathcal{R}/\mathfrak{S}$ i.e. the disjoint union of quotients

$$\bigsqcup_{n\in\mathbb{N}} \mathcal{R}_n/\mathfrak{S}_n$$

Theorem 9.3.2. The pair (α, γ) of Corollary 6.3.3 induces morphisms of (pre)ordered commutative monoids which becomes isomorphisms if one restricts to the image of α (*i.e.* equivalence classes of maximal block coverings.)

$$\begin{tikzcd} \mathcal{H}(\mathbb{A}) \arrow[r, shift left, "\gamma"] & \mathcal{H}(\mathcal{X}) \arrow[l, shift left, "\alpha"] \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} \mathcal{H}_f(\mathbb{A}) \arrow[r, shift left, "\gamma"] & \mathcal{R}/\mathfrak{S} \arrow[l, shift left, "\alpha"] \end{tikzcd}$$

Proof. The action of the symmetric groups on the homogeneous languages is compatible with the (pre)ordered monoid morphisms (α, γ) of in the sense that for all $\sigma \in \mathfrak{S}_n$, for all $D \in \mathcal{D}_n(\mathbb{A})$ (resp. $\mathcal{D}_{nf}(\mathbb{A})$), and for all $X \in \mathcal{D}_n(\mathcal{X})$ (resp. $\mathcal{D}_{nf}(\mathcal{X})$), we have

$$\sigma \cdot \gamma_n(D) = \gamma_n(\sigma \cdot D) \quad \text{and} \quad \sigma \cdot \alpha_n(X) = \alpha_n(\sigma \cdot X)$$

□

Remark 9.3.3. At first sight, the prime decomposition of a continuous model given by Corollary 9.2.7 and Theorem 9.3.2 does not directly provide a family of model independent Paml programs. In practice, the algorithm performing that decomposition keeps track of the correspondence between positions of letters in words and process identifiers.

Exploiting the extra assumption that $\mathbb{A}$ is the Boolean algebra associated with a fine connectology, Nicolas Ninin (2016) discovered an algorithm that is incomparably more efficient than the one described in Remark 9.2.8. Given $H \in \mathcal{H}_n(\mathbb{A})$ (for $n \in \mathbb{N}$) let $\mathcal{F}$ be a collection of blocks (*cf.* Definition 6.2.12) whose union is the complement (*cf.* Theorem 6.2.21) of some representative of H. The elements of $\mathcal{F}$ are words of length n over $\mathbb{A}$ so we denote by proj_k the operator that returns the k^{th} letter of any such word. Then let $\sim$ be the equivalence relation generated by the following binary relation over $\{1, \ldots, n\}$

$$\big\{(i,j) \,\big|\, \exists B \in \mathcal{F} \text{ s.t. neither } \mathrm{proj}_i(B) \text{ nor } \mathrm{proj}_j(B) \text{ are the maximum element of } \mathbb{A}\big\}$$

Theorem 9.3.4 (Ninin (2016))**.** The $\sim$-equivalence classes provide a decomposition of H. Moreover, if $\mathcal{F}$ is the collection of maximal blocks of the chosen representative of H, then it is the prime decomposition of H.

Remark 9.3.5. In practice, the raw output of the ALCOOL software is the complement of the continuous model of the program to analyze given as a finite collection $\mathcal{F}$ of blocks of $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$. So we have to be careful applying Theorem 9.3.4 which requires to work in $\rceil G \lfloor^n$ for some finite graph G. In that case $\mathbb{A}$ is the Boolean algebra associated with the fine connectology of $\rceil G \lfloor$ and the maximum of $\mathbb{A}$ is $\rceil G \lfloor$ itself. One can circumvent the problem either by:

- defining G as the disjoint union of the graphs $G_1, \ldots, G_n$ and turning each block $B_1 \times \cdots \times B_n \in \mathcal{F}$ into $B'_1 \times \cdots \times B'_n$ with
$$B'_k \quad = \quad B_k \cup \bigcup_{i \neq k} \rceil G_i \lfloor\,,$$
- or keeping track of the fact that complement has been computed in the product $\rceil G_1 \lfloor \times \cdots \times \rceil G_n \lfloor$ and thus testing whether $\mathrm{proj}_i(M)$ is the maximum of $\mathbb{A}_i$ (*i.e.* the Boolean algebra associated with the fine connectology of $\rceil G_i \lfloor$, the maximum of $\mathbb{A}_i$ being $\rceil G_i \lfloor$ itself) instead of $\mathbb{A}$.

Example 9.3.6. The n-dining philosophers case ($n \in \mathbb{N}$) strikingly illustrates the efficiency of the factoring method derived from Theorem 9.3.4. The corresponding Paml program declares n mutices and n processes (*i.e.* the “philosophers”)

$$\texttt{proc:}\ \ \texttt{p}_i\ \texttt{=}\ \texttt{P(a}_i\texttt{);P(a}_{i+1}\texttt{);V(a}_i\texttt{);V(a}_{i+1}\texttt{)}$$

with the indices being taken modulo n. The complement of the continuous model of the program (*cf.* Definition 7.1.2) has exactly n maximal blocks $M_0, \ldots, M_{n-1}$, and for $k \in \{0, \ldots, n-1\}$,

$$\mathrm{proj}_k(M_i) = \begin{cases} [1,4[& \text{if } k = i\ ; \\ [2,3[& \text{if } k = i+1 \mod n\ ; \\ \text{the maximal element of } \mathbb{A} & \text{otherwise.} \end{cases}$$

from which one deduces, by Theorem 9.3.4, that the continuous model of the program is prime in $\mathcal{H}_f(\mathbb{A})$.

Example 9.3.7. Definition 7.3.1 and Theorem 9.3.4 should be understood as the refinement that fills the gap between Definition 1.5.3 and Definition 1.5.7 by detecting when a semaphore is “artificially” involved in a program. Let us go back to Example 7.3.3 to illustrate the last remark. The raw output of the ALCOOL software is

$$\{BBCC, CCBB, AAAC, AACA, ACAA, CAAA\}$$

with $A = [2,3[$, $B = [1,4[$, and $C = [0,5]$ (the maximum of $\mathbb{A}$). Hence $A \subseteq B \subseteq C$ and the preceding language is equivalent to $\{BBCC, CCBB\}$. Applying Theorem 9.3.4 we obtain that both programs of Example 7.3.3 are model independent. Note that the forbidden regions F_a, F_b, and F_c generated by a, b, and c are respectively $\{BBCC\}$, $\{CCBB\}$, and $\{AAAC, AACA, ACAA, CAAA\}$. In particular $F_c \subseteq F_a \cup F_b$, which reflects the “handcrafted” analysis made in Example 7.3.3.

Beyond its combinatorial nature, a region is endowed with a topology, several metrics inducing its topology, several measures (related to its metrics), and it is associated with invariants (*e.g.* its category of components). Each of the aforementioned structures lies in a class of mathematical objects that may satisfy a unique decomposition theorem. When it is the case, we would like to determine whether both decompositions match. When the region of interest arises as the model of some Paml program P, the underlying idea is that the prime decompositions in all of these classes should be strongly related to the process decomposition of P. The subsequent sections are dedicated to this question.

9.4 Finite Connected Loop-Free Categories

From the basic facts of category theory given therein after:
– $\mathcal{A} \cong \mathcal{A}'$ and $\mathcal{B} \cong \mathcal{B}'$ implies $\mathcal{A} \times \mathcal{B} \cong \mathcal{A}' \times \mathcal{B}'$,
– $\mathcal{A} \times (\mathcal{B} \times C) \cong (\mathcal{A} \times \mathcal{B}) \times C$,
– $\mathcal{A} \times \mathcal{B} \cong \mathcal{B} \times \mathcal{A}$, and
– $\mathcal{A} \times \mathbb{1} \cong \mathcal{A}$ with $\mathbb{1}$ a category with one object an one morphism,
one deduces that the collection of isomorphism classes of small categories can be given a structure of commutative monoid (provided one relaxes the condition that the underlying collection of a monoid should be a set). One readily checks that the collection $\mathbb{M}$ of isomorphism classes of nonempty finite connected loop-free categories (which is a countable set indeed) is a pure submonoid of the previous one. Then $\mathbb{M}$ is graded by the mapping that sends any element of $\mathbb{M}$ to its number of morphisms

$$\mathrm{Card}(\mathrm{Mo}(\mathcal{A} \times \mathcal{B})) = \mathrm{Card}(\mathrm{Mo}(\mathcal{A})) \times \mathrm{Card}(\mathrm{Mo}(\mathcal{B}))$$

In particular there are countably many isomorphism classes of such categories since for any of them to be prime, it suffices that its number of morphisms be prime. Thibaut Balabonski (2007) proved the following result during an internship I supervised.

Theorem 9.4.1. Any irreducible element of $\mathbb{M}$ is prime.

As a straightforward consequence of Theorem 9.4.1 and Corollary 9.1.16, we have

Corollary 9.4.2. The commutative monoid $\mathbb{M}$ is free.

Lemma 9.4.3, Corollary 9.4.5 and Proposition 9.4.6 are the starting points of the proof of Theorem 9.4.1.

Lemma 9.4.3. Any morphism of a finite loop-free category can be written as a composite of irreducible morphisms.

Proof. A morphism of a loop-free category C is irreducible when it cannot be written as the composite of two nonidentity morphisms. Since C is loop-free any decomposition (without identities) of length n gives rise to a sequence of $n+1$ pairwise distinct objects of C. The finiteness of C thus imposes an upper bound on n. □

A set G of morphisms of C is said to be **generating** when all the nonidentity morphisms of C can be written as a composite of elements of G.

Remark 9.4.4. If the collection of irreducible morphisms of C forms a generating set of morphisms, then it is the least one. Moreover C has neither isomorphisms nor idempotent morphisms but the trivial ones.

Corollary 9.4.5. If C is a finite loop-free category then its set of irreducible morphisms is the least generating one.

For all finite loop-free categories C, we denote by $\mathrm{Irr}(C)$ the graph whose vertices are the objects of C and arrows are the irreducible morphisms of C. Given an equivalence relation on the collection of morphisms, we denote by $\sim^*$ the least congruence containing $\sim$.

Proposition 9.4.6. If C is a finite loop-free category then the collection of equivalence relations $\sim$ such that

$$\mathrm{Free}(\mathrm{Irr}(C))/{\sim^*} \;\cong\; C$$

admits a least element denoted by $\sim_C$.

Definition 9.4.7. A category whose irreducible elements form a generating set of morphisms, and that satisfies Proposition 9.4.6, is said to have a **least presentation**.

Hence any finite loop-free category has a least presentation. The proof of Theorem 9.4.1 is based on the relation between the Cartesian product of finite loop-free categories and the Cartesian product of their least presentations.

Remark 9.4.8. The connectedness hypothesis cannot be dropped. Indeed we obtain a counter-example by interpreting the product and the sum of monomials as the Cartesian product and the coproduct of categories, and then by substituting the category $\{\cdot \to \cdot\}$ to X in Hashimoto's polynomial

$$X^5 + X^4 + X^3 + X^2 + X + 1$$

Remark 9.4.9. The unique decomposition property on $\mathbb{M}$ actually generalizes a result on finite posets – see Hashimoto (1951); Schröder (2002). We then wonder whether it could be deduced from a combination of results from Hashimoto (1951) and Ore (1936). For example the unique factorization theorem for finite groups (a.k.a. Krull-Schmidt theorem) is known to be a consequence of the latter.

Definition 9.4.10. A region A is said to be loop-free (resp. diconnected) when its fundamental category is loop-free (resp. connected).

In this case the category of components of (the fundamental category of) A is finite. Moreover the category of components of a loop-free (resp. connected) category is loop-free (resp. connected). The category of components of a diconnected loop-free region is thus a finite connected loop-free category. One easily checks that a region $A \subseteq \lceil G \rfloor^n$ is loop-free (resp. diconnected) iff for any $\sigma \in \mathfrak{S}_n$, so is $\sigma \cdot A$. Hence it makes sense to write that an element of the commutative monoid of isothetic regions, which is an orbit under the action of $\mathfrak{S}_n$, is loop-free (resp. diconnected), see Section 9.2.

Lemma 9.4.11. A product (in the commutative monoid of G-regions, see Definition 9.3.1) is loop-free (resp. diconnected) iff so are both terms of the product.

Corollary 9.4.12. The commutative monoid of connected loop-free regions is free.

Proof. By Lemma 9.4.11, the connected loop-free elements of the commutative monoid of regions form a pure submonoid of it. We conclude by Lemma 9.1.18. □

Since both fundamental category and category of components constructions preserve binary Cartesian product (*cf.* Lemma 5.2.12, Corollary 8.5.3, and Proposition 8.5.10), we have a morphism of commutative monoids

$$\{\text{connected loop-free regions}\} \xrightarrow{\overrightarrow{\pi_0} \circ \overrightarrow{\pi_1}} \{\text{finite connected loop-free categories}\}$$

As a consequence of the fourth point of Theorem 8.3.10, the category of components of a loop-free category C is reduced to a single morphism iff C is isomorphic to a lattice.

Conjecture 9.4.13. If A is a prime connected loop-free region whose fundamental category is not isomorphic to a lattice, then $\overrightarrow{\pi_0}(\overrightarrow{\pi_1}A)$ is prime.

As we have seen in Section 8.7, the notion of category of components is not well understood in the presence of loops. However we are confident that it can be defined for all fundamental categories (*cf.* Definition 5.2.10) of isothetic regions (*cf.* Definition 6.2.1). In this case, we suspect that the resulting category of components has a least presentation. As the proof of Theorem 9.4.1 heavily relies on the existence of a least presentation, it is natural to define $\mathbb{M}'$ as the commutative monoid of (isomorphism classes of) nonempty finite diconnected categories with a least presentation. One has $\mathbb{M} \subseteq \mathbb{M}'$ by Proposition 9.4.6 so it is natural to ask whether $\mathbb{M}'$ is free and whether decompositions are preserved by the mapping that sends a diconnected region to the category of components of its fundamental category.

9.5 Boolean Algebras

Given an isothetic region A, the collection of isothetic regions contained in A forms a Boolean algebra denoted by $\mathcal{R}_A$ (*cf.* Definition 6.2.1 and Theorem 6.2.21). Note however that if $A \subseteq A'$ then $\mathcal{R}_A$ is not a Boolean subalgebra of $\mathcal{R}_{A'}$ since their greatest elements are respectively A and A'. Except for this glitch, both share their join and meet operators as well as their least element, namely the empty region. We have seen in Definition 7.1.2 that the continuous model of a program is expressed in terms of Boolean operations over regions. Moreover, A being the continuous model of a Paml program P, the deadlock attractor (*cf.* Remark 7.1.3) as well as many other significant pieces of information about P are obtained from Boolean operations in $\mathcal{R}_A$, which is therefore of a great interest. Given isothetic regions $A_1, \ldots, A_n$ the Boolean algebra $\mathcal{R}_{A_1 \times \cdots \times A_n}$ is actually the coproduct of the Boolean algebras $\mathcal{R}_{A_i}$ for $i \in \{1, \ldots, n\}$ (Givant and Halmos, 2009, p.433).

Existence of coproducts of Boolean algebras is given by the Stone representation theorem. The category **Stone** is the full subcategory of **CHaus** whose objects are totally disconnected (*i.e.* the connected components of the objects are singletons). A **Stone space** is an objects of the category **Stone**. The categorical form of the Stone representation theorem provides an isomorphism between the category **BoolAlg** and the dual of the category **Stone** (Johnstone, 1982, p.71).

$$\begin{tikzcd}\mathbf{BoolAlg} \arrow[r, shift left, "\mathcal{S}"] & \mathbf{Stone}^{\mathrm{op}} \arrow[l, shift left, "\mathcal{B}"]\end{tikzcd}$$

The category **Stone** is complete because the products of Stone spaces in **Top** are Stone spaces. This is mainly a consequence of Tychonoff's theorem (Kelley, 1955, p.143). The coproduct $A_1 \sqcup A_2$ of a pair of Boolean algebras is thus $\mathcal{B}(\mathcal{S}(A_1) \times \mathcal{S}(A_2))$ where the product is indifferently taken in the category **Stone**, **CHaus**, or **Top**.

Coproducts of Boolean algebras are given many different names in literature. The most misleading terminology is perhaps the one adopted in Sikorski (1950) where coproducts of Boolean algebras were considered for the first time. In that article, they are indeed called *Cartesian products*. Similarly Sikorski (1969) writes *Boolean products* (resp. *direct unions*) to denote the coproducts (resp. products) of Boolean algebras. The term *free product* is the most common one (Koppelberg, 1989, p.158) due to the link with free Boolean algebras (Givant and Halmos, 2009, Chap. 44). That connection is to be compared to the one between free groups and free products of groups. One also meets the word *sum* which echoes the name given to coproducts of modules in commutative algebra (Givant and Halmos, 2009, Chap.44). From this point of view, Lemma 9.5.1 is the Boolean algebra version of a well-known result of linear algebra. To see this, it suffices to read "vector space" and "sum" instead of "Boolean algebra" and "meet".

