ALGEBRAIC TOPOLOGY AND CONCURRENCY

LISBETH FAJSTRUP, ERIC GOUBAULT AND MARTIN RAUSSEN

1. Introduction

This article is intended to provide some new insights about concurrency theory using ideas from geometry, and more specifically from algebraic topology. The aim of the paper is two-fold: we justify applications of geometrical methods in concurrency through some chosen examples and we give the mathematical foundations needed to understand the geometric phenomenon that we identify. In particular we show that the usual notion of homotopy has to be refined to take into account some partial ordering describing the way time goes. This gives rise to some new interesting mathematical problems as well as give some common grounds to computer-scientific problems that have not been precisely related otherwise in the past.

The organization of the paper is as follows. In Section 2 we explain to which extent we can use some geometrical ideas in computer science: we list a few of the potential or well known areas of application and try to exemplify some of the properties of concurrent (and distributed) systems we are interested in. We first explain the interest of using some geometric ideas for semantical reasons. Then we take the example of concurrent databases with the problem of finding deadlocks and with some aspects of serializability theory. More general questions about schedules can be asked as well and related to some geometric considerations, even for scheduling micro-instructions (and not only coarse-grained transactions as for databases). The final example is the one of fault-tolerant protocols for distributed systems, where subtle scheduling properties go into play.

In Section 3 we give the first few definitions needed for modeling the topological spaces arising from Section 2. Basically, we need to define a topological space containing all traces of executions of the concurrent systems we want to characterize plus the information about how time flows. This is the main difference with standard topological reasoning in which there is no information about relation "in time" among points. The central notion here is that of a local po-space, which is a topological space with a local partial-order of time on it. Some examples are given, but we will only see in Section 6 that cubical complexes (or Higher-Dimensional Automata, [27] and [49]) give rise naturally to such spaces, hence most "combinatorial" concurrency models are instances of these local po-spaces. It is worth noting that some models in General Relativity [48] consider timed spaces, and the authors benefited from some of these physical concepts when developing this theory.

Section 4 then gives the first definitions of the new homotopy theory we need in order to define equivalence of paths along the intuitions developped in Section 2. A central notion here is that of homotopy history components, which contains the relevant information for computer-scientific applications, as well as for classification of local po-spaces modulo "directed" homotopy. Some examples are given that show that this directed homotopy is finer than the usual homotopy theory in the sense that it can distinguish homotopy equivalent (in the standard sense) topological spaces.

We then study a particular subcategory of local po-spaces, those which are locally euclidean, i.e. the local partial order is that of $\mathbb{R}^n$ (for some n). A central statement is that we can take "still pictures" of the dynamics on such spaces, i.e. look at cuts which contain points not related through time, and this can give obstructions to deformation in the directed sense.

We carry on by looking at cubical complexes (or HDA) and show they are in some sense a combinatorial counterpart of these local po-spaces (at least of some large subcategory). We refer the reader to [27] or to the more recent [19] for actual semantics of some concurrent systems using these cubical complexes. Some of the "combinatorial" deformation theory in cubical complexes is developped and related to the directed homotopy in the continuous case, using in particular the notion of subdivision, in Section 7.

A major application is fully treated in Section 8, the one of concurrent databases presented informally in Section 2. It is an application of the preceding theory and a refinement and extension of the result in [35].

Then some mathematical directions are given in Section 9, some related computer-scientific perspectives are listed in Section 10, and finally we refer the reader to some related papers where algebraic topology is at the center of computer-scientific modeling and proofs, in Section 11.

Part of this was presented by two of the co-authors at the fourteenth conference on the Mathematical Foundations of Programming Semantics (London, may 1998).

2. Motivation and Examples of applications

2.1. **Semantics and static analyses.** Without the ambition to be complete, we can trace back the use of geometrical models and properties to the beginning of theoretical computer science, in the use of graph theory, or of partial orders to describe the semantics of systems.