Lemma 9.5.1 (see Givant and Halmos (2009), p.431)**.** A Boolean algebra A is the sum of a family of Boolean algebras A_i just in case there are homomorphisms $A_i \longrightarrow A$ such that the union

$$\bigcup_i f_i(A_i)$$

generates A, and whenever J is a finite, non-empty subset of the indices, and p_i is an element in A_i such that $f_i(p_i) \neq 0$, for each $i \in J$, then

$$\bigwedge_{i \in J} f_i(p_i) \neq 0$$

Occasionally one also reads the term *tensor product*, for example in (Pierce (1983)) and (Haucourt and Ninin (2014)), because of the similarity with commutative algebra where the coproduct of a finite family of commutative algebras is its ordinary tensor product (Lang, 2002, Prop.6.1, p.630). Going deeper, the preceding remark has to do with universal algebra, this is precisely the purpose of this section. We also state some conjectures about uniqueness of the coproduct decomposition in the case of Boolean algebras of isothetic regions (*cf.* Conjecture 9.5.28, 9.5.29, and 9.5.30).

Tensor Product of Models of an Algebraic Theory

The concept of algebraic theory is related to universal algebra – see Grätzer (2008); Burris and Sankappanavar (1981), and (Borceux, 1994b, Chap.3). A **signature** is a mapping $\alpha : \Omega \longrightarrow \mathbb{N}$. Each element $\omega \in \Omega$ should be thought of as an operator and $\alpha(\omega)$ as its **arity** (*i.e.* the number of arguments of ω). An **interpretation** of the signature is a set X together with a mapping $\omega_X : X^{\alpha(\omega)} \longrightarrow X$ for each $\omega \in \Omega$. Given two interpretations X and Y of the same signature, a **morphism of interpretations** from X to Y is a mapping $f : X \longrightarrow Y$ such that for all $\omega \in \Omega$ and for all $(x_1, \ldots, x_{\alpha(\omega)}) \in X^{\alpha(\omega)}$ the equality below holds.

$$f(\omega_X(x_1, \ldots, x_{\alpha(\omega)})) \quad = \quad \omega_Y(f(x_1), \ldots, f(x_{\alpha(\omega)}))$$

An **algebraic theory** $\mathbb{T}$ is a signature together with a collection of axioms of the form below, where Φ and Ψ are terms built on the operators of Ω and whose free variables are in $\{x_1, \ldots, x_n\}$.

$$\forall x_1 \ldots \forall x_n \qquad \Phi(x_1, \ldots, x_n) \quad = \quad \Psi(x_1, \ldots, x_n)$$

A **model** of the theory is an interpretation of its signature satisfying all its axioms. A **morphism of models** is just a morphism of interpretations between models of the theory. The models of $\mathbb{T}$ and their morphisms form the category $\mathbf{Mdl}_{\mathbb{T}}$.

Remark 9.5.2. If $\mathbb{T}_2 \subseteq \mathbb{T}_1$ are two theories sharing the same signature, then $\mathbf{Mdl}_{\mathbb{T}_1}$ is a full subcategory of $\mathbf{Mdl}_{\mathbb{T}_2}$.

Most of the mathematical objects considered in algebra arise as models of some algebraic theory.

Example 9.5.3. The theory of semigroups is algebraic, being built on a single binary operator $\cdot$ and the associativity axiom:

$$(x \cdot y) \cdot z \quad = \quad x \cdot (y \cdot z)$$

The theory of monoids is also algebraic; it suffices to add a constant ε to the signature and the neutral element axiom:

$$\varepsilon \cdot x \quad = \quad x \quad = \quad x \cdot \varepsilon$$

The theory of groups is then obtained by adding a unary operator ι and the inverse axiom:

$$x \cdot \iota(x) \quad = \quad \varepsilon \quad = \quad \iota(x) \cdot x$$

The theory of commutative semigroups (resp. monoids, groups[2]) is obtained by adding the commutativity axiom:

$$x \cdot y \quad = \quad y \cdot x$$

The additive notation "+" is often preferred when the binary operator is commutative. In that case, the neutral element is denoted by 0 instead of ε. The theory of rings is also algebraic, its signature contains two binary operators, a unary operator, and a constant. The binary operators and the constant are usually denoted by $+$, $\cdot$, and 0, and called *sum*, *product*, and *zero*. The unary operator, called *opposite*, is usually represented by a minus sign "−" in prefix position. The axioms of the theory of rings state that $\{+, -, 0\}$ is an abelian group, that $\cdot$ is a semigroup, and that the latter acts on both sides of the former which is formalized by the left and right distributivity axioms:

$$x \cdot (y + z) \quad = \quad x \cdot y + x \cdot z \qquad \text{and} \qquad (y + z) \cdot x \quad = \quad y \cdot x + z \cdot x$$

The theory of commutative rings is obtained by requiring that the multiplicative semigroup be commutative, in which case the left distributivity axiom is equivalent to the right one. From there, the theory of commutative rings with unit is obtained by adding a constant, usually denoted by 1, together with an extra axiom stating that 1 is the neutral element of the product $\cdot$ in which case the multiplicative semigroup is actually a commutative monoid.

Given A, B and X three models of the same theory, a **bimorphism** from $A \times B$ to X is a mapping $f : A \times B \longrightarrow X$ such that for all $a \in A$ and for all $b \in B$ the mappings $f(a, \_) : B \longrightarrow X$ and $f(\_ , b) : A \longrightarrow X$ are morphisms. The composite $f \circ g$ of a bimorphism $f : A \times B \longrightarrow X$ and a morphism $g : X \longrightarrow Y$ is again a bimorphism. As a consequence there is a functor $\mathrm{Bim}(A, B)$ from the category of models of the theory to **Set** sending X to the set of bimorphisms from $A \times B$ to X. An important result about algebraic theories is that the functor $\mathrm{Bim}(A, B)$ is **representable** (*i.e.* there is a (necessarily unique) model $A \otimes B$ such that the functor $\mathrm{Bim}(A, B)$ is isomorphic to $\mathbf{Mdl}_{\mathbb{T}}(A \otimes B, \_)$). It amounts to say that there is a bimorphism $T : A \times B \longrightarrow A \otimes B$ such that for every bimorphism $F : A \times B \longrightarrow X$ there is a unique morphism $h \in \mathbf{Mdl}_{\mathbb{T}}(A \otimes B, X)$ such that $F = h \circ T$. Following the common usage, the elements of the image of T are called the **pure tensors** and we write $a \otimes b$ instead of $T(a, b)$ for all $(a, b) \in A \times B$. By definition T is the **tensor product** of A and B in $\mathbf{Mdl}_{\mathbb{T}}$ though it is also referred to as $A \otimes B$. We also write **universal tensor product** in the event that another notion tensor product be under consideration for the objects of $\mathbf{Mdl}_{\mathbb{T}}$ (*cf.* Example 9.5.8). For a general treatment of tensor product of algebraic theories see (Borceux, 1994b, Th.3.11.3 p.173). Since the notion of bimorphism dramatically depends on the underlying algebraic category (*i.e.* on the theory modelled by its objects) so does the tensor product.

[2] Commutative groups are usually called abelian groups.

Example 9.5.4. Let A and B be two monoids and $f : A \times B \longrightarrow X$ be a bimorphism of monoids. Since morphisms preserve neutral elements we have $f(\varepsilon_A, b) = f(a, \varepsilon_B) = \varepsilon_X$ for all $a \in A$ and all $b \in B$. If $g : A \times B \longrightarrow X$ is just a bimorphism of semigroups one may have $a \in A$ and $b \in B$ such that $g(\varepsilon_A, b) \neq g(a, \varepsilon_B)$. The tensor product of semigroups have been introduced and studied in (Grillet (1969a,b)).

Some inclusion functors preserve tensor products.

Remark 9.5.5. Assume that $\mathbf{Mdl}_{\mathbb{T}_1}$ is a full subcategory of $\mathbf{Mdl}_{\mathbb{T}_2}$ and that A and B are two models of $\mathbb{T}_1$ whose tensor product in $\mathbb{T}_2$ is still a model of $\mathbb{T}_1$. The tensor products of A and B in $\mathbf{Mdl}_{\mathbb{T}_1}$ and $\mathbf{Mdl}_{\mathbb{T}_2}$ are denoted by $A \otimes_1 B$ and $A \otimes_2 B$. For all $a \in A$ and all $b \in B$ the mappings $x \mapsto x \otimes_2 b$ and $y \mapsto a \otimes_2 y$ are actually morphisms of $\mathbf{Mdl}_{\mathbb{T}_1}$. Therefore one has a unique morphism $f : A \otimes_{\mathbf{Mdl}_{\mathbb{T}_1}} B \longrightarrow A \otimes_{\mathbf{Mdl}_{\mathbb{T}_2}} B$ of $\mathbf{Mdl}_{\mathbb{T}_1}$ such that $f(a \otimes_1 b) = a \otimes_2 b$. Since $\mathbf{Mdl}_{\mathbb{T}_1}$ is a subcategory of $\mathbf{Mdl}_{\mathbb{T}_2}$ the mapping $a, b \mapsto a \otimes_1 b$ is a bimorphism of $\mathbf{Mdl}_{\mathbb{T}_2}$. Therefore one has a unique morphism $g : A \otimes_2 B \longrightarrow A \otimes_1 B$ of $\mathbf{Mdl}_{\mathbb{T}_2}$ such that $g(a \otimes_2 b) = a \otimes_1 b$. By hypothesis g is in fact a morphism of $\mathbf{Mdl}_{\mathbb{T}_1}$. Due to the universal properties characterizing tensor products f and g are inverse of each other and we get an isomorphism.

$$A \otimes_1 B \quad \cong \quad A \otimes_2 B$$

The tensor product of R-modules as defined in linear algebra is actually a special case of universal tensor product.

Example 9.5.6. Given a commutative ring with unit R, the theory of (left) R-module is algebraic. Its signature consist of a binary operator $+$ (*sum*), a unary operator $-$ (*opposite*), a constant 0 (*zero*), and an extra unary operator $r \cdot (\_)$ for each element $r \in R$ providing the *scalar multiplication*. The axioms state that $\{+, -, 0\}$ is an abelian group and that R acts on the left of the abelian group. The latter is formalized by a family of axioms indexed by pairs (r, r') of elements of R:

$$(r +_{\mathrm{R}} r') \cdot x \quad = \quad (r \cdot x) + (r \cdot x) \qquad \text{and} \qquad (r \cdot_{\mathrm{R}} r') \cdot x \quad = \quad (r \cdot (r' \cdot x))$$

The tensor product of (left) R-modules usually defined in linear algebra[3] matches the universal tensor product of (left) R-modules. In particular, if V and V' are vector spaces of dimensions n and n' their product and coproduct are vector spaces of dimension $n + n'$ (hence they are isomorphic). In comparison, their tensor product is a vector space of dimension $n \cdot n'$. The latter property is even used in (Brešar, 2014, p.79) to introduce the notion of tensor product of vector spaces.

Tensor products can be drastically degenerated.

Remark 9.5.7. Assuming that the theory $\mathbb{T}$ has two constants denoted by 0 and 1, any bimorphism f satisfies $f(0, 1) = 0$ and $f(0, 1) = 1$ because both mappings $f(0, \_)$ and $f(\_ , 1)$ preserve constants. In particular, if we consider the theory of commutative rings with unit, the codomain of any bimorphism is reduced to the null ring $\{0\}$.

Consequently, there are situations where the universal tensor product is relaxed to obtain a much more relevant construction.

[3] See (Lang, 2002, Chap.16), (Cohn, 2003, Sect.4.8, p117-125), (Grillet, 2007, p.434-441), (Roman, 2008a, Chap.14), or (Douady and Douady, 1999, p.156-167).

Example 9.5.8. Starting from the theory of (left) R-modules, one defines the theory of commutative algebras over R by adding a binary operator $*$ (*internal product*) to the signature together with axioms expressing that $*$ is bilinear. Formally, it amounts to state that $\{+, *, 0\}$ is a commutative ring and to add the following axiom for each pair (r, r') of elements of R:

$$(r \cdot_{\mathrm{R}} r') \cdot (x * x') \quad = \quad (r \cdot x) * (r' \cdot x')$$

The theory of unital commutative algebras over R is obtained by adding a constant 1 together with the axiom that misses to state that $\{+, *, 0, 1\}$ is a commutative ring with unit. It is clear from Remark 9.5.7 that universal tensor products of unital commutative algebras is pointless. Strictly applying the universal algebra approach, a bimorphism of commutative algebras would satisfy the following relation:

$$(x * x') \otimes (y * y') \quad = \quad (x \otimes y) * (x \otimes y') * (x' \otimes y) * (x' \otimes y')$$

The universal tensor product of the commutative R-algebras A and B does not match their **ordinary tensor product**[4]. The latter indeed consists of the tensor product of their underlying modules endowed with the unique bilinear product that extends the following exchange law (with $a, a' \in A$ and $b, b' \in B$).

$$(a \otimes b) * (a' \otimes b') \quad = \quad (a * a') \otimes (b * b')$$

In particular one easily checks that if both A and B have a unit, then the pure tensor $1_A \otimes 1_B$ is the unit of the ordinary tensor product $A \otimes B$. The motivation here is that the ordinary tensor product of two commutative algebras is actually their coproduct, see (Lang, 2002, Prop.6.1, p.630), (Grillet, 2007, Prop.5.6, p.529), or (Douady and Douady, 1999, p.168).

Semilattices and Friends

Because the notion of semilattice is at the core of this section and because the way it is understood may differ from a source to another, we dedicate some lines to make things clear about it. According to most references, a **join-semilattice** (resp. **meet-semilattice**) is a poset in which all *nonempty* finite subsets admit a least upper bound (resp. a greatest lower bound)[5]. The corresponding algebraic notion is that of semilattice that is to say an idempotent commutative semigroup[6]. The link between the order theoretic and the algebraic approaches is fully detailed in (Roman, 2008b, p.51-52). However, some authors admit the empty subset in the definition of a join-semilattice (resp. meet-semilattice) and consequently, the corresponding algebraic notion becomes that of idempotent commutative monoid[7] which we call *semilattice with zero*. In this section, the semilattices under consideration mainly arise as the join-semilattices of Boolean algebras. In accordance with that remark, the binary operator of a semilattice is denoted by $\vee$ and its neutral element, when it exists, by 0 (compare to Example 9.5.3).

[4] The term *ordinary tensor product* is used in (Lang, 2002, p.629-631). That tensor product is also discussed in (Grillet, 2007, p.527-530) or (Douady and Douady, 1999, p.167-169).

[5] See (Gierz et al., 2003, p.5), (Blyth, 2005, p.19), (Grillet, 2007, p.539), (Grätzer, 2008, p.18), (Roman, 2008b, p.49,51), and (Goubault-Larrecq, 2013, p.10).

[6] See (Birkhoff, 1967, p.9), (Amadio and Curien, 1998, p.225), and (Mac Lane and Birkhoff, 1999, p.475).

[7] See (Hofmann et al., 1974, p.5), (Johnstone, 1982, p.1-2), and (Pedicchio et al., 2003, p.20).

The formal definitions are summarized below.

A **semigroup** is a set X together with an associative law $\vee$. An element $0 \in X$ is said to be **neutral** or **zero** when the mappings $0 \vee \_$ and $\_ \vee 0$ are both id_X. There is at most one such element in a semigroup. A **monoid** is a semigroup with a neutral element. The semigroup (resp. monoid) is said to be **idempotent** when $x \vee x = x$ for all $x \in X$. A **semilattice** is a commutative idempotent semigroup. Therefore a **semilattice with zero** is a commutative idempotent monoid. Any semilattice induces a partial order on X putting $x \sqsubseteq_\vee y$ when $x \vee y = y$. Conversely a partial order on X whose pairs have a least upper bound induces a semilattice. A **lattice** is a pair $\vee, \wedge$ of semilattices structures such that $\sqsubseteq_\wedge = \sqsubseteq_\vee^{\mathrm{op}}$, which is the case if and only if the absorptive law is satisfied for all $x, y \in X$.

$$x \vee (x \wedge y) \quad = \quad x \quad = \quad x \wedge (x \vee y)$$

A lattice is said to be **bounded** when it comes with a neutral element 0 for $\vee$ and a neutral element 1 for $\wedge$, which mean that the following holds for all $x \in X$.

$$0 \vee x \quad = \quad x \quad = \quad x \wedge 1$$

The lattice is said to be degenerated when $0 = 1$. Moreover a lattice is said to be **distributive** when the equality therein under is satisfied for all $x, y, z \in X$.

$$x \wedge (y \vee z) \quad = \quad (x \wedge y) \vee (x \wedge z)$$

In this case it is well-known that the dual equality also holds (Birkhoff, 1967, p.11).

$$x \vee (y \wedge z) \quad = \quad (x \vee y) \wedge (x \vee z)$$

Then a **Boolean algebra** is defined as a bounded distributive lattice together with a **complement** that is a bijection $x \in X \mapsto x^c \in X$ such that $x \vee x^c = 0$ and $x \wedge x^c = 1$ for all $x \in X$, where 0 and 1 are the neutral elements for $\vee$ and $\wedge$. The fact that this permutation is actually an isomorphism of commutative idempotent monoid between $(X, \vee, 0)$ and $(X, \wedge, 1)$ is known as **De Morgan laws**. The definitions given above are summarised on Figure 9.2. In particular the tensor product of a pair of Boolean algebras can be taken in any category appearing on the diagram of Figure 9.3, the arrows on it being inclusion functors.

Tensor Product of Boolean Algebras as Idempotent Rings

The coproduct of two Boolean algebras can also be defined as the ordinary tensor product (*cf.* Example 9.5.8) of their corresponding Boolean rings seen as commutative algebras over the 2 elements field $\mathbb{F}_2$. The purpose of this section is to formalize the above statement which seems to be part of the Boolean algebra folklore. What might not be so well-known is that in this case, it also matches the universal tensor product of the corresponding idempotent rings. A ring R is said to be **idempotent** when $x^2 = x$ holds for all its elements. The theory of idempotent Boolean rings is thus algebraic. Any idempotent ring has characteristic 2 (*i.e.* $x + x = 0$ holds for all its elements), and any ring of characteristic 2 is commutative (Givant and Halmos, 2009, p.3). An idempotent ring A is said to be **Boolean** when it is unital (*i.e.* the product has a neutral element, denoted by 1). In particular, a Boolean ring can be seen as a commutative $\mathbb{F}_2$-algebra. The only possible scalar product is given by $0 \cdot a = 0_A$ and $1 \cdot a = a$. As

Structure	Signature	Axioms	Category
semilattice	$\vee$	commutative idempotent semigroup	**SLat**
semilattice with zero	$\vee, 0$	commutative idempotent monoid	$\mathbf{SLat_0}$
lattice	$\vee, \wedge$	two semilattices with $\sqsubseteq_\wedge = \sqsubseteq_\vee^{op}$	**Lat**
distributive lattice	$\vee, \wedge$	lattice in which $\wedge$ distributes over $\vee$	**DLat**
distributive lattice with zero	$\vee, 0, \wedge$	distributive lattice in which $\vee$ has a neutral element	$\mathbf{DLat_0}$
distributive lattice with difference	$\vee, 0, \wedge, \backslash$	distributive lattice with zero s.t. $(x \backslash y) \vee (x \wedge y) = x$ $(x \backslash y) \wedge y = 0$	$\mathbf{DLat_d}$
bounded distributive lattice	$\vee, 0, \wedge, 1$	lattice in which both $\vee$ and $\wedge$ have a neutral element	$\mathbf{DLat_b}$
Boolean algebra	$\vee, 0, \wedge, 1, \_^c$	bounded distributive lattice s.t. $x^c \wedge x = 0$ and $x^c \vee x = 1$	**BoolAlg**
	$\vee, 0, \wedge, 1, \backslash$	bounded distributive lattice with difference	

Figure 9.2: Semilattices and friends

$$\begin{tikzcd}
 & \mathbf{DLat_b} \arrow[dr] & & \mathbf{DLat} \arrow[dr] & \\
\mathbf{BoolAlg} \arrow[ur] \arrow[dr] & & \mathbf{DLat_0} \arrow[ur] \arrow[dr] & & \mathbf{SLat} \\
 & \mathbf{DLat_d} \arrow[ur] & & \mathbf{SLat_0} \arrow[ur] &
\end{tikzcd}$$

Figure 9.3: Between Boolean algebras and semilattices

a consequence, the mapping that sends an idempotent ring to its associated $\mathbb{F}_2$-algebra extends to a fully faithful functor from **IdemRng** (*i.e.* the category of idempotent rings) to $\mathbb{F}_2$-**CAlg** (*i.e.* the category of $\mathbb{F}_2$-algebras). Hence we can form $A \otimes_{\mathbb{F}_2} B$ the ordinary tensor product of the Boolean rings A and B seen as $\mathbb{F}_2$-algebras (Lang, 2002, p.629-631). From Example 9.5.8 we know that any element of the ordinary tensor product is a linear combination of elements of the form $(a \otimes b)$, the latter being called the **pure tensors**, while the coefficients are taken in $\mathbb{F}_2$. Moreover, by definition of the ordinary tensor product and by idempotency we have the following equality.

$$(a \otimes b) \cdot (a \otimes b) \quad = \quad (a \cdot a) \otimes (b \cdot b) \quad = \quad (a \otimes b)$$

The square function $x \mapsto x^2$ being linear in characteristic 2 we deduce that the $\mathbb{F}_2$-algebra $\mathcal{R}(A) \otimes_{\mathbb{F}_2} \mathcal{R}(B)$ is actually an idempotent ring.

$$\left(\sum_{i=1}^{n} a_i \otimes b_i\right)^2 \quad = \quad \sum_{i=1}^{n} (a_i \otimes b_i)^2 \quad = \quad \sum_{i=1}^{n} a_i \otimes b_i$$

Moreover $(1_A \otimes 1_B)$ is clearly the unit element of $\mathcal{R}(A) \otimes_{\mathbb{F}_2} \mathcal{R}(B)$. We have proven the following fact.

Lemma 9.5.9. The ordinary tensor product of a pair of idempotent (resp. Boolean) rings (seen as commutative $\mathbb{F}_2$-algebras) is an idempotent (resp. Boolean) ring.

As we have seen in Example 9.5.8 the ordinary tensor product of the commutative $\mathbb{F}_2$-algebras A and B is their coproduct in the category $\mathbb{F}_2$-**CAlg**. By Lemma 9.5.9 the colimit diagram is in the image of the inclusion **IdemRng** $\hookrightarrow$ $\mathbb{F}_2$-**CAlg** which is full and faithful, and therefore reflects colimits (Borceux, 1994a, p. 65).

Remark 9.5.10. The inclusion functor **BoolRng** $\hookrightarrow$ **IdemRng** is not full because the morphisms of its codomain are not require to preserve units, however it reflects coproducts. Let A_1 and A_2 be Boolean rings. Their coproduct in **IdemRng** exists, it is given by the morphisms $i_k : A_k \longrightarrow A_1 \otimes_{\mathbb{F}_2} A_2$ which are defined by $i_1(x) = x \otimes 1$ for all $x \in A_1$ and $i_2(x) = 1 \otimes x$ for all $x \in A_2$. Since $1 \otimes 1$ is the unit of $A_1 \otimes_{\mathbb{F}_2} A_2$ both i_1 and i_2 are indeed morphisms of Boolean rings. Let $f_k : A_k \longrightarrow A_3$ be morphisms of Boolean rings for $k \in \{1, 2\}$. There is a unique idempotent ring morphism $h : A_1 \otimes_{\mathbb{F}_2} A_2 \longrightarrow A_3$ such that $f_k = h \circ i_k$ for $k \in \{1, 2\}$. In particular we have $h(1_{A_1} \otimes 1_{A_2}) = h \circ i_1(1_{A_1}) = f_1(1_{A_1}) = 1_{A_3}$ so h is actually a Boolean ring morphism.

We have proven the following result.

Lemma 9.5.11. The coproduct of two Boolean rings is given by the ordinary tensor product of their corresponding $\mathbb{F}_2$-algebras.

Remark 9.5.12. As they are commutative rings in the usual sense (Lang, 2002, p.83-84), Boolean rings can also be seen as commutative $\mathbb{Z}$-algebra so we can form the ordinary tensor product $R \otimes_{\mathbb{Z}} S$. However, in that case, the square function is no longer linear, and consequently the commutative ring $R \otimes_{\mathbb{Z}} S$ is not idempotent.

By Remark 9.5.7 the universal tensor product of Boolean rings or algebras is pointless. However one may wonder how interesting is the universal tensor product of idempotent rings. The next result answers that question.

Proposition 9.5.13. Let R_1 and R_2 be Boolean rings. The universal tensor product of the associated idempotent rings (*i.e.* $R_1 \otimes_{\mathrm{Id}} R_2$) and the ordinary tensor product of the associated $\mathbb{F}_2$-algebras (*i.e.* $R_1 \otimes_{\mathbb{F}_2} R_2$) are isomorphic as Boolean rings.

Proof. The mapping which sends an ordered pair $(x, y) \in R_1 \times R_2$ to the pure tensor $x \otimes y \in R_1 \otimes_{\mathrm{Id}} R_2$ induces a bimorphism of $\mathbb{F}_2$-vector spaces. Hence we have a unique $\mathbb{F}_2$-linear map f that sends a pure tensor of $R_1 \otimes_{\mathbb{F}_2} R_2$ to the corresponding one in $R_1 \otimes_{\mathrm{Id}} R_2$. Conversely, the pure tensors of the ordinary tensor product satisfy the following equalities for all $x, x' \in R_1$ and all $y, y' \in R_2$

$$\begin{array}{rclcl} xx' \otimes y & = & xx' \otimes y^2 & = & (x \otimes y) \cdot (x' \otimes y) \\ x \otimes yy' & = & x^2 \otimes yy' & = & (x \otimes y) \cdot (x \otimes y') \end{array}$$

Therefore the mapping which sends an ordered pair $(x, y) \in R_1 \times R_2$ to the pure tensor $x \otimes y \in R_1 \otimes_{\mathbb{F}_2} R_2$ induces a bimorphism of idempotent rings. Hence we have a unique idempotent ring morphism g that sends a pure tensor of $R_1 \otimes_{\mathrm{Id}} R_2$ to the corresponding one in $R_1 \otimes_{\mathbb{F}_2} R_2$. As a consequence of the universal properties satisfied by $R_1 \otimes_{\mathrm{Id}} R_2$ and $R_1 \otimes_{\mathbb{F}_2} R_2$ the $\mathbb{F}_2$-linear map $g \circ f$ and the morphism of idempotent rings $f \circ g$ are identities. In particular g is a bijective morphism of idempotent rings so it is actually an isomorphism. □

A Boolean algebra $(A, \vee, \wedge, 0, 1, (\_)^{\mathrm{c}})$ can be turned into a Boolean ring $\mathcal{R}(A) = (A, +, \cdot, 0, 1)$ by setting $a + b = (a \vee b) \wedge (a \wedge b)^{\mathrm{c}} = (a \wedge b^{\mathrm{c}}) \vee (b \wedge a^{\mathrm{c}})$ and $a \cdot b = a \wedge b$. Conversely, any Boolean ring $(R, +, \cdot, 0, 1)$ can be turned into a Boolean algebra $\mathcal{A}(R) = (R, \vee, \wedge, 0, 1, (\_)^{\mathrm{c}})$ by setting $x^{\mathrm{c}} = x + 1$, $x \wedge y = x \cdot y$, and $x \vee y = x + y + x \cdot y$. The constructions $\mathcal{A}(\_)$ and $\mathcal{R}(\_)$ extend to an isomorphism between the category **BoolAlg** of Boolean algebras and the category **BoolRng** of Boolean rings. That correspondence is concisely explained in (Johnstone, 1982, p.4-7), and more thoroughly in the first three chapters of Givant and Halmos (2009).

$$\begin{tikzcd} \mathbf{BoolAlg} \arrow[r, shift left, "\mathcal{R}ng"] & \mathbf{BoolRng} \arrow[l, shift left, "\mathcal{A}lg"] \end{tikzcd}$$

Lemma 9.5.11, Proposition 9.5.13 and the isomorphism between **BoolAlg** and **BoolRng** provide a natural notion of tensor product of Boolean algebras which also corresponds to the binary coproducts of Boolean algebras.

$$A_1 \otimes A_2 \quad = \quad \mathcal{A}lg(\mathcal{R}ng(A_1) \otimes_{\mathbb{F}_2} \mathcal{R}ng(A_2)) \quad = \quad \mathcal{A}lg(\mathcal{R}ng(A_1) \otimes_{\mathrm{Id}} \mathcal{R}ng(A_2))$$

Tensor Product of Boolean Algebras as Semilattices with Zero

This section is dedicated to finding an inclusion **BoolAlg** $\hookrightarrow$ $\mathbf{Mdl}_{\mathbb{T}}$ such that the tensor product $\mathcal{R}_{A_1} \otimes \cdots \otimes \mathcal{R}_{A_n}$ in $\mathbf{Mdl}_{\mathbb{T}}$ be the Boolean algebra $\mathcal{R}_{A_1 \times \cdots \times A_n}$. Hence the term *tensor product* should always be understood as *universal tensor product* in the sense of Subsection 9.5. In practice we ask which axioms of the theory of Boolean algebras should be dropped to obtain an algebraic theory $\mathbb{T}$ fulfilling the previous requirements. Indeed, noting that the following relations[8]

$$\begin{array}{rcl} (A \times B) \cup (A \times C) & = & A \times (B \cup C) \\ (A \times B) \cap (A \times C) & = & A \times (B \cap C) \\ A \times \emptyset & = & \emptyset \end{array}$$

[8] Besides their theoretical interests, these formulas are actually the ones used to perform calculations in the OCaml library handling regions.

hold for all elements A, B, and C of a given powerset, we are tempted to think that the following relation

$$\mathcal{R}_{A_1\times\cdots\times A_n} \quad\cong\quad \mathcal{R}_{A_1}\otimes\cdots\otimes\mathcal{R}_{A_n}$$

could hold for a suitably chosen universal tensor product. That approach is motivated by the intuition that blocks (*cf.* Definition 6.2.1) should play the role of pure tensors. However by noting that given an element X of the powerset of A, the set product $X\times A$ is not the greatest element of the powerset of the set product $A\times A$, one gets convinced that the tensor product of Boolean algebras does not fit because, for example, the mapping $X \mapsto X\times A$ do not preserve the unit. The first axiom one may wish to drop from the theory of Boolean algebras is thus the presence of a unit. By the way, we also address the issue that the notion of bimorphisms of Boolean algebras is so rigid that when they exist, their codomain is degenerated (*cf.* Remark 9.5.7).

A block is a Cartesian product of *nonempty* sets (*cf.* Definition 6.2.1). Pushing the intuition that they are pure tensors the mappings $X \mapsto X\times A$ preserve the empty set. The preceding statement translated to the language of tensor products amounts to say that bimorphisms preserve the constant zero. It is thus reasonable to think that the algebraic theory we are looking for at least contains that of semilattices with zero.

However, dropping the unit from the theory of Boolean algebras described in Subsection 9.5 also implies losing the *complement* operator because the unit occurs in the axioms related to it. What remains is the theory of distributive lattices with zero. Nevertheless, by describing the notion of Boolean algebra by another set of axioms that loss can be avoided. Replace the unary *complement* operator $(\_)^{\mathrm{c}}$ by the binary *difference* operator $\backslash$ in the signature. Also replace, in the theory, all the axioms involving the complement operator by the following ones:

$$(x\backslash y)\vee(x\wedge y) \quad=\quad x$$

$$(x\backslash y)\wedge y \quad=\quad 0$$

The relevant point is that the unit does not occur in those axioms. One easily checks that the theory of Boolean algebras defined by means of the unary *complement* and the one defined by means of the binary *difference* are equivalent. One switches from one to the other by setting $x\backslash y := x\wedge y^{\mathrm{c}}$ and $x^{\mathrm{c}} := 1\backslash x$. The signature of the theory of **distributive lattice with difference** is $\{\vee, 0, \wedge, \backslash\}$ and its axioms are those of the theory of distributive lattices with zero plus the *difference* axioms given above, see also Figure 9.2. De Morgan's law have their counterparts in distributive lattices with difference.

Lemma 9.5.14. Let $x, y_1, \ldots, y_n$ be elements of a distributive lattice. If all the differences $x\backslash y_i$ exist then the difference $x\backslash(y_1\vee\cdots\vee y_n)$ also exist and it is equal to the meet of differences $(x\backslash y_1)\wedge\cdots\wedge(x\backslash y_n)$.

Proof. Because the lattice is distributive we have the following equality.

$$\left(\bigwedge_{i=1}^{n}(x\backslash y_i)\right)\vee\left(x\wedge\bigvee_{i=1}^{n}y_i\right) \quad=\quad \underbrace{\left(x\vee\bigwedge_{i=1}^{n}(x\backslash y_i)\right)}_{a} \quad\wedge\quad \underbrace{\left(\bigwedge_{i=1}^{n}(x\backslash y_i)\vee\bigvee_{i=1}^{n}y_i\right)}_{b}$$

Note that for each $i\in\{1,\ldots,n\}$ we have $x\backslash y_i \sqsubseteq x$ hence $a = x$.

$$a \quad=\quad \bigwedge_{i=1}^{n}\underbrace{(x\vee x\backslash y_i)}_{x} \quad=\quad x$$

Also note that b can be written as follows.

$$b \quad = \quad \bigwedge_{i=1}^{n}(x\backslash y_i) \vee \bigvee_{i=1}^{n} y_i \quad = \quad \bigwedge_{i=1}^{n}\left(x\backslash y_i \vee \bigvee_{j=1}^{n} y_j\right)$$

Since $x \sqsubseteq x\backslash y_i \vee y_i$ for all $i \in \{1, \ldots, n\}$ we have $x \sqsubseteq b$. Therefore $a \wedge b = x$. Then we consider the following equality

$$\left(\bigwedge_{i=1}^{n}(x\backslash y_i)\right) \wedge \left(\bigvee_{j=1}^{n} y_j\right) \quad = \quad \bigvee_{j=1}^{n} \underbrace{\left(y_j \wedge \bigwedge_{i=1}^{n}(x\backslash y_i)\right)}_{0}$$

and observe that the right hand term is 0 because $x\backslash y_j \wedge y_j = 0$. □

Lemma 9.5.15. Let $x_1, \ldots, x_n, y$ be elements of a distributive lattice. If all the differences $x_i\backslash y$ exist then the difference $(x_1 \vee \cdots \vee x_n)\backslash y$ also exist and it is equal to the join of differences $(x_1\backslash y) \vee \cdots \vee (x_n\backslash y)$.