For instance, sequential machines can be studied by examining their operational behaviours – that is by looking at their state transition graphs. One of the fundamental properties that we might want to study is confluence of the performed computation. This is obviously a property of a highly geometric nature: we must be able to complete all non-deterministic applications of conflicting reductions by some other reductions that all converge to the same result; i.e. we must have diamond shapes in the state transition graphs describing the sequences of operations of our sequential machines.

For concurrent machines, the geometric properties of computation include those of sequential machines but are even more intricate. Purely (interference free) asynchronous executions of two processes are confluent and therefore recognizable geometrically as diamonds (or squares). For example, the operational semantics of the interference free parallel composition of two actions a and b is,

$$\begin{tikzcd}
 & s_3 & \\
s_1 \arrow[ur, "b"] & & s_2 \arrow[ul, "a"'] \\
 & s_0 \arrow[ul, "a"] \arrow[ur, "b"'] &
\end{tikzcd}$$

The first problem identified [63] of this semantic description is that it is not stable under refinement of actions. Refinement is a property that is interesting when it comes to automatically verifying programs. It means that in the case of checking a property for a given program, we would like to be able to check it on a view of the program that looks directly at some sequences of actions and not at each "atomic" action composing it. As an example of non-stability here, the parallel composition of A (abstracting the sequential composition of a with b denoted by $a.b$) and of action c is shown in Figure 1 whereas the parallel composition of $a.b$ with c should be as shown in Figure 2. We see that there are less paths in Figure 1 than in Figure 2. So we might lose information in that refinement process. This practically compels many static analyses of parallel programs, based on a transition system's semantics, to be of an exponential complexity in the number of atomic transitions.

A second problem is a purely semantical one. In some cases, we would like to be able to specify the actual use of shared ressources of a parallel program, like, how many processors are busy or idle, or should a process wait for a shared variable? As you can see in the diagram above, the parallel execution of a with b is identified with the non-deterministic choice between $a.b$ and $b.a$, called interleaving of a and b. These two should denote entirely different behaviours in fact. The former should indicate that actions a and b can overlap in time, whereas the latter should prescribe that a and b are conflicting operations and that one has to be executed before the other. This is central to the discussion of mutual exclusion properties for instance.

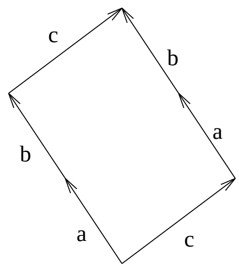

FIGURE 1. Interleaving of A and c.

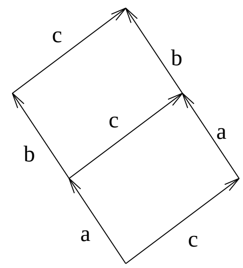

FIGURE 2. Interleaving of $a.b$ and c.

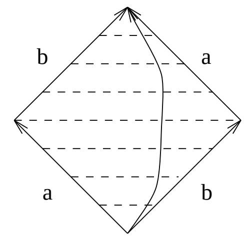

FIGURE 3. New traces

A first solution has been proposed in slightly different forms, asynchronous transition systems [4], concurrent automata [59], transition systems with independence [64] etc. These solutions very often consist in adding an independence relation between atomic actions involved in an ordinary transition system. In these semantics, the interleaving of two independent actions means their execution in parallel, whereas the interleaving of two non independent actions means their execution in mutual exclusion. Unfortunately, in these models, it is difficult to speak in a natural manner of more complex mutual exclusion properties, like shared ressources that one can access in parallel n times but not $n+1$ ($n \geq 2$), nor of the number of busy processes at some instant in a distributed system. For instance, given three actions a, b and c, should we understand a and b independent, b and c independent and c and a independent as the same as a, b and c are independent? This probably is not true if you are considering a, b and c as the requests to print a (different) file on a printer addressed to a server of printers. If the server controls two printers then, on the program side, all pairs of actions are independent, whereas three requests cannot be treated at the same time. If the server controls three printers, all three requests are independent. These points are actually crucial in a certain number of applications. In particular, concerning the proof of parallel programs on constrained architectures, or the proof of fault-tolerant distributed protocols in which the number of busy processors is of primary importance (see Section 10 and 11), or for optimization of the use of shared ressources.