Proof. It derives from the following routine calculations.

$$\bigvee_{i=1}^{n}(x_i\backslash y) \vee \left(y \wedge \bigvee_{i=1}^{n} x_i\right) \quad = \quad \bigvee_{i=1}^{n} \underbrace{(x_i\backslash y \vee (y \wedge x_i))}_{x_i} \quad = \quad \bigvee_{i=1}^{n} x_i$$

$$\left(\bigvee_{i=1}^{n} x_i\backslash y\right) \wedge y \quad = \quad \bigvee_{i=1}^{n} \underbrace{(x_i\backslash y \wedge y)}_{0} \quad = \quad 0$$

□

Remark 9.5.16. In (Birkhoff, 1967, p.16) a lattice is said to be **relatively complemented** when for all $a \sqsubseteq b \sqsubseteq c$ there exists some d such that $c \wedge d = a$ and $c \vee d = b$. If the lattice is distributive, then such an element d is unique. The uniqueness is due to a more general fact that holds in any distributive lattice: if $c \wedge x = c \wedge y$ and $c \vee x = c \vee y$, then $x = y$ (Birkhoff, 1967, Thm.10, p.12). It follows that the distributive lattices with difference are exactly the relatively complemented distributive lattices with zero. Given a distributive lattice with difference, the element d is given by $(b\backslash c) \vee a$. Conversely, if one has a relatively complemented distributive lattice with zero, then one has $0 \sqsubseteq x \wedge y \sqsubseteq x$ and $x\backslash y$ is the unique z such that $(x \wedge y) \vee z = x$ and $x \wedge y \wedge z = 0$. Another consequence is that the category $\mathbf{DLat_d}$ of distributive lattices with difference is a full subcategory of the category $\mathbf{DLat_0}$ of distributive lattices with zero (see Figure 9.2).

Remark 9.5.17. In some sense the theory of distributive lattices with difference is that of *Boolean algebras without unit*. That statement is formalized by an extension of the isomorphism between the categories **BoolAlg** and **BoolRng**. A distributive lattice with difference $(A, \vee, \wedge, 0, \backslash)$ can be turned into a idempotent ring $\mathcal{R}_{ng}(A) = (A, +, \cdot, 0)$ by setting $a + b = (a\backslash b) \vee (b\backslash a)$ and $a \cdot b = a \wedge b$. Conversely, any idempotent ring $(R, +, \cdot, 0)$ can be turned into a distributive lattice with difference $\mathcal{A}_{lg}(R) = (R, \vee, \wedge, 0, \backslash)$ by setting $x\backslash y = x + x \cdot y$, $x \wedge y = x \cdot y$, and $x \vee y = x + y + x \cdot y$. The constructions

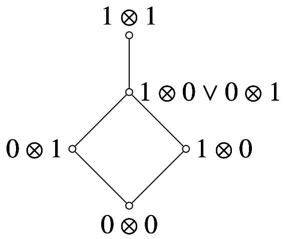


Figure 9.4: The Hasse diagram of $\{0 < 1\} \otimes \{0 < 1\}$ in **SLat**

$\mathcal{A}lg(\_)$ and $\mathcal{R}ng(\_)$ extend to an isomorphism between the categories of distributive lattices with difference and that of idempotent rings.

$$\begin{tikzcd}
\mathbf{DLat_d} \arrow[r, shift left, "\mathcal{R}ng"] & \mathbf{IdemRng} \arrow[l, shift left, "\mathcal{A}lg"] \\
\mathbf{BoolAlg} \arrow[u, "\subseteq"] \arrow[r, shift left, "\mathcal{R}ng"] & \mathbf{BoolRng} \arrow[l, shift left, "\mathcal{A}lg"] \arrow[u, "\subseteq"']
\end{tikzcd}$$

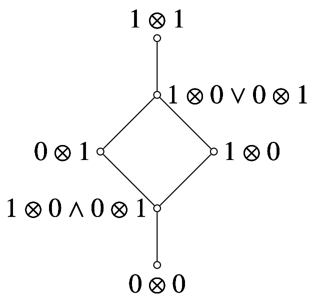
The tensor products in $\mathbf{DLat_d}$ and the one in **IdemRng** are therefore related.

We have therefore three algebraic theories that might meet the requirements of the question asked at the beginning of this section: the theory of semilattices with zero, that of distributive lattices with zero, and that of distributive lattices with difference. As we shall see, they all fit. What really matters is actually the presence of a zero so the meet operator can indeed be removed. Therefore we first study the case of semilattices with zero. Tensor products of semilattices and other related structures have already been the source of many publications (*e.g.* tensor product in **DLat** and tensor product of distributive lattices in **SLat** are treated in Fraser (1976a,b)). The tensor product in **SLat** appears in Fraser (1978), and in Anderson and Kimura (1978), while the tensor product in $\mathbf{SLat_0}$ is the subject of Grätzer et al. (1981), and Grätzer and Wehrung (2000, 2001). The basic properties of the category $\mathbf{SLat_0}$, in particular the construction of tensor product in it, are very well exposed in the first chapter of (Hofmann et al. (1974)). Note that semilattices and semilattices with zero are called "protosemilattice" and "semilattice" in (Hofmann et al., 1974, p.5). The next example highlights the distinction between tensor products in **SLat** and $\mathbf{SLat_0}$.

Example 9.5.18 (Haucourt and Ninin (2014))**.** Let B be the Boolean algebra $\{0, 1\}$ with $x \wedge y = \min(x, y)$, $x \vee y = \max(x, y)$, and $x^c = x + 1 \mod 2$. We determine $B \otimes B$ in $\mathbf{SLat_0}$, $\mathbf{DLat_0}$, **SLat**, and **DLat**. First note that a bimorphism $F : \{0, 1\}^2 \to S$ of $\mathbf{SLat_0}$ sends all the elements of $\{0, 1\}^2$ but $(1, 1)$ to the neutral element of S, namely 0_S. Then note that any order-preserving map defined over B also preserve binary joins and meets. It follows that the relation $B \otimes B \cong B$ holds in both $\mathbf{SLat_0}$ and $\mathbf{DLat_0}$. By similar arguments we prove that given $X \in \mathbf{SLat_0}$ (resp. $X \in \mathbf{DLat_0}$) $B \otimes X \cong X$ holds in $\mathbf{SLat_0}$ (resp. $\mathbf{DLat_0}$).

We now prove that the tensor product $B \otimes B$ in **SLat** is the bounded distributive lattice C whose corresponding Hasse diagram is depicted on Figure 9.4. Remark that a set map $F : \{0, 1\}^2 \to S$ is a bimorphism of **SLat** iff we have the relations

$$F(0, 0) \sqsubseteq F(0, 1) \sqsubseteq F(1, 1) \qquad \text{and} \qquad F(0, 0) \sqsubseteq F(1, 0) \sqsubseteq F(1, 1)\ .$$



$1 \otimes 1$
$1 \otimes 0 \vee 0 \otimes 1$
$0 \otimes 1$
$1 \otimes 0$
$1 \otimes 0 \wedge 0 \otimes 1$
$0 \otimes 0$

Figure 9.5: The Hasse diagram of $\{0 < 1\} \otimes \{0 < 1\}$ in **DLat**

Let us check that there is a unique $h \in \mathbf{SLat}(C, S)$ satisfying $F = h \circ T$. Firstly we have $h(a \otimes b) = F(a, b)$ for all $(a, b) \in \{0, 1\}^2$ in order to have the equality $h(a \otimes b) = F(a, b)$ satisfied. Because h is a morphism of **SLat** it preserves binary join therefore it comes

$$h(0 \otimes 1 \vee 1 \otimes 0) \quad = \quad h(0 \otimes 1) \vee h(1 \otimes 0) \quad = \quad F(0, 1) \vee F(1, 0)$$

so h is uniquely defined. Checking that h is indeed a morphism of **SLat** is a routine verification based on the previous relation and the fact that F is a bimorphism. Note that h might not preserve existing meets.

A similar reasoning proves that the tensor product $B \otimes B$ in **DLat** is the bounded distributive lattice whose corresponding Hasse diagram is depicted on Figure 9.5.

The next lemma solves the word problem for tensor products in $\mathbf{SLat_0}$.

Lemma 9.5.19 (Grätzer et al. (1981), Thm.2.1, p.505)**.** Let A, B be lattices with zero. Let $a, a_1, \ldots, a_n \in A \setminus \{0_A\}$ and let $b, b_1, \ldots, b_n \in B \setminus \{0_B\}$. Then

$$a \otimes b \quad \sqsubseteq \quad \bigvee_{i=1}^{n} a_i \otimes b_i$$

iff there are finite subsets $S_1, \ldots, S_m$ of $\{1, \ldots, n\}$ such that

$$a \sqsubseteq \bigvee_{j=1}^{m} \bigwedge_{i \in S_j} a_i \quad \text{and} \quad b \sqsubseteq \bigwedge_{j=1}^{m} \bigvee_{i \in S_j} b_i$$

Remark 9.5.20. Grätzer et al. (1981) state Lemma 9.5.19 without proof, claiming that it is an immediate variant of a result by G. A. Fraser (1978). The latter deals with tensor product of semilattices in **SLat** (instead of $\mathbf{SLat_0}$) assuming that A and B are semilattices (instead of lattices). In doing so, one has to replace the meets of elements by the corresponding meets of principal ideals, which are ideals but may fail to be principal. The prototype of the preceding two results is (Fraser, 1976a, Th.2.5, p.185).

Remark 9.5.21. As an immediate consequence of Lemma 9.5.19 – see also (Grätzer et al., 1981, Cor.2.2, p.505), the pure tensors are compared component by component. In other words the equivalence $a \otimes b \sqsubseteq c \otimes d$ iff $a = 0_A$, or $b = 0_B$, or $a \sqsubseteq c$ and $b \sqsubseteq d$, holds for all $a, c \in A$ and $b, d \in B$. It is worth noticing the following subtlety. Since we are taking the tensor product in $\mathbf{SLat_0}$ the morphisms $a \mapsto a \otimes 0_B$ and $b \mapsto 0_A \otimes b$ preserve joins. As a consequence $0_A \otimes 0_B$ is the least pure tensor, and since every element of $A \otimes B$ is the join of finitely many pure tensors and $\vee$ is associative, the element $0_A \otimes 0_B$ is actually the neutral element of $\vee$, in other words the zero of the semilattice $A \otimes B$. The above reasoning is actually valid for both tensor products in **SLat** and $\mathbf{SLat_0}$. However, in the latter case, the morphisms $a \mapsto a \otimes 0_B$ and $b \mapsto 0_A \otimes b$ also preserve zero, therefore all the pure tensors $a \otimes 0_B$ and $0_A \otimes b$ are identified with $0_A \otimes 0_B$. The latter argument is not valid for the tensor product in **SLat**. See also Section 6.3.

Assuming that both A and B have a greatest element, respectively 1_A and 1_B, the pure tensor $1_A \otimes 1_B$ is greater than any other. As before we conclude that it is absorbing for $\vee$. Moreover, if $A \otimes B$ is actually a lattice, then $1_A \otimes 1_B$ is its unit.

Remark 9.5.22. As a consequence of Lemma 9.5.19 any element x of $A \otimes_{\mathbf{SLat_0}} B$ can be written as a finite join of pure tensors $c_j \otimes d_j$, with $j \in \{1, \ldots, m\}$, such that for all pure tensors $a \otimes b$, one has $a \otimes b \sqsubseteq x$ iff there exists $j \in \{1, \ldots, m\}$ such that $a \otimes b \sqsubseteq c_j \otimes d_j$. Dropping some elements, one can even suppose the pure tensors $c_j \otimes d_j$ are maximal in the sense that $c_j \otimes d_j \sqsubseteq c_{j'} \otimes d_{j'}$ implies that $j = j'$. One readily checks that the resulting family is unique. By analogy with the maximal block covering of an isothetic region (*cf.* Definition 6.2.1) the family $(c_j \otimes d_j)_{j=1\ldots m}$ is called the **maximal pure tensor covering** of x. Corollary 9.5.27 actually derives from this observation.

The two next remarks are standard facts about isomorphisms of semilattices, they will be applied in the proofs of Proposition 9.5.25 and Corollary 9.5.27.

Remark 9.5.23. A semilattice isomorphism $f : A \to B$ between lattices A and B is actually a lattice isomorphism. Given $a, a' \in A$ we have $a \wedge a' \sqsubseteq a, a'$ from which we get $f(a \wedge a') \sqsubseteq f(a), f(a')$, and $f(a \wedge a') \sqsubseteq f(a) \wedge f(a')$ because the latter is the greatest lower bound of $f(a)$ and $f(a')$. Repeating the same argument with $f(a)$, $f(a')$, and the semilattice isomorphism f^{-1} we get the inequality

$$f^{-1}(f(a) \wedge f(a')) \quad \sqsubseteq \quad f^{-1}(f(a))\,,\ f^{-1}(f(a'))$$

which can be rephrased as below

$$f^{-1}(f(a) \wedge f(a')) \quad \sqsubseteq \quad a \wedge a'$$

the latter inequality being equivalent to $f(a) \wedge f(a') \sqsubseteq f(a \wedge a')$. Thus $f(a \wedge a')$ is equal to $f(a) \wedge f(a')$. Moreover, one easily checks that if A has a least (resp. a greatest) element, then so does B and it is preserved by f. One readily deduces that if A is a Boolean algebra (resp. distributive lattice with difference) then so is B, and f is actually an isomorphism of Boolean algebras (resp. distributive lattices with difference).

Remark 9.5.24. Given two semilattices A and B, a mapping $f : A \to B$ induces a semilattice isomorphism iff it induces a poset isomorphism.

In the many papers published around the notion of tensor product of semilattices and related structures (see the discussion before Example 9.5.18) it seems that the case where the factors are Boolean algebras has never been considered, not even mentioned. For example (Bell et al., 1984, Thm.3.1, p.244) provides a simple formula to compute an invariant of $A \otimes_{\mathbf{DLat}} B$ under the hypothesis that both A and B are Boolean algebras. The tensor product being taken in **DLat** it may not be a Boolean algebra.

Proposition 9.5.25 (Haucourt and Ninin (2014))**.**
Given two distributive lattices with difference (resp. Boolean algebras) A and B, the tensor products $A \otimes_{\mathbf{SLat_0}} B$, $A \otimes_{\mathbf{DLat_0}} B$ and $A \otimes_{\mathbf{DLat_d}} B$ are isomorphic distributive lattices with difference (resp. Boolean algebras).

Before starting the proof, let us make the statement of Proposition 9.5.25 a bit more explicit. The two tensor products exist for abstract reasons explained in Subsection 9.5. However they are defined only up to isomorphism of $\mathbf{SLat_0}$ (resp. $\mathbf{DLat_0}$). What Proposition 9.5.25 actually claims is that any representative of such a tensor product is actually a distributive lattice with difference (resp. a Boolean algebra) provided that the factors are so. By Remark 9.5.23 two representatives of $A \otimes_{\mathbf{SLat_0}} B$ (resp. $A \otimes_{\mathbf{DLat_0}} B$) are thus isomorphic distributive lattices with difference (resp. Boolean algebras). However it might be that the isomorphism class of the distributive lattices with difference (resp. Boolean algebras) $A \otimes_{\mathbf{SLat_0}} B$ differs from that of $A \otimes_{\mathbf{DLat_0}} B$. The fact that they do not is the last part of the statement.