Most of these aspects are dealt with traditionnally by resorting to Petri nets. But, even if the operational meaning of Petri nets is simple, it is not of the same nature as for transition systems. For instance, Petri nets are difficult to use in a compositional way, which is not the case of transition systems [1].

Let us take a closer look at the geometry of transition systems used in concurrency now. We only have to think of a concurrent execution of two actions a and b on two processors P_1 and P_2 as a curve in $\mathbb{R}^2$ whose points have abscissa (respectively ordinate) the local time of P_1 taken to execute a (respectively the local time of P_2 taken to execute b). This gives new traces, other than just the interleavings, as in Figure 3, which are all increasing paths in the two coordinates (because we cannot invert the time flow) included in the square delineated by the interleavings of a and b.

This was first proposed in [49] and [62], and further treated in [27]. Now we can understand the problem of pools of printers explained briefly above as follows. If we have three printers in the pool then we allow traces (i.e. paths) that are inside the cube delimited by the three actions a, b and c whereas if we have only two printers, we do not allow them, but only those which are on the boundary of the cube (which is a closed surface). Here we are confronted to two presentations of essentially the same phenomenon. The first one is the geometry of continuous paths (like one could study in mechanics). The second one is a discretization of it, which in general comes first in the semantics applications (but not in others see Section 10); basically, abstract all paths inside a n dimensional cube by the interior of the cube itself, then describe the geometry of executions as the amalgamation (or pasting) of all the different k dimensional cubes entering into play, as shown in Figure 2.1. This is precisely what is called a cubical complex in combinatorial algebraic topology (see Section 6) or Higher-Dimensional Automata [49] in computer science. But

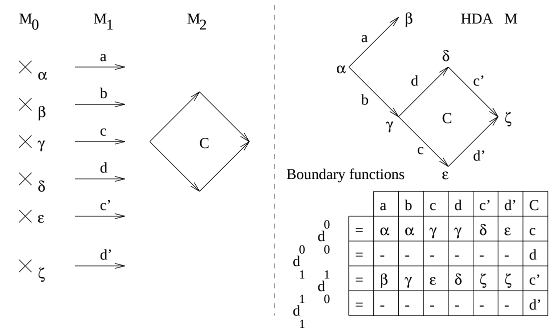

FIGURE 4. The glueing of elementary cubes.

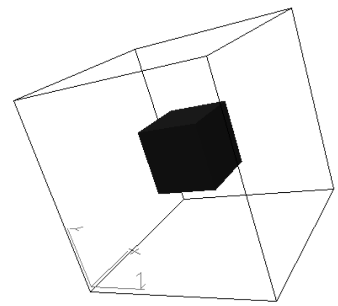

FIGURE 5. A 2-semaphore.

we are considering cubical complexes with an orientation given by the time flow (by orienting the segments constituting it), whereas in ordinary algebraic topology we do not orient shapes in such a manner. So the continuous counterpart of such discretizations is more than a topological space, it contains also order relations. This is developped in Section 3 under the name of po-spaces (and local po-spaces) whose formalisation and understanding is the main objective of this article.

Now, do we gain something by using such geometric concepts? Do we use in particular mathematical theorems and techniques to derive a new knowledge on concurrent and distributed systems that way? We will try to argue that the answer is yes indeed.

2.2. **Distributed databases.** Let us first of all take a simple example, first given in [35]. Consider a distributed database in which transactions T_1 to T_n access shared variables a, b,$\cdots$ using locks: Pa to lock the exclusive access to a and Va to unlock a so that other transactions can use a. In order to simplify the presentation, we can consider that the language of transactions is given by the actions P and V (on any variable), and that we do not take care of the actual values of the variables nor of the numerical calculations made. In this abstraction, Pa represents the request for a lock on a and Va the action of unlocking it.