Proof. By (Grätzer et al., 1981, Th.4.7, p.514) we know that the tensor product of two distributive lattice with difference in $\mathbf{SLat_0}$ is a distributive lattice with zero. By Remark 9.5.21 we know that its zero is the pure tensor $0_A \otimes 0_B$. Moreover if both A and B have a unit (*i.e.* is both A and B are actually Boolean algebras) then its unit is $1_A \otimes 1_B$. We prove that an exchange law between $\otimes$ and $\wedge$ holds. First note that $(a \wedge c) \otimes (b \wedge d)$ is less than both $a \otimes b$ and $c \otimes d$. Then suppose that we have

$$\left(\bigvee_{i=1}^{n} x_i \otimes y_i \right) \vee (a \otimes b) \quad = \quad a \otimes b$$

for some $n \in \mathbb{N}$, $x_i \in A$, and $y_i \in B$. By associativity of $\vee$ and an immediate induction we prove that $x_i \otimes y_i \sqsubseteq a \otimes b$ holds for all $i \in \{1, \ldots, n\}$. According to the characterization of the partial order $\sqsubseteq$ between pure tensors (*cf.* Remark 9.5.21) we have $x_i \sqsubseteq a$ and $y_i \sqsubseteq b$ for all $i \in \{1, \ldots, n\}$. The same way we prove that $x_i \sqsubseteq c$ and $y_i \sqsubseteq d$ for all $i \in \{1, \ldots, n\}$. We deduce the following relations

$$\bigvee_{i=1}^{n} x_i \otimes y_i \quad \sqsubseteq \quad \left(\bigvee_{i=1}^{n} x_i \right) \otimes \left(\bigvee_{i=1}^{n} y_i \right) \quad \sqsubseteq \quad (a \wedge c) \otimes (b \wedge d)$$

and conclude that $(a \wedge c) \otimes (b \wedge d)$ is the greatest lower bound of $a \otimes b$ and $c \otimes d$ which gives the relation

$$(a \otimes b) \wedge (c \otimes d) \quad = \quad (a \wedge c) \otimes (b \wedge d)\ .$$

To prove that the tensor product $A \otimes B$ is a distributive lattice with difference, it remains to check that the difference operator is well-defined. First we prove that the difference between pure tensors exist and satisfies the following relation.

$$(a \otimes b) \backslash (c \otimes d) \quad = \quad \big((a \backslash c) \otimes b\big) \vee \big(a \otimes (b \backslash d)\big)$$

The general case will follow from Lemma 9.5.14 and 9.5.15. We evaluate the expression below.

$$(a \backslash c \otimes b) \qquad \vee \qquad \underbrace{(a \otimes b \backslash d)}_{(a \backslash c\ \otimes\ b \backslash d)\ \vee\ (a \wedge c\ \otimes\ b \backslash d)} \qquad \vee \qquad \underbrace{(a \otimes b \wedge c \otimes d)}_{(a \wedge c\ \otimes\ b \wedge d)}$$

The third term is rewritten applying the exchange law. The mapping $x \mapsto x \otimes b \backslash d$ preserves joins and a can be written as $a \backslash c \vee (a \wedge c)$ so the second term can be rewritten as above. By the same arguments we gather the first two terms and the last two ones in the next expression.

$$\underbrace{(a\backslash c \otimes b) \quad \vee \quad (a\backslash c \otimes b\backslash d)}_{a\backslash c\otimes b} \quad \vee \quad \underbrace{(a \wedge c \otimes b\backslash d) \quad \vee \quad (a \wedge c \otimes b \wedge d)}_{a\wedge c\otimes b}$$

The original expression thus boils down to $(a\backslash c \otimes b) \vee (a \wedge c \otimes b) = a \otimes b$. We also need to evaluate the following expression.

$$\big((a\backslash c \otimes b) \ \vee (a \otimes b\backslash d)\big) \ \wedge \ \underbrace{(a \otimes b \wedge c \otimes d)}_{(a\wedge c \ \otimes \ b\wedge d)}$$

By distributivity of ∧ over ∨ and and the exchange law it can be expressed as follows:

$$\big(\underbrace{(a\backslash c \wedge a \wedge c)}_{0}\otimes b \wedge d\big) \quad \vee \quad \big(a \wedge c \otimes \underbrace{(b\backslash d \wedge b \wedge d)}_{0}\big)$$

Therefore it is reduced to $(0 \otimes b \wedge d) \ \vee \ (a \wedge c \otimes 0)$. Since tensor products are taken in $\mathbf{SLat_0}$ the mappings $x \mapsto x \otimes b \wedge d$ and $y \mapsto a \wedge c \otimes y$ preserve zero, so the last expression is actually zero. We emphasize that the last argument does not hold in **SLat** – see Remark 9.5.21. It is the only place in the proof where this subtlety indeed matters, however, as shown by Example 9.5.18, it is crucial.

It remains to prove that the tensor product of distribute lattice with difference in $\mathbf{SLat_0}$ matches the ones in $\mathbf{DLat_0}$ and $\mathbf{DLat_d}$. As a consequence of the exchange law, the bimorphism $T_{\mathbf{SLat_0}}$ which sends (a, b) to the pure tensor $a \otimes b$ in $A \otimes_{\mathbf{SLat_0}} B$ is actually a bimorphism of distributive lattices with zero. Therefore we have a unique morphism $\sigma \in \mathbf{DLat_0}(A \otimes_{\mathbf{DLat_0}} B, A \otimes_{\mathbf{SLat_0}} B)$ such that $T_{\mathbf{SLat_0}} = \sigma \circ T_{\mathbf{DLat_0}}$. Of course we also have a unique morphism $\delta \in \mathbf{SLat_0}(A \otimes_{\mathbf{SLat_0}} B, A \otimes_{\mathbf{DLat_0}} B)$ such that $T_{\mathbf{DLat_0}} = \delta \circ T_{\mathbf{SLat_0}}$ because $T_{\mathbf{DLat_0}}$ is, in particular, a bimorphism of $\mathbf{SLat_0}$. From the universal properties satisfied by $T_{\mathbf{SLat_0}}$ and $T_{\mathbf{DLat_0}}$ we obtain that both morphisms $\delta \circ \sigma$ and $\sigma \circ \delta$ are identities of $\mathbf{SLat_0}$. From Remark 9.5.23 we deduce that both σ and δ are isomorphisms of distributive lattice with difference (resp. Boolean algebras). The isomorphism between the tensor product of distributive lattice with difference and that of distributive lattice with zero is given by Remark 9.5.5 and the fact that $\mathbf{DLat_d}$ is a full subcategory of $\mathbf{DLat_0}$ (*cf.* Remark 9.5.16). □

* * *

The next theorem is a generalization of a result by Haucourt and Ninin (2014). Given $n \in \mathbb{N}$ and for each $i \in \{1, \ldots, n\}$ a Boolean subalgebra $\mathcal{B}_i$ of some powerset $\mathrm{Pow}(E_i)$, we slightly extend Definition 6.2.1 allowing a block of dimension $n \in \mathbb{N}$ to be a subset of $E_1 \times \cdots \times E_n$ of the form $B_1 \times \cdots \times B_n$ with $B_i \in \mathcal{B}_i$. Consequently we write $\mathcal{R}_{\mathcal{B}_1,\ldots,\mathcal{B}_n}$ instead of $\mathcal{R}_n$.

Theorem 9.5.26. With the notation introduced before we have an isomorphism of Boolean algebras, the tensor product being taken in $\mathbf{SLat_0}$, $\mathbf{DLat_0}$, or $\mathbf{DLat_d}$.

$$\mathcal{R}_{\mathcal{B}_1,\ldots,\mathcal{B}_n} \quad \cong \quad \mathcal{B}_1 \otimes \cdots \otimes \mathcal{B}_n$$

Proof. By Proposition 6.2.7 and Theorem 6.2.21 we know that every element of $\mathcal{R}_{\mathcal{B}_1,\ldots,\mathcal{B}_n}$ is the finite union of its maximal blocks. Analogously by Remark 9.5.22 we know that every element of $\mathcal{B}_1 \otimes \cdots \otimes \mathcal{B}_n$ is the finite join of its maximal pure tensors. From Remark 9.5.21 we deduce that for tuples $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ with $a_i, b_i \in \mathcal{B}_i$ we have $a_1 \otimes \cdots \otimes a_n \sqsubseteq b_1 \otimes \cdots \otimes b_n$ if and only if $a_1 \times \cdots \times a_n \subseteq b_1 \times \cdots \times b_n$. As a consequence, the bijection between the pure tensors and the blocks extends to a bijection between $\mathcal{B}_1 \otimes \cdots \otimes \mathcal{B}_n$ and $\mathcal{R}_{\mathcal{B}_1,\ldots,\mathcal{B}_n}$ which exchanges the maximal pure tensors of an element of $\mathcal{B}_1 \otimes \cdots \otimes \mathcal{B}_n$ with the maximal blocks of the corresponding element of $\mathcal{R}_{\mathcal{B}_1,\ldots,\mathcal{B}_n}$. This bijection is readily a poset isomorphism. By Remark 9.5.24 that poset isomorphism is actually a semillatice isomorphism, and thus an isomorphism of Boolean algebras by Remark 9.5.23. □

As announced at the beginning of this section, our incursion in the realm of universal algebra is motivated by Corollary 9.5.27 which, in particular, expresses the Boolean algebra of n-dimensional isothetic regions in terms of the Boolean algebra of 1-dimensional isothetic regions. We recall that the isothetic subregions of an isothetic region A form a Boolean algebra denoted by $\mathcal{R}_A$. In particular, if $A_1, \ldots, A_n$ are isothetic regions, we can form their Cartesian product and consider the Boolean algebra $\mathcal{R}_{A_1 \times \cdots \times A_n}$. The latter can be expressed as the tensor product of the Boolean algebras $\mathcal{R}_{A_i}$ for $i \in \{1, \ldots, n\}$.

Corollary 9.5.27 (Theorem 4.1, Haucourt and Ninin (2014))**.** Given isothetic regions $A_1, \ldots, A_n$ we have the following equality.

$$\mathcal{R}_{A_1 \times \cdots \times A_n} \quad = \quad \mathcal{R}_{A_1} \otimes \cdots \otimes \mathcal{R}_{A_n}$$

Proof. Note that for all $i \in \{1, \ldots, n\}$, $\mathcal{R}_{A_i}$ is a Boolean subalgebras of $\mathrm{Pow}(A_i)$. Then apply Theorem 9.5.26 noting that $\mathcal{R}_{A_1 \times \cdots \times A_n}$ is actually $\mathcal{R}_{\mathcal{R}_{A_1},\ldots,\mathcal{R}_{A_n}}$. □

It is then natural to pay attention to the following conjecture.

Conjecture 9.5.28.
For all isothetic regions $X_1, \ldots, X_n, Y$, if $\mathcal{R}_Y \cong \mathcal{R}_{X_1} \otimes \cdots \otimes \mathcal{R}_{X_n}$ then $Y \cong X_1 \times \cdots \times X_n$.

Declaring $\mathcal{R}_X$ as **region-irreducible** if it cannot be written as a nontrivial tensor product of Boolean algebras of the form $\mathcal{R}_Y$ with Y being an isothetic region, we deduce from Conjecture 9.5.28 and Corollary 9.2.12 that

Conjecture 9.5.29. The Boolean algebra $\mathcal{R}_X$ has, up to terms reordering, a unique decomposition in region-irreducible terms.

A Boolean algebra is said to be **irreducible** when it cannot be written as a nontrivial tensor product of Boolean algebras. It is natural to ask whether Conjecture 9.5.29 remains valid replacing "region-irreducible" by "irreducible".

Conjecture 9.5.30. The Boolean algebra $\mathcal{B}_X$ has, up to terms reordering, a unique decomposition in irreducible terms.

As an illustration consider the singleton as a 0-dimensional isothetic region. It is the neutral element of the monoid of isothetic regions. The corresponding Boolean algebra of subregions is the two elements one (*i.e.* the neutral element for the tensor product). In fact we would like to go even further and establish the next conjecture, referring to suitably designed measures (*cf.* the discussion in Section 6.5).

Conjecture 9.5.31. For all isothetic regions $X_1, \ldots, X_n, Y$, we have the following equivalence.

$$Y = X_1 \times \cdots \times X_n \qquad \Leftrightarrow \qquad \mu_Y = \mu_{X_1} \otimes \cdots \otimes \mu_{X_n}$$

9.6 Metrics

As a broad generalization of a classic result by Georges de Rham (1952), Foertsch and Lytchak (2008) provides a unique decomposition result for geodesic metric spaces of finite affine rank. The affine rank of a region boils down to the greatest value of n such that the hypercube $[0,1]^n$ can be embedded in it. As one can guess, the affine rank of a region of shape $|G_1|, \ldots, |G_n|$ is at most n. The formal definition of the affine rank depends on the topological dimension – see (Pears, 1975, Chap.3) or (Engelking, 1978, Chap.1). In the context of this chapter, it is therefore tempting to apply this result to the metric spaces we have defined over regions in Section 6.5. Unfortunately these metric spaces may not be geodesic though they are length metrics (*cf.* Example 6.5.4 and its introductory discussion). Yet, assuming the result of Foertsch and Lytchak (2008) still holds for length-metric spaces, we would like to compare the prime decomposition of a region with the prime decomposition of its metrics.

10

Perspectives

10.1 Implementation

Several theoretical results presented in this memoir are implemented in the static analyzer ALCOOL whose input language is Paml. From a mathematical point of view ALCOOL is based on the notion of continuous models (*cf.* Definition 7.1.2). The ALCOOL software is entirely written in OCaml and has grown at CEA from 2006 to 2014, up to 40kloc. A preceding prototype had been developed (in C) by Éric Goubault to initiate a partnership with EDF which has partly funded the development of the current version. The results obtained from this collaboration are gathered in Bonichon et al. (2011). Given a Paml program P the ALCOOL software is currently able to:

- build the continuous model,
- compute the deadlock attractor,
- factorize the continuous model,

and provided the continuous model is loop-free, it can also:

- find all the dihomotopy classes,
- compute the category of components, and
- generate the Čech complex associated with the continuous model.

10.2 Does Model Category Fit with Directed Topology ?

The compact unit interval plays a fundamental role in algebraic topology. It can be seen as the building block of the whole theory. This statement can be made formal in the context of path functors and P-categories (resp. cylinder functors and I-categories) – see Baues (1989). In **Top** for example, the path functor P and the cylinder functor I are the following ones.

$$P : X \mapsto X^{[0,1]} \qquad I : X \mapsto X \times [0,1]$$

It is well known that $I \dashv P$ and that both induce fibrations and cofibrations for a standard model structure on **Top**. Still following Baues (1989) a path-object P (resp. a cylinder object I) induces a P-category (resp. I-category) structure, with P and I (the path functor and the cylinder functor) satisfying ad hoc properties. In most cases they are obtained as $(\_)^P$ and $(\_ \times I)$ both being related when I is exponentiable and isomorphic to P, which is the case in algebraic topology. In addition any P-category (resp. I-category) induces a fibration (resp. cofibration) category, and when both structures coexist in a compatible way, we have a model category – Baues (1989). With respect to the machinery provided by model category theory, it would suffice to find a satisfactory directed counterpart of the compact unit interval to provide algebraic directed topology with a firm foundation. Loosely speaking, the only reasonable choice is the compact unit interval together with the standard order over real numbers. The constructions $(\_ \times ▯)$ and $(\_)^{▯}$ make sense in all the categories described in Chapter 4, moreover they enjoy all the nice properties one can expect from them in a directed context. This is one of the motivations for Marco Grandis's work on algebraic directed topology (Grandis (2009)). Unfortunately there is a glitch, the functors $(\_ \times ▯)$ and $(\_)^{▯}$ do not induce a cofibration category nor a fibration one. Let us see why.

In directed topology, the notion of dihomotopy equivalence is the expected counterpart of the classical homotopy equivalences. A naive approach consists of substituting directed spaces and their morphisms to topological spaces and continuous maps in the standard definition. But following that way, all the directed stars (*cf.* Example 4.1.15) are equivalent with one another which is, from a computer science point of view, definitely prohibitive. In fact, there is not even a clear consensus about the properties that dihomotopy equivalences should enjoy. This fact actually derives from a more general issue. If we stick to the model category setting of Quillen (1967) (see also Hovey (1999); May and Ponto (2010)) the dihomotopy equivalences should come with fibrations and cofibrations. In particular the trivial cofibrations (*i.e.* the ones that are also weak equivalences) are closed under pushout: if the pushout of a (trivial) cofibration along a given morphism exists then it is still a (trivial) cofibration. As a consequence if the dimap $\{0\} \hookrightarrow [0, 1]$ is a trivial cofibration (which is a trivial fact considering one of the standard model categories over **Top**) then, once again, all the directed stars are equivalent. For example Roman Bruckner (2015) has proven that the full subcategory of **Cat** whose objects are the loop-free categories inherits from the Thomason model structure on **Cat**, and Bubenik and Worytkiewicz (2006) provide a certain category of local pospaces[1] with a model category structure yet it is not clear how the latter differs from the standard model structure on **Top**. The preceding remarks suggest that model category theory just ignores direction.

A reasonable way to circumvent the problem is to restrict the collection of weak equivalences by imposing an additional constraint. For model structures in which weak equivalences arise as a byproduct of a path (resp. cylinder) functor, a common way to express such constraints is the notion of relative homotopy. In other words one requires that directed homotopies (which are mappings) be constant over some fixed directed subspace. For example Thomas Kahl (2006) has followed this way to equip **PoTop** (his definition of pospace slightly differs from our's in that the partial order is not supposed to be closed) with both a P-category structure and an I-category structure. In (Kahl (2009)), he also exhibits a fibration category structure on the slice under any given local pospace[1]. Peter Bubenik (2009) has also followed this approach and advocated for it. However, it somewhat sweeps the dust under the carpet since it leaves all latitude about

[1] The definition of local pospace they use differ from the one given in Section 4.3.

the choice of the subspace that should be preserved.

The model category setting might therefore be not compatible with directed topology. An alternative one could be provided by the notion of homotopical category, which is obtained by relaxing the axioms of model categories – see the second part of Dwyer et al. (2004). A category C together with a distinguished class of morphisms Σ is said to be homotopical when Σ contains all the identities and satisfies the 2-out-of-6 property: for all morphisms α, β, and γ if both $\gamma\beta$ and $\beta\alpha$ exist and belong to Σ, then so do α, β, γ, and $\gamma\beta\alpha$. It is funny to remark that any loop-free category with any system of weak isomorphisms is a homotopical category (*cf.* Section 8.6). Looking back to the previous paragraph, the problem of choosing a "good" subspace to be preserved seems to be tightly related to categories of components. The collection of directed weak equivalences (in the preceding naive sense) that induces isomorphisms of categories of components could lead to a satisfactory homotopical category (at least in the case of **PoTop**).