The semantics of this language, using cubical complexes, is easy to describe (see [19] for a complete treatment). All executions of a P action can be made in an asynchronous manner – the lock requests, even on the same object, are independent – but the calculations on the same object are serialized given that one processor, or the other has obtained the lock on the object. This is what is shown in Figure 7 where two transactions try to access the same object a for a calculation. A more intricate example is shown in Figure 6. The continuous counterpart of these examples are process graphs in the sense of E. W. Dijkstra [16].

Now, it is easy to see (at least on these examples) that every continuous deformation of an execution path (i.e. a path going from the bottom left, ending at the top right, and which is increasing in each coordinate) does not change the history of accesses to shared variables, hence cannot change the final values (at the end of execution) of the shared variables. This implies that we can try to characterize traces up to that sort of "homotopy" when we want to determine the possible outcomes of a concurrent program: in Figure 7, all executions going above and left of the hole have the property that T_2 modifies a before T_1. In the sequel, a scheduler will be any sequence of actions modulo this homotopy (which will be formally defined in Section 4). This is very much akin to what is done in Mazurkiewicz trace theory, homotopy does correspond (at least for paths) to partial commutation rules here. But the theory that we develop in this paper is more general, in particular when it comes to higher dimensions (there is no counterpart in trace theory).

Two classical problems in concurrent databases' theory are (see Section 4 and 8 for a partial treatment),

- Can transactions go into a deadlock? (then how can we properly schedule them?)
- Is the transactions systems serializable?

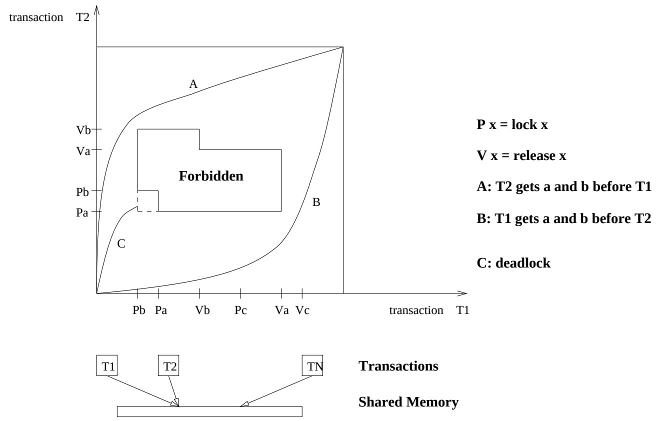

FIGURE 6. Cubical complex arising from $Pb.Pa.Vb.Pc.Va.Vc \mid Pa.Pb.Va.Vb$.

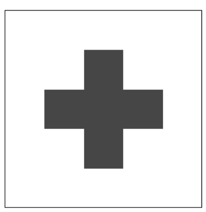

FIGURE 7. $Pa.Va.Pc.Vc \mid Pa.Va.Pb.Vb$.

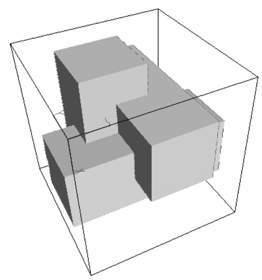

FIGURE 8. 3 dining philosophers.

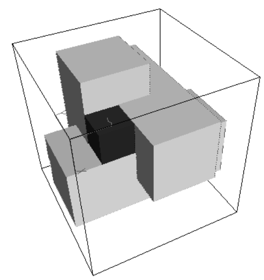

FIGURE 9. Unsafe part.

Deadlocks, or more generally unsafe areas containing all points which will eventually lead to a deadlock have a geometric characterization, see [19] (look also at Figure 8 and 9 for a picture of the three dining philosophers' problem). We will see later on in Section 4 and 8 that these correspond to certain "diconnected components" characterized by homotopy theoretic properties.