10.3 About Homology of Directed Spaces ?

The classification of spaces provided by homotopy is finer than the one resulting from homology, but unlike homotopy groups, which are notoriously hard to determine, homology groups are so tractable that their computation can even be automated, being finally reduced to linear algebra. For example the k^{th} homology group of the n-dimensional sphere is either $\mathbb{Z}$ or null according to whether $k = n$ or not, while there are entire books dedicated to the computation of higher homotopy groups of spheres (*e.g.* Ravenel (2003)). Nevertheless, in many cases, homology suffices to distinguish a space from another. Since homological algebra is an extremely supple tool that is pervasively used in mathematics (see Mac Lane (1995); Cartan and Eilenberg (1999); Weibel (1994)) one would like to apply it to directed topology. Basically, given any functor towards the category of simplicial sets, one defines the homology of an object as the homology of its image by the functor. The whole problem being then to understand what it classifies. Applying this to directed topology, we obtain a classification that might have the same defect as a hypothetical homotopy of directed spaces. Indeed it is thus natural to take as standard simplices the directed convex subsets of $\mathbb{R}^n$ generated by the following points for $k \in \{1, \ldots, n\}$.

$$p_k = (\underbrace{1, \ldots, 1}_{k}, \underbrace{0, \ldots, 0}_{n-k})$$

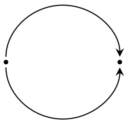
However, the homology obtained that way does not distinguish between the directed stars. In fact we conjecture that the construction amounts, at least when X is the continuous model of Paml program (*cf.* Definition 7.1.2), to the singular homology of its underlying topological space. Moreover it is not clear whether another choice of directed simplices would alter the resulting homology: consider for example the simplices generated by the following points.

$$p'_k = (\underbrace{0, \ldots, 0}_{k-1}, 1, \underbrace{0, \ldots, 0}_{n-k}) \ .$$

In spite of these obstacles, many notions of directed homology have been proposed: Goubault (1995), Fahrenberg (2004), Grandis (2005, 2009), Gaucher (2005, 2006), Husainov (2013), Dubut et al. (2015).

10.4 A Glance at Directed Universal Coverings

The exponential map $\mathbb{R} \to \mathbb{S}^1$ is the universal covering of the circle in the sense that any path on $\mathbb{S}^1$ can be lifted to a path on $\mathbb{R}$ in a unique way provided one fixes basepoints. Any connected, locally path-connected and semi-locally simply connected X has a universal covering (*i.e.* a covering $p : E \to X$ with E being simply connected in addition with the three previously mentioned properties of X). This well-known construction is obtained by unfolding the loops. If we endow $\mathbb{R}$ and $\mathbb{S}^1$ with their standard directed structures, the exponential map is actually a dimap and it enjoys the same lifting properties than in the undirected case. This heavenly case could let us expect for an easy notion of directed universal covering. Things are not so simple and the example of the circle should be regarded as extremely misleading. The holes detected by the first homotopy groups, that is non-nullhomotopic loops, have nothing to do with directed loops. First observe that the underlying space of the d-space realization of the following graph is the circle though it does not contain any directed loop.



Therefore it should be its own directed universal covering. The other way round, the underlying space of the directed complex plane is simply connected though it contains directed loops that are not dihomotopic with a constant path. In fact its directed universal covering should be given by the mapping that sends $(\rho, \theta) \in \mathbb{R}_+ \times \mathbb{R}$ to $\rho e^{i\theta}$ in $\overrightarrow{\mathbb{C}}$. The "right" directed universal covering is an issue even for the d-space realization of graphs. The question have been studied in the following papers: Goubault et al. (2009), Fajstrup (2003, 2005 / 2006, 2008, 2011).

10.5 Finding Linear Representations of Fundamental Categories

During his doctoral studies, Nicolas Ninin has worked on the linear representations of small categories. The problem is well-known for categories freely generated by graphs, which are, in this context, preferably called **quivers**. In this case indeed, Gabriel (1972) gave a complete classification of those quivers having finitely many isomorphism classes of indecomposable linear representations. More details can be found in Gabriel (1975), (Hazewinkel et al., 2007, Chap. 2) or (Assem et al., 2006, Chap. II and VII). The class of categories to consider is therefore much broader but the aim is also much more modest: for computational and applied purposes, we want to provide certain small categories (at least the category of components of continuous models of Paml programs) with a tractable faithful representation. Basically, we would like to label the generating morphisms of any category with a least presentation (*cf.* Definition 9.4.7) with matrices so that two composable sequences of irreducible elements of it are equal iff they have the same source and the same target and the products of their labels are equal. We also require that the entries of the labelling matrices are taken in a tractable (semi)ring. Proposition 9.4.6 suggests that we focus on finite loop-free categories first, but even in this case there is no obvious solution.

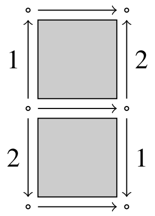


Figure 10.1: Directed Möbius strip as a precubical set

10.6 Locally Star-Shaped Pospaces

We have seen that $\mathcal{X}^n$ (with G some essentially finite graph and $\mathcal{X} = \rceil G \lfloor$) behaves much like $\mathbb{R}^n$. In mathematics, the study of spaces that locally look like $\mathbb{R}^n$ has proven to be very fruitful. From this observation we introduce the local star-shaped pospaces.

Definition 10.6.1. A local pospace (*cf.* Definition 4.3.17) is said to be **star-shaped** when it admits an atlas whose charts (*cf.* Definition 4.3.1) are finite products of open stars (*cf.* Example 4.1.15).

Example 10.6.2. The local pospace $\mathcal{X}^n$ is star-shaped.

Example 10.6.3. A directed version of the Möbius band is obtained as the realization of the precubical set on Figure 10.1 identifying arrows labelled with 1 and 2 accordingly. One can check that it is star-shaped though not isomorphic to $\rceil G_1 \lfloor \times \rceil G_2 \lfloor$.

A classical problem in mathematics consists of determining how far a space is from being a nontrivial Cartesian product of spaces. When the spaces of interest are manifolds, the answer arises from the study of **vector bundles**, the tangent bundle of a manifold being the prototypical example – see (Husemoller, 1993, Chap. 3) or (Lang, 1999, Chap. III). The proof that any region admits a unique decomposition derives from a rather combinatorial approach that would not fit with random local star-shaped pospaces. Yet, the more general notion **fiber bundle** (Husemoller, 1993, Chap. 4) could be a relevant candidate for an adaptation to directed topology.

Bibliography

Aho, A.V., Lam, M.S., Sethi, R., and Ullman, J.D. Compilers: Principles, Techniques, and Tools (Addison-Wesley 2007), 2nd ed. ISBN 0-321-48681-1. 11, 17

Aliprantis, C.D. and Border, K.C. Infinite Dimensional Analysis: A Hitchhiker's Guide (Springer 2006), 3rd ed. 69

Allen, F.E. Control Flow Analysis. In Proceedings of a Symposium on Compiler Optimization, pp. 1–19 (ACM, New York, NY, USA 1970). doi:10.1145/800028.808479. 4, 49

Amadio, R.M. and Curien, P.L. Domains and Lambda-Calculi. Cambridge Tracts in Theoretical Computer Science (Cambridge University Press 1998). 209

Anderson, D.F. Factorization in Integral Domains. In D. Anderson (ed.), Factorization in Integral Domains, Lecture Notes in Pure and Applied Mathematics (CRC Press 1997). ISBN 9780824700324. 196

Anderson, J.A. and Kimura, N. The tensor product of semilattices. Semigroup Forum 16:83–88 1978. doi:10.1007/BF02194615. 216

Antolini, R. Cubical Structures, Homotopy Theory. Annali di Matematica pura ed applicata CLXXVIII(IV):317–324 2000. 44

Antolini, R. Geometric Realisations of Cubical Sets with Connections, and Classifying Spaces of Categories. Applied Categorical Structures 10(5):481–494 2002. doi:10.1023/A:1020506404904. 44

Assem, I., Simson, D., and Skowronski, A. Elements of the Representation Theory of Associative Algebras: Volume 1: Techniques of Representation Theory, vol. 65 of *London Mathematical Society Student Texts* (Cambridge University Press 2006). ISBN 978-0521586313. 226

Balabonski, T. Concurrence, géométrie, et factorisation de catégories. Master's thesis, École Normale Supérieure de Lyon 2007. 203

Balabonski, T. and Haucourt, E. A Geometric Approach to the problem of Unique Decomposition of Processes. In Concurrency Theory 21th International Conference, vol. 6269 of *Lecture Notes in Computer Science*, pp. 132–146 (Springer 2010). doi:10.1007/978-3-642-15375-4_10. 7, 163, 164, 198

Baues, H.J. Algebraic Homotopy. Cambridge Studies in Advanced Mathematics (Cambridge University Press 1989). 223, 224

Bednarczyk, M.A., Borzyszkowski, A.M., and Pawlowski, W. Generalized Congruences – Epimorphisms in Cat . Theory and Applications of Categories 5(11):266–280 1999. 165, 172

Beer, G. Topologies on Closed and Closed Convex Sets, vol. 268 of *Mathematics and Its Applications* (Kluwer Academic Publishers 1993). ISBN 0-7923-2531-1. 69

Bell, A.M., Brown, M.R., and Fraser, G.A. The tensor product of distributive lattices: Structural results. Proceedings of the Edinburgh Mathematical Society 27:237–245 1984. 218

Berry, G. and Boudol, G. The chemical abstract machine. In Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, POPL '90, pp. 81–94 (ACM, New York, NY, USA 1990). ISBN 0-89791-343-4. 193

Berry, G. and Boudol, G. The chemical abstract machine. Theoretical Computer Science 96(1):217–248 1992. 193

Birkhoff, G. Lattice Theory, vol. 25 of *Colloquium Publications* (American Mathematical Society 1967), 3rd ed. 209, 210, 215

Bishop, R.L. and Goldberg, S.I. Tensor Analysis on Manifolds (Dover Publications 1980). ISBN 0-486-64039-6. 102, 104

Blyth, T. Lattices and Ordered Algebraic Structures Lattices and Ordered Algebraic Structures. Universitext (Springer 2005). ISBN 978-1-84628-127-3. 194, 209

Bogachev, V.I. Measure Theory vol. I (Springer 2007). ISBN 978-3-540-34514-5. 152

Bonichon, R., Canet, G., Correnson, L., Goubault, É., Haucourt, E., Hirschowitz, M., Labbé, S., and Mimram, S. Rigorous Evidence of Freedom from Concurrency Faults in Industrial Control Software. In Computer Safety, Reliability, and Security, vol. 6894 of *Lecture Notes in Computer Science*, pp. 85–98 (Springer 2011). doi: 10.1007/978-3-642-24270-0_7. 223

Borceux, F. Handbook of Categorical Algebra, I. Basic Category Theory, vol. 50 of *Encyclopedia of Mathematics and its Applications* (Cambridge University Press 1994a). ISBN 0-521-44178-1. 33, 36, 37, 45, 68, 73, 101, 108, 114, 117, 165, 166, 168, 172, 180, 185, 212

Borceux, F. Handbook of Categorical Algebra, II. Categories and Structures, vol. 51 of *Encyclopedia of Mathematics and its Applications* (Cambridge University Press 1994b). ISBN 0-521-44179-X. 39, 76, 206, 207

Borceux, F. Handbook of Categorical Algebra, III. Categories of Sheaves, vol. 52 of *Encyclopedia of Mathematics and its Applications* (Cambridge University Press 1994c). ISBN 0-521-44180-3. 34, 169

Bressoud, D.M. Factorization and Primality Testing. Undergraduate Texts in Mathematics (Springer 1989). ISBN 978-0387970400. 195

Brešar, M. Introduction to Noncommutative Algebra. Universitext (Springer 2014). ISBN 978-3-319-08692-7. 208

Bridson, M.R. and Haefliger, A. Metric Spaces of Non-Positive Curvature, vol. 319 of *Grundlehren der mathematischen Wissenschaften* (Springer 1999). 122, 150, 151

Brown, R. Topology and Groupoids (BookSurge Publishing 2006). 7, 107, 111, 116, 117

Brown, R. and Higgins, P.J. On the algebra of cubes. Journal of Pure and Applied Algebra 21:233–260 1981. 44, 45

Brown, R., Higgins, P.J., and Sivera, R. Nonabelian Algebraic Topology. Tracts in Mathematics (European Mathematical Society 2011). 36, 41, 44, 45, 148

Bruckner, R. A Model Structure on the Category of Small Acyclic Categories. arXiv 2015. 224

Bubenik, P. Context for Models of Concurrency. In Proceedings of the Workshops on Geometric and Topological Methods in Concurrency Theory (GETCO 2004+2005+2006), vol. 230 of *Electronic Notes in Theoretical Computer Science*, pp. 3–21 (Elsevier 2009). doi:10.1016/j.entcs.2009.02.014. 224

Bubenik, P. and Worytkiewicz, K. A model category for local pospaces. Homology, Homotopy and Applications 8(1):263–292 2006. 224

Burago, D., Burago, Y., and Ivanov, S. A Course in Metric Geometry, vol. 33 of *Graduate Studies in Mathematics* (American Mathematical Society 2001). ISBN 978-0821821299. 32

Burris, S. and Sankappanavar, H.P. A Course in Universal Algebra, vol. 78 of *Graduate Texts in Mathematics* (Springer 1981), millenium ed. ISBN 978-0387905785. 206

Carson, S.D. and Reynolds Jr., P.F. The Geometry of Semaphore Programs. ACM Transactions on Programming Languages and Systems 9(1):25–53 1987. doi:10.1145/9758.9759. 6

Cartan, H. and Eilenberg, S. Homological Algebra, vol. 19 of *Princeton Mathematical Series* (Princeton University Press 1999). ISBN 9780691049915. 225

Cattani, G.L. and Sassone, V. Higher Dimensional Transition Systems. In 11th Symposium of Logics in Computer Science, LICS ‘96, pp. 55–62 (IEEE 1996). 20

Chandy, K.M. and Misra, J. The Drinking Philosophers Problem. ACM Transactions on Programming Languages and Systems 6(4):632–646 1984. 16

Cisinski, D.C. Les préfaisceaux comme modèles des types d’homotopie. Astérisque (Société Mathématique de France 2006). 31

Clerc, F. and Mimram, S. Presenting a Category Modulo a Rewriting System. In M. Fernández (ed.), 26th International Conference on Rewriting Techniques and Applications (RTA 2015), vol. 36 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 89–105 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany 2015). ISBN 978-3-939897-85-9. ISSN 1868-8969. doi:10.4230/LIPIcs.RTA.2015.89. 181

Coffman, E.G., Elphick, M., and Shoshani, A. System Deadlocks. ACM Comput. Surv. 3(2):67–78 1971. ISSN 0360-0300. doi:10.1145/356586.356588. 6, 157

Cohn, D.L. Measure Theory (Birkhäuser 1980). 152

Cohn, P.M. Basic Algebra: Groups, Rings and Fields (Springer 2003), 2nd ed. ISBN 978-1-4471-1060-6. 208

Conlon, L. Differentiable Manifolds. Modern Birkhäuser Classics (Birkhäuser 2008), 2nd ed. 104

Cooper, K. and Torczon, L. Engineering a compiler (Morgan Kaufmann 2011), 2nd ed. ISBN 978-0120884780. 14, 52

Crans, S.E. On combinatorial models for higher dimensional homotopies. Ph.D. thesis, Universiteit Utrecht 1995. 41

Cridlig, R. Semantic Analysis of Shared-memory Concurrent Languages Using Abstract Model-checking. In Proceedings of the 1995 ACM SIGPLAN Symposium on Partial Evaluation and Semantics-based Program Manipulation, PEPM '95, pp. 214–225 (ACM, New York, NY, USA 1995). ISBN 0-89791-720-0. doi:10.1145/215465.215593. 10

Cridlig, R. Implementing a Static Analyzer of Concurrent Programs: Problems and Perspectives. In M. Dam (ed.), Analysis and Verification of Multiple-Agent Languages, selected papers from the 5th LOMAPS, June 24-26, 1996, vol. 1192, pp. 244–259 (Springer, London, UK 1997). ISBN 3-540-62503-8. 10

Cridlig, R. and Goubault, É. Semantics and analysis of Linda-based languages. In P. Cousot, M. Falaschi, G. Filé, and A. Rauzy (eds.), Static Analysis: Third International Workshop, WSA '93 Padova, Italy, September 22–24, 1993 Proceedings, vol. 724 of *Lecture Notes in Computer Science*, pp. 72–86 (Springer 1993). ISBN 978-3-540-57264-0. ISSN 0302-9743. 6

Cristescu, I.D., Krivine, J., and Varacca, D. A compositional semantics for the reversible pi-calculus. In Logic in Computer Science, pp. 388–397 (New Orleans, United States 2013). doi:10.1109/LICS.2013.45. 118

Danos, V., Krivine, J., and Sobociński, P. General Reversibility. Electronic Notes in Theoretical Computer Science 175:75–86 2007. 118

Davey, B.A. and Priestley, H.A. Introduction to Lattices and Order (Cambridge University Press 2002), 2nd ed. ISBN 9780521784511. 94

De Lellis, C. Rectifiable Sets, Densities and Tangent Measures. Zurich Lectures in Advanced Mathematics (European Mathematical Society 2008). ISBN 978-3-03719-044-9. 153

de Rham, G. Sur la réductibilité d'un espace de Riemann. Commentarii Mathematici Helvetici 26:328–344 1952. 222

Diestel, R. and Kühn, D. Graph-theoretical versus topological ends of graphs. Journal of Combinatorial Theory 87:197–206 2003. 134