Let us now have a look at the serialization problem. Consider the following two transactions `R` and `S` put in parallel.

```
R: P A; P B;              S: P B;
   A:=B+1;                   B:=B+2;
   V A; V B;                 V B;
   P B;                      P A; P B;
   B:=3;                     A:=2*B;
   V B;                      P A; P B;
```

When beginning with the initial values `A=0`, `B=0` we can get the following values,

R: A=1	R: A=1	R: A=1	S: B=2	S: B=2	S: B=2
R: B=3	S: B=2	S: B=2	R: A=3	R: A=3	S: A=6
S: B=5-	R: B=3-	S: A=4-	S: A=4-	R: B=3-	R: A=3-
S: A=10-	S: A=6-	R: B=3-	R: B=3-	S: A=6-	R: B=3-

Only the first trace (with result `A=10, B=5`) and the last trace (with result `A=3, B=3`) are correct. The other traces are interferences: as a matter of fact we want that all the execution traces give the same result as a sequential trace, i.e., `R` then `S` or `S` then `R` in their totality. It is the property called *serializability*.

instructions/cycle	0	1	2	3	4	5
add.s	U	S+A	A+R	R+S	$\emptyset$	$\emptyset$
add.s		U	S+A	A+R	R+S	$\emptyset$

where U is unpack, S is shift, A is adder and R is round.

FIGURE 10. MIPS R4000 floating point unit.

A solution is given by the 2-phase locking protocol: all processes P accessing to a database should first do all the lock operations then the computation then all the unlock operations. The same operations programmed using this protocol are the following,

```
R:                      S:
   P A;                    P A;
   P B;                    P B;
   A:=B+1;                 B:=2;
   B:=3;                   A:=2*B;
   V B;                    V B;
   V A;                    V A;
```

We will see in Section 8 a geometric proof of the serializability of the 2-phase protocol.

2.3. **Scheduling algorithms.** Let us take an example from [36] and [50]. Many modern CPUs like SPARCs or MIPS pipeline instructions. Of course, their functional units, registers or bus are all used in mutual exclusion. Unfortunately the pipelined instructions overlap in time as they use more than one clock cycle and some of them cannot be executed (otherwise "**structural hazards**" occur) within a certain number of cycles after some others (see figure 2.3). We do not want to use the pipeline in mutual exclusion since we would have to empty it after every instruction. The problem addressed in [50] is to verify that schedulers for a single process ensure that structural hazards will not occur (safety). In a concurrent framework, if there are more processes than processors, we can address the new problem of finding a way to interleave actions from different processes executed on the same processor, that verify the constraints while using the pipeline at the best of its capabilities (this is a view formalised in [3] see example 2.1). Some processors (like INTEL's Pentium) are even more complex to deal with since some resources may be used by at most two processes in parallel but not three[1].

Example 2.1. We see from figure[2] 2.3 that Suppose that we want to execute two instructions add.s one after the other on the MIPS R4000 floating point unit Then at cycle 2 the adder A has to be used by both instructions (coming from the same thread). The same holds at cycle 3 for the round unit R. We say in that case that there is a hazard on A at cycle 2 and a hazard on R at cycle 3. A good scheduler should have prevented us from this situation by interleaving the two threads after the first add.s and continue with non-conflicting instructions of the second thread for the pipeline to be emptied a bit before executing the second add.s.

Once again, if we translate this problem by using P and V operations on the shared resources, we see that our problem is to find schedules that do not deadlock.

2.4. **Fault-tolerant distributed protocols.** Let us take another simple example. Let two processors P and P' communicate by writing and reading variables in a shared memory. For instance, processor P (respectively P') can write, by action `scan`, a local variable u (respectively u') in a variable of the shared memory x (respectively x'), and can also copy x and x' in its local memory, by a `update` action, in u and v (respectively in u' and v'). We can also make some choices using a `case` statement. A scheme of the underlying concurrent machine is pictured in Figure 11.