Dijkstra, E.W. Cooperating sequential processes. Tech. rep., Technological University, Eindhoven, The Netherlands, september 1965. Reprinted in F. Genuys Ed. 1968. *Programming Languages*, Academic Press, New York, 43-112. *Article 1.* 1

Dijkstra, E.W. Cooperating sequential processes. In F. Genuys (ed.), Programming Languages: NATO Advanced Study Institute, proceedings of the summer school held at Villars-de-Lans, 1966, pp. 43–112 (Academic Press 1968). ISBN 0-12-279750-7. Reprint of the eponymous technical report published in september 1965 by the Technological University of Eindhoven, The Netherlands. 6, 7, 8, 10, 16, 21, 67

Dijkstra, E.W. Hierarchical Ordering of Sequential Processes. Acta Informatica 1:115–138 1971. 157

Douady, A. and Douady, R. Algèbre et Théories Galoisiennes. Nouvelle Bibliothèque Mathématique (Cassini 1999). ISBN 978-2842250058. 208, 209

Dreier, J., Ene, C., Lafourcade, P., and Lakhnech, Y. On Unique Decomposition of Processes in the Applied Pi-Calculus. In FOSSACS 2013 : 16th International Conference on Foundations of Software Science and Computation Structures (2013). 7, 194

Dubut, J., Goubault, É., and Goubault-Larrecq, J. Natural Homology. In M.M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann (eds.), Automata, Languages, and Programming: 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II, vol. 9135, pp. 171–183 (Springer 2015). doi: 10.1007/978-3-662-47666-6_14. 225

Dwyer, W.G., Hirschhorn, P.S., Kan, D.M., and Smith, J.H. Homotopy Limit Functors on Model Categories and Homotopical Categories, vol. 113 of *Mathematical Surveys and Monographs* (American Mathematical Society 2004). ISBN 978-0-8218-3975-1. 7, 38, 186, 225

Engelking, R. Dimension theory (North-Holland 1978). 222

Engelking, R. General Topology (Heldermann 1989), revised and completed edition ed. ISBN 3-88538-006-4. 39, 72

Epstein, D.B. Word Processing in Groups (Jones and Bartlett 1992). 150, 191

Evans, L.C. and Gariepy, R.F. Measure Theory and Fine Properties of Functions. Studies in Advanced Mathematics (CRC Press 1992). ISBN 0-8493-7157-0. 153

Fahrenberg, U. Towards an Efficient Algorithm for Detecting Unsafe States in Timed Concurrent Sytems. Master's thesis, The Faculty of Engineering and Science 2002. 57, 59

Fahrenberg, U. Directed Homology. Electronic Notes in Theoretical Computer Science 100:111–125 2004. 225

Fahrenberg, U. and Legay, A. Partial Higher-Dimensional Automata. In L.S. Moss and P. Sobociński (eds.), 6th Conference on Algebra and Coalgebra in Computer Science (CALCO 2015), vol. 35 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 101–115 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany 2015). ISBN 978-3-939897-84-2. ISSN 1868-8969. doi: 10.4230/LIPIcs.CALCO.2015.101. 48

Fahrenberg, U. and Raußen, M. Reparametrizations of continuous paths. Journal of Homotopy and Related Structures 2(2):93–117 2007. 99, 119

Fajstrup, L. Dicovering Spaces. Homology, Homotopy and Applications 5(2):1–17 2003. 226

Fajstrup, L. Dipaths and dihomotopies in a cubical complex. Advances in Applied Mathematics 35(2):188–206 2005. 111, 117, 159

Fajstrup, L. Dicovering as quotients. Tech. rep., Institut Mittag-Leffler 2005 / 2006. 226

Fajstrup, L. On directed coverings. Tech. rep., Aablorg university 2008. 226

Fajstrup, L. Erratum to 'Dicovering spaces'. Homology, Homotopy and Applications 13(1):403–406 2011. 226

Fajstrup, L., Goubault, É., Haucourt, E., Mimram, S., and Raußen, M. Trace Spaces: An Efficient New Technique for State-Space Reduction. In H. Seidl (ed.), ESOP, vol. 7211 of *Lecture Notes in Computer Science*, pp. 274–294 (Springer 2012). ISBN 978-3-642-28868-5. 119

Fajstrup, L., Goubault, É., Haucourt, E., Mimram, S., and Raußen, M. Directed Algebraic Topology and Concurrency. Springer Briefs in Applied Science and Technology - Mathematical Methods (Springer 2016). ISBN 978-3-319-15397-1. 119

Fajstrup, L., Goubault, É., Haucourt, E., and Raußen, M. Components of the Fundamental Category. Applied Categorical Structures 12(1):81–108 2004. doi:10.1023/B:APCS.0000013812.75342.de. 7, 165

Fajstrup, L., Goubault, É., and Raußen, M. Directed Algebraic Topology and Concurrency. Tech. rep., Aablorg university, DK-9220 Aalborg Øst 1999. 110

Fajstrup, L., Goubault, É., and Raußen, M. Algebraic Topology and Concurrency. Theoretical Computer Science 357(1):241–278 2006. doi:10.1016/j.tcs.2006.03.022. Presented at the 14th conference on the Mathematical Foundations of Programming Semantics (10-13 May 1998, Queen Mary - Westfield College, London, England). 7, 79, 80, 89, 121

Fajstrup, L. and Rosický, J. A convenient category for directed homotopy. Theory and Applications of Categories 21(1):7–20 2008. 105

Federer, H. Geometric measure theory. Classics in Mathematics (Springer 2013), reprint of the 1st ed. 1969 ed. 152, 153

Floyd, R.W. Assigning meanings to programs. In J.T. Schwartz (ed.), Mathematical Aspects of Computer Science, vol. 19 of *Proceedings of Symposia in Applied Mathematics*, pp. 19–32 (American Mathematical Society 1967). ISBN 978-0821813195. 5, 49

Foertsch, T. and Lytchak, A. The De Rham Decomposition Theorem for Metric Spaces. Geometric and Functional Analysis 18:120–143 2008. 7, 222

Folland, G.B. Real Analysis Modern Techniques and Their Applications (Addison-Wesley 1999), 2nd ed. 152

Fournet, C. and Gonthier, G. The Join Calculus: A Language for Distributed Mobile Programming. In Applied Semantics, International Summer School, APPSEM 2000, Caminha, Portugal, September 9-15, 2000, Advanced Lectures (2000). 193

Fournet, C., Le Fessant, F., Maranget, L., and Schmitt, A. JoCaml: A Language for Concurrent Distributed and Mobile Programming. In J. Jeuring and S. Jones (eds.), Advanced Functional Programming, vol. 2638 of *Lecture Notes in Computer Science*, pp. 129–158 (Springer 2003). ISBN 978-3-540-40132-2. doi:10.1007/978-3-540-44833-4_5. 15

Fraser, G.A. The semilattice tensor product of distributive lattices. Transactions of the American Mathematical Society 217 1976a. doi:10.1090/S0002-9947-1976-0392728-8. 216, 217

Fraser, G.A. The tensor product of distributive lattices. Proceedings of the Edinburgh Mathematical Society 20(2) 1976b. doi:10.1017/S0013091500010622. 216

Fraser, G.A. The tensor product of semilattices. Algebra Universalis 8:1–3 1978. doi:10.1007/BF02485362. 216, 217

Fremlin, D.H. Measure Theory: The Irreducible Minimum (Torres Fremlin 2011), 2nd ed. ISBN 978-0953812905. 152

Fröschle, S. and Lasota, S. Normed Processes, Unique Decomposition, and Complexity of Bisimulation Equivalences. Electronic Notes in Theoretical Computer Science 239:17–42 2009. 194

Gabriel, P. Unzerlegbare Darstellungen I. Manuscripta Mathematica 6:71–103 1972. 226

Gabriel, P. Représentation Indécomposable. In Séminaire Bourbaki (1973-74), vol. 431 of *Lecture Notes in Mathematics*, pp. 143–211 (Springer 1975). 226

Gabriel, P. and Zisman, M. Calculus of Fractions and Homotopy Theory. Ergebnisse der Mathematik und ihrer Grenzgebiete (Springer 1967). 36, 37, 38, 39, 117

Gale, D. Compact sets of functions and function rings. Proceeding of the American Mathematical Society 1:303–308 1950. 39

Gaucher, P. A model category for the Homotopy Theory of Concurrency. Homology, Homotopy and Applications 5(1):549–599 2003. 105

Gaucher, P. Homological properties of non-deterministic branchings and mergings in Higher Dimensional Automata. Homology, Homotopy and Applications 7(1):51–76 2005. 225

Gaucher, P. T-homotopy and refinement of observation (III) : Invariance of the branching and merging homologies. New York J. Math. 12:319–348 2006. 225

Gaucher, P. Towards a Homotopy Theory of Process Algebra. Homology, Homotopy and Applications 10(1):353–388 2008. 105

Gaucher, P. Combinatorics of labelling in Higher Dimensional Automata. Theoretical Computer Science 411(11-13):1452–1483 2010a. 105

Gaucher, P. Directed algebraic topology and higher dimensional transition systems. New York J. Math. 16:409–461 (electronic) 2010b. 105

Géraud, R. Fundamental categories and their enveloping groupoids. Master's thesis, École Centrale Paris 2012. 117

Geroldinger, A. and Halter-Koch, F. Non-Unique Factorizations: Algebraic, Combinatorial, and Analytic Theory. Pure and Applied Mathematics (Chapman & Hall 2006). ISBN 978-1-58488-576-4. 195

Gierz, G., Hofmann, K.H., Keimel, K., Lawson, J.D., Mislove, M.W., and Scott, D.S. Continuous Lattices and Domains, vol. 93 of *Encyclopedia of Mathematics and its Applications* (Cambridge University Press 2003). ISBN 978-0521803380. 209

Givant, S. and Halmos, P. Introduction to Boolean Algebras. Undergraduate Texts in Mathematics (Springer 2009). ISBN 978-0-387-40293-2. 134, 205, 206, 210, 213

Goerss, P.G. and Jardine, J.F. Simplicial Homotopy Theory, vol. 174 of *Progress in Mathematics* (Birkhäuser 1999). 36, 37, 39

Goldblatt, R. Topoi: The Categorial Analysis of Logic (Dover Publications 1984), reprint of the north-holland 1983 edition ed. 38

Goubault, É. Domains of higher-dimensional automata. In E. Best (ed.), CONCUR'93: 4th Intrenational Conference on Concurrency Theory Hildesheim, Germany, August 23–26, 1993 Proceedings, vol. 715 of *Lecture Notes in Computer Science*, pp. 293–307 (Springer 1993). ISBN 978-3-540-57208-4. doi:10.1007/3-540-57208-2_21. 6

Goubault, É. The Geometry of Concurrency. Ph.D. thesis, Ecole Normale Supérieure 1995. 225

Goubault, É. Geometry and Concurrency: A user's guide. Mathematical Structures in Computer Science 10(4):411–425 2000. 6

Goubault, É. Some Geometric Perspectives in Concurrency Theory. Homology, Homotopy and Applications 5(2):95–136 2003. 6

Goubault, É. and Haucourt, E. A practical application of Geometric semantics to Static analysis of Concurrent programs. In Concurrency Theory 16th International Conference, vol. 3653 of *Lecture Notes in Computer Science*, pp. 503–517 (Springer 2005). doi:10.1007/11539452_38. 165

Goubault, É. and Haucourt, E. Components of the Fundamental Category II. Applied Categorical Structures 15(4):387–414 2007. doi:10.1007/s10485-007-9082-7. 165

Goubault, É., Haucourt, E., and Krishnan, S. Covering Space Theory for the Directed Topology. Theory and Applications of Categories 22(9):252–268 2009. ISSN 1201 - 561X. 226

Goubault, É., Haucourt, E., and Krishnan, S. Future Path-components in Directed Topology. Electr. Notes Theor. Comput. Sci. 265:325–335 2010. 165

Goubault, É. and Jensen, T.P. Homology of Higher-Dimensional Automata. In Proc. of 3rd International Conference on Concurrency Theory CONCUR, no. 630 in Lecture Notes in Computer Science (Springer 1992). 6

Goubault, É. and Mimram, S. Formal Relationships Between Geometrical and Classical Models for Concurrency. Electronic Notes in Theoretical Computer Science 283:77–109 2012. 6, 45

Goubault-Larrecq, J. Non-Hausdorff Topology and Domain Theory: Selected Topics in Point-Set Topology, vol. 22 of *New Mathematical Monographs* (Cambridge University Press 2013). ISBN 978-1107034136. 39, 209

Goubault-Larrecq, J. Exponentiable Streams and Prestreams. Applied Categorical Structures 22:514–549 2014. doi:10.1007/s10485-013-9315-x. 93

Grandis, M. Directed Homotopy Theory, I. The Fundamental Category. Cahiers de Topologie et Géométrie Différentielle Catégoriques 44(4):281–316 2003. ISSN 0008-0004. 7, 93

Grandis, M. Directed Combinatorial Homology and Noncommutative Tori. Math. Proc. Cambridge Philos. Soc. 138:233–262 2005. 225

Grandis, M. Directed Algebraic Topology : Models of Non-Reversible Worlds, vol. 13 of *New Mathematical Monographs* (Cambridge University Press 2009). ISBN 9780521760362. 7, 93, 97, 105, 224, 225

Grätzer, G. General Lattice Theory (Birkhäuser 2003), 2nd ed. ISBN 978-3764369965. 194

Grätzer, G. Universal Algebra (Springer 2008), updated 2nd ed. ISBN 978-0-387-77486-2. 206, 209

Grätzer, G., Lakser, H., and Quackenbush, R. The structure of tensor products of semilattices with zero. Transactions of the American Mathematical Society 267(2):503–515 1981. doi:10.1090/S0002-9947-1981-0626486-8. 216, 217, 219

Grätzer, G. and Wehrung, F. Tensor products of semilattices with zero, revisited. Journal of Pure and Applied Algebra 147:273–301 2000. doi:10.1016/S0022-4049(98) 00145-5. 216

Grätzer, G. and Wehrung, F. A survey of tensor products and related constructions in two lectures. Algebra Universalis 45:117–134 2001. 216

Grillet, P.A. The Tensor Product of Commutative Semigroups. Transactions of the American Mathematical Society 138:281–293 1969a. doi:10.2307/1994915. 208

Grillet, P.A. The Tensor Product of Semigroups. Transactions of the American Mathematical Society 138:267–280 1969b. doi:10.2307/1994914. 208

Grillet, P.A. Abstract Algebra, vol. 242 of *Graduate Texts in Mathematics* (Springer 2007), 2nd ed. ISBN 978-0-387-71567-4. 194, 208, 209

Gromov, M. Metric Structures for Riemannian and Non-Riemannian Spaces. Modern Birkhäuser Classics (Birkhäuser 2007), reprint of 2001 ed. 150

Haefliger, A. Extension of Complexes of Groups. Annales de l'institut Fourrier 42(1-2):275–311 1992. 165

Halmos, P.R. Measure Theory. Graduate Texts in Mathematics (Springer 1974). ISBN 978-0387900889. 152

Hansen, P.B. The Origin of Concurrent Programming: From Semaphores to Remote Procedure Calls (Springer 2002). ISBN 978-0387954011. 6

Hashimoto, J. On Direct Product Decomposition of Partially Ordered Sets. Annals of Mathematics 54(2):315–318 1951. 204

Hatcher, A. Algebraic Topology (Cambridge University Press 2002). 7, 84

Haucourt, E. Categories of Components and Loop-free Categories. Theory and Applications of Categories 16(27):736–770 2006. ISSN 1201 - 561X. 7, 165, 166, 181

Haucourt, E. Streams, d-Spaces and their Fundamental Categories. Electronic Notes in Theoretical Computer Science 283:111–151 2012. doi:10.1016/j.entcs.2012.05.008. 7, 41, 73, 97, 98, 122, 123

Haucourt, E. Local pospaces, areas, and concurrency. Tech. rep., CEA Tech, List 2014. 7

Haucourt, E. and Ninin, N. The Boolean Algebra of Cubical Areas as a Tensor Product in the Category of Semilattices with Zero. In 7th Interaction and Concurrency Experience (ICE 2014), Electronic Proceedings in Theoretical Computer Science (2014). 132, 164, 206, 216, 218, 220, 221

Hazewinkel, M., Gubareni, N., and Kirichenko, V. Algebras, Rings and Modules vol. 2, vol. 2 of *Mathematics and Its Applications* (Springer 2007). 226

Hempel, J. 3-Manifolds, vol. 349 of *AMS Chelsea Publishing* (American Mathematical Society 1976). ISBN 9780691081830. 196

Higgins, P.J. Categories and Groupoids, vol. 32 of *Mathematical Studies* (Van Nostrand (Princeton) 1971). ISBN 442-03406-7. 7, 116

Hirschhorn, P.S. Model Categories and Their Localizations. Mathematical Surveys and Monographs (American Mathematical Society 2003). ISBN 978-0-8218-4917-0. 105

Hirschowitz, A., Hirschowitz, M., and Hirschowitz, T. Saturating directed spaces. Journal of Homotopy and Related Structures 16:25–32 2013. 99

Hoare, C.A.R. Communicating sequential processes. Communications of the ACM 21(8):666–677 1978. 16, 193

Hoare, C.A.R. Communicating sequential processes. Prentice Hall International Series in Computing Science (Prentice-Hall 1985). ISBN 978-0131532717. 193

Hofmann, K.H., Mislove, M.W., and Stralka, A. The Pontryagin Duality of Compact 0-Dimensional Semilattices and its Applications, vol. 396 of *Lecture Notes in Mathematics* (Springer 1974). ISBN 978-3-540-06807-5. 209, 216

Hovey, M. Model Categories, vol. 63 of *Mathematical Surveys and Monographs* (American Mathematical Society 1999). 7, 31, 36, 37, 39, 40, 105, 186, 224

Hungerford, T.W. Algebra. Graduate Texts in Mathematics (Springer 2003). ISBN 978-0387905181. 193, 194, 195

Husainov, A.A. Homology and Bisimulation of Asynchronous Transition Systems and Petri Nets. arXiv 2013. 225

Husemoller, D. Fibre Bundles. Graduate Texts in Mathematics (Springer 1993), 3rd ed. 227

Isaacson, S.B. Cubical Homotopy Theory and Monoidal Model Categories. Ph.D. thesis, Harvard University 2009. 44

Isaacson, S.B. Symmetric cubical sets. Journal of Pure and Applied Algebra 215:1146–1173 2011. 44

Jaco, W. Lectures on Three-Manifold Topology, vol. 43 of *CBMS Regional Conference* (American Mathematical Society 1980). ISBN 978-0-8218-1693-6. 196

Jardine, J.F. Cubical Homotopy Theory: a beginning. Tech. rep., Newton Institute 2002. 44

Jech, T.J. Set Theory. Springer Monographs in Mathematics (Springer 2002), 3rd ed. ISBN 3-540-44085-2. 70

Johnstone, P.T. Stone Spaces, vol. 3 of *Studies in Advanced Mathematics* (Cambridge University Press 1982). ISBN 978-0521337793. 101, 205, 209, 213

Joyal, A. and Tierney, M. Notes on simplicial homotopy 2008. 37

Kahl, T. Relative Directed Homotopy Theory of Partially Ordered Spaces. Journal of Homotopy and Related Structures 1(1):79–100 2006. 224

Kahl, T. A fibration category of local pospaces. Electronic Notes in Theoretical Computer Science 230:129–140 2009. 224

Kan, D.M. Abstract Homotopy I. Proceedings of the National Academy of Science U.S.A. 41(12):1092–1096 1955. 44, 45

Karrass, A., Magnus, W., and Solitar, D. Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations (Dover Publications 2004), 2nd ed. ISBN 0-486-63281-4. 48

Kelley, J.L. General Topology, vol. 27 of *Graduate Texts in Mathematics* (Springer 1955). 39, 101, 102, 118, 205

Kernighan, B.W. and Ritchie, D.M. The C programming language (Prentice-Hall 1988). 11

Klenke, A. Probability Theory: a comprehensive course. Universitext (Springer 2008). ISBN 978-1-84800-047. 152

Kneser, H. Geschlossene Flächen in dreidimensionalen Mannigfaltigkeiten. Jahresbericht der Deutschen Mathematiker-Vereinigung 38:248–259 1929. 196

Koppelberg, S. Handbook of Boolean Algebras I (North-Holland 1989). ISBN 978-0444702616. 134, 205

Krantz, S.G. and Parks, H.R. The Geometry of Domains in Space. Birkhäuser Advanced Texts (Birkhäuser 1999). ISBN 978-0817640972. 153

Krishnan, S. Directed Algebraic Topology and Concurrency. Ph.D. thesis, Chicago University 2006. 90

Krishnan, S. A Convenient Category of Locally Preordered Spaces. Applied Categorical Structures 17(5):445–466 2009. doi:10.1007/s10485-008-9140-9. 7, 90, 91, 92, 95, 105

Krishnan, S. Cubical Approximation for Directed Topology I. Applied Categorical Structures 23(2):177–214 2013. 43, 111, 159

Krivine, J. Algèbres de Processus Réversibles. These, Université Pierre et Marie Curie - Paris VI 2006. 118

Krivine, J. A Verification Technique for Reversible Process Algebra. In Fourth international workshop on reversible computation (RC 2012), vol. 7581, pp. 204–217 (Springer 2012). 118

Kuroš, A.G. Theory of Groups, vol. 2. AMS Chelsea Publishing (American Mathematical Society 1956). ISBN 978-0-8218-3477-0. 194

Lang, S. Fundamentals of Differential Geometry. Graduate Texts in Mathematics (Springer 1999). ISBN 0-387-98593-X. 102, 103, 227

Lang, S. Algebra. Graduate Texts in Mathematics (Springer 2002), 3rd ed. 193, 206, 208, 209, 212

Lawson, M.V. Finite Automata (Chapman & Hall/CRC 2004). ISBN 1-58488-255-7. 191

Lewis, G. The Stable Category and Generalized Thom Spectra. Ph.D. thesis, University of Chicago 1978. 31, 39

Lipski, W. and Papadimitriou, C.H. A Fast Algorithm for Testing for Safety and Detecting Deadlocks in Locked Transaction Systems. Journal of Algorithms pp. 211–226 1981. 159

Luttik, B. A Unique Decomposition Theorem for Ordered Monoids with Applications in Process Theory. In Mathematical Foundations of Computer Science 2003, vol. 2747, pp. 562–571 (Springer 2003). 7, 193

Luttik, B. and Oostrom, V. Decomposition orders: another generalisation of the fundamental theorem of arithmetic. Theoretical Computer Science 335(2-3):147–186 2005. 7, 193, 196

Mac Lane, S. Homology. Classics in Mathematics (Springer 1995), 4th ed. ISBN 978-3540586623. 225

Mac Lane, S. Categories for the Working Mathematician, vol. 5 of *Graduate Texts in Mathematics* (Springer 1998), 2nd ed. ISBN 0-387-98403-8. 32, 36, 41, 71, 108, 110, 121, 189

Mac Lane, S. and Birkhoff, G. Algebra. AMS Chelsea Publishing (American Mathematical Society 1999), 3rd ed. ISBN 978-0821816462. 209

Mac Lane, S. and Moerdijk, I. Sheaves in Geometry and Logic. Universitext (Springer 1994), 2nd ed. 33, 35, 38, 82

Maltsiniotis, G. La catégorie cubique avec connexions est une catégorie test stricte. Homology, Homotopy and Applications 11(2):309–326 2009. 44, 45

Mandel, L. and Pouzet, M. ReactiveML, a Reactive Extension to ML. In Proceedings of 7th ACM SIGPLAN International Symposium on Principles and Practice of Declarative Programming (PPDP'05) (Lisbon, Portugal 2005). 15

Massey, W.S. A Basic Course in Algebraic Topology. Graduate Texts in Mathematics (Springer 1991), 3rd ed. ISBN 978-3540974307. 196

May, J.P. Simplicial Objects in Algebraic Topology. Chicago Lectures in Mathematics (The University of Chicago Press 1967). 36, 37, 38

May, J.P. and Ponto, K. More Concise Algebraic Topology: Localization, completion, and model categories (Chicago University Press 2010). 105, 224

McLarty, C. Every Grothendieck topos has a one-way site. Theory and Applications of Categories 16(5):123–126 2006. 165

Milner, R. Communication and Concurrency (Prentice-Hall 1989). 193

Milner, R. Communicating Systems and The Pi-Calculus (Cambridge University Press 1999). 193

Milner, R. and Moller, F. Unique decomposition of processes. Theoretical Computer Science 107:357–363 1993. 7, 193

Milnor, J. The Geometric Realization of a Semi-Simplicial Complex. Annals of Mathematics 65(2):357–362 1957. 38

Milnor, J. A Unique Decomposition Theorem for 3-Manifolds. American Journal of Mathematics 84(1):1–7 1962. doi:10.2307/2372800. 196

Miné, A. Static analysis by abstract interpretation of concurrent programs. Habilitation thesis, École Normale Supérieure 2013. 10

Morgan, F. Geometric measure theory: A beginner's guide (Academic Press 2008). ISBN 978-0123744449. 152

Muchnick, S.S. Advanced Compiler Design and Implementation (Morgan Kaufmann 1997). ISBN 978-1558603202. 52

Nachbin, L. Sur les espaces topologiques ordonnés. Compte Rendus de l'Académie des Sciences de Paris 226 1948a. 68

Nachbin, L. Sur les espaces uniformes ordonnés. Compte Rendus de l'Académie des Sciences de Paris 226 1948b. 68

Nachbin, L. Sur les espaces uniformisables ordonnés. Compte Rendus de l'Académie des Sciences de Paris 226 1948c. 68

Nachbin, L. Topology and Order, vol. 4 of *Mathematical Studies* (Van Nostrand (Princeton) 1965). ISBN 978-0882753874. 7, 68, 69, 100

Nadler Jr., S.B. Continuum Theory, An Introduction, vol. 158 of *Monographs and Textbooks in Pure and Applied Mathematics* (Marcel Dekker 1992). ISBN 0-8247-8669-9. 69, 70, 134

Nakayama, T. and Hashimoto, J. On a problem of G. Birkoff. Proceeding of the American Mathematical Society 1:141–142 1950. 196

Naur, P. and Randell, B. (eds.). Software Engineering: Report of a conference sponsored by the NATO Science Committee (NATO 1969). 6

Nikiel, J., Tuncali, H.M., and Tymchatyn, E.D. Continuous Images of Arcs and Inverse Limit Methods, vol. 104 of *Memoirs of the American Mathematical Society* (American Mathematical Society 1993). 70

Ninin, N. Untitled. Ph.D. thesis, Université Paris 11 Orsay 2016. 201

Ore, Ø. On the foundation of abstract algebra. II. Annals of Mathematics 37(2):265–292 1936. 194, 204

Papadopoulos, A. Metric Spaces, Convexity, and Nonpositive Curvature. IRMA Lectures in Mathematics & Theoretical Physics (European Mathematical Society 2013), 2nd ed. ISBN 978-3037191323. 150

Pears, A.R. Dimension theory of general spaces (Cambridge University Press 1975). 222

Pedicchio, M.C., Tholen, W., et al. Categorical Foundations: Special Topics in Order, Topology, Algebra, and Sheaf Theory. Encyclopedia of Mathematics and its Applications (Cambridge University Press 2003). ISBN 9780521834148. 79, 82, 169, 209

Pierce, R.S. Tensor products of Boolean algebras. In Universal Algebra and Lattice Theory:Fourth International Confererence, Puebla, Mexico, 1982, vol. 1004 of *Lecture Notes in Mathematics*, pp. 232–239 (Springer 1983). ISBN 978-3-540-12329-3. 206

Plazar, Q. Untitled. Master's thesis, École Polytechnique 2015. 191

Plotkin, G.D. A structural approach to operational semantics. Journal of Logic and Algebraic Programming 60-61:17–139 2004. doi:10.1016/j.jlap.2004.03.002. Reprint of Tech. Rep. DAIMI FN-19, eponym, Computer Science Department, Aarhus University, Aarhus, Denmark. 1981. 22

Porter, T. Proper Homotopy Theory. In I.M. James (ed.), Handbook of Algebraic Topology, pp. 127–167 (North-Holland 1995). ISBN 978-0444817792. 123

Porter, T. Steps towards a 'directed homotopy hypothesis'. $(\infty, 1)$-categories, directed spaces and perhaps rewriting. presented at Homotopy in Concurrency and Rewriting conference, Palaiseau, France 2015. 105

Pratt, V. Modeling Concurrency with Partial Orders. International Journal of Parallel Programming 15(1):33–71 1986. 45

Pratt, V. Modeling Concurrency with Geometry. In D.S. Wise (ed.), Proc. 18th Ann. ACM Symposium on Principles of Programming Languages, pp. 311–322 (1991). doi:10.1145/99583.99625. 1, 6, 45, 106

Pratt, V. Higher dimensional automata revisited. Mathematical Structures in Computer Science 10(4):525–548 2000. doi:10.1017/S0960129500003169. 45

Preiss, D. Geometry of measures in $\mathbb{R}^n$: Distribution, rectifiablity, and densities. Annals of Mathematics 125(3):537–643 1987. 153

Quillen, D.G. Homotopical Algebra, vol. 43 of *Lecture Notes in Mathematics* (Springer 1967). 7, 37, 105, 166, 186, 224

Randell, B. Software Engineering in 1968. In 4th International Conference on Software Engineering, pp. 1–10 (IEEE Computer Society 1979). 6

Raußen, M. On the classification of dipaths in geometric models of concurrency. Mathematical Structures in Computer Science 10(4):427–457 2000. 110

Raußen, M. State Spaces and and Dipaths up to Dihomotopy. Homology, Homotopy and Applications 5(2):257–280 2003. 111

Raußen, M. Invariants of directed spaces. Applied Categorical Structures 15(4):355–386 2007. 119

Raußen, M. Reparametrizations with given stop data. Journal of Homotopy and Related Structures 4(1):1–5 2009a. 119

Raußen, M. Trace spaces in a pre-cubical complex. Topology and its Applications 156(9):1718–1728 2009b. 99, 119

Raußen, M. Simplicial models of trace spaces. Algebraic & Geometric Topology 10(3):1683–1714 2010. doi:10.2140/agt.2010.10.1683. 119

Raußen, M. Execution spaces for simple higher dimensional automata. Applicable Algebra in Engineering, Communication and Computing 23(1-2):59–84 2012a. 119

Raußen, M. Simplicial models for trace spaces II: General higher dimensional automata. Algebraic & Geometric Topology 12(3):1741–1761 2012b. doi:10.2140/agt.2012.12.1741. 119

Ravenel, D.C. Complex Cobordism and Stable Homotopy Groups of Spheres. AMS Chelsea Publishing (American Mathematical Society 2003), 2nd ed. ISBN 978-0-8218-2967-7a. 225

Raynal, M. Concurrent Programming: Algorithms, Principles, and Foundations: Algorithms, Principles, and Foundations (Springer 2013). ISBN 978-3-642-32026-2. doi:10.1007/978-3-642-32027-9. 6, 14, 15

Riehl, E. Categorical Homotopy Theory, vol. 24 of *New Mathematical Monographs* (Cambridge University Press 2014). ISBN 9781107048454. 36

Rockafellar, R.T. and Wets, R.J.B. Variational Analysis, vol. 317 of *Grundlehren der mathematischen Wissenschaften* (Springer 2004), corr. 2nd printing ed. 69

Roman, S. Advanced Linear Algebra, vol. 135 of *Graduate Texts in Mathematics* (Springer 2008a), 3rd ed. ISBN 978-0-387-72831-5. 208

Roman, S. Lattices and Ordered Sets (Springer 2008b). ISBN 978-0-387-78900-2. doi:10.1007/978-0-387-78901-9. 209

Rotman, J.J. An Introduction to the Theory of Groups. Graduate Texts in Mathematics (Springer 1994), 4th ed. ISBN 0-387-94285-8. 194

Schröder, B.S.W. Ordered Sets, An Introduction (Birkhauser 2002). 204

Serre, J.P. Homologie Singulière ds Espaces Fibrés. Ph.D. thesis, École Normale Supérieure 1951. 44

Sikorski, R. The Cartesian Product of Boolean Algeras. Fundamenta Matematicae 37(1):25–54 1950. 205

Sikorski, R. Boolean Algebras, vol. 25 of *Ergebnisse der Mathematik und ihrer Grenzgebiete* (Springer 1969), 3rd ed. ISBN 978-3-642-85822-2. 205

Starynkevitch, B. GCC MELT Official Website. `http://gcc-melt.org/` 2012. 4

Steen, L.A. and Seebach, J.A. Counterexamples in Topology. Dover Books on Mathematics (Dover Publications 1996). 124

Steenrod, N.E. A convenient category of topological spaces. Michigan Math. J. 14:133–152 1967. 39

Strickland, N.P. The Category of CGWH Spaces. Manuscript available electronically at `http://neil-strickland.staff.shef.ac.uk/courses/homotopy/cgwh.pdf` 2009. 39

tom Dieck, T. Algebraic Topology. EMS Textbooks in Mathematics (European Mathematical Society 2008). 196

van Glabbeek, R.J. Bisimulations for Higher Dimensional Automata. Manuscript available electronically at `http://theory.stanford.edu/~rvg/hda` 1991. 6, 45

van Glabbeek, R.J. Erratum to 'On the Expressiveness of Higher Dimensional Automata'. Theoretical Computer Science 368(1-2):168–194 2006. 6, 45

Wang, J. Handbook of Finite State Based Models and Applications. Discrete Mathematics and Its Applications (Chapman & Hall/CRC 2012). ISBN 978-1439846186. 191

Weibel, C.A. An Introduction to Homological Algebra. Cambridge Studies in Advanced Mathematics (Cambridge University Press 1994). 225

Weintraub, S.H. Factorization: Unique and Otherwise. CMS Treatises in Mathematics (A K Peters 2008). ISBN 978-1568812410. 195

Willard, S. General Topology (Addison-Wesley 1970). 101

Winskel, G. Handbook of Logic in Computer Science, vol. 4: Semantic Modelling, chap. 1: Models for concurrency, pp. 70–102 (Oxford University Press 1995). 6, 46

Worytkiewicz, K. Sheaves of ordered spaces and interval theories. Journal of Homotopy and Related Structures 5(1):37–61 2010. 105

Index

List of Figures