The processors can also do some calculations in a purely local manner. Such a machine has the property that every computation is done in a "wait-free" manner, i.e. there is no synchronisation between the processors. This implies that this machine is fault-tolerant to the extent that if one

[1]it has two integer arithmetic units.
[2]taken from [50].

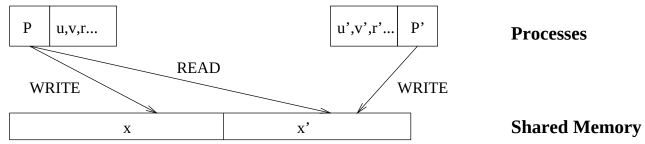

FIGURE 11. A simple shared memory concurrent machine.

of the two processors is faulty and stops computing, the other can carry on its own computation, with the partial data it possesses.

An important application case is, once more, a distributed database. Suppose that transactions, in some remote booths, want to modify the same variable, but with two distinct values. This can be the case of two customers willing to book the same plane. Can we find a practical protocol that, given the architecture of the distributed database, ensures that only one of the two will have its ticket booked, whereas the other will be notified of the failure of its transaction? This is what is called the *consensus* problem: we want to make two processors agree on a common value.

This question is a particular case of a more general one which is to know what this concurrent machine can compute. In the case of two processors, this was solved in [22]: it is by no means possible to solve the consensus problem on our simple machine. But we had to wait until quite recently for a characterization of what can be computed on our asynchronous machine with n processors (for any n). This has been done by methods borrowed from combinatorial algebraic topology (simplicial complexes). This in fact is again a directed homotopy problem, as we explain below.

Let us take an example of a program $Prog$ having the two following processes in parallel,

$$\begin{array}{rcl} P & = & update; \\ & & scan; \\ & & case\ (u,v)\ of \\ & & \ (x,y'):\ u=x'; update; \\ & & \ default:\ update \end{array} \qquad \begin{array}{rcl} P' & = & update; \\ & & scan; \\ & & case\ (u,v)\ of \\ & & \ (x,y'):\ v=y; update; \\ & & \ default:\ update \end{array}$$

We have mainly the following three schedules since the only possible interactions are between the *scan* and *update* statements,

(i) Suppose the *scan* operation of P is completed before the *update* operation of P' is started: P does not know y so it chooses to write x. $Prog$ ends up with $((P,x),(P',y))$.
(ii) Symmetric case: $Prog$ ends up with $((P,x'),(P',y'))$.
(iii) The *scan* operation of P is after the *update* of P' and the *scan* of P' is after the *update* of P. $Prog$ ends up with $((P,x'),(P',y))$.

Now, we see that each of these three schedules correspond to the reordering of scan and update operations.

If we represent commutation of two transitions by filling their interleaving by a 2-transition, and represent non-commutation by not filling the interleaving with 2-transitions, we come up with the three paths modulo homotopy of the left part of Figure 14. This amounts to identifying the control flow of our asynchronous language with a semaphore program, for which the pair $(scan, update)$ of actions is identified with an exchange of information, by P/V synchronisation. This is a good analogue up to the extent we are only interested by the effect of the history of the communications on the environment, look at Figure 13.

We then have mainly two configurations[3] of "holes" on a square, when we look at the homotopy classes of directed paths (1-schedules), as shown in Figure 14.

In the first configuration, there are three schedules (when the holes are "incomparable"), whereas in the second configuration, there are four schedules (when the holes are "comparable"). Therefore, if you look at some more complex configuration between many holes, as in Figure 15, its set of schedules is described by a complex tree-like picture (also in Figure 15).

[3] There can be no overlapping of holes as in the case of P/V programs

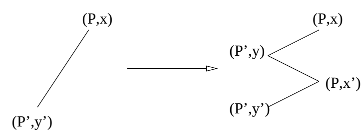

FIGURE 12. Subdivision of a segment into three segments.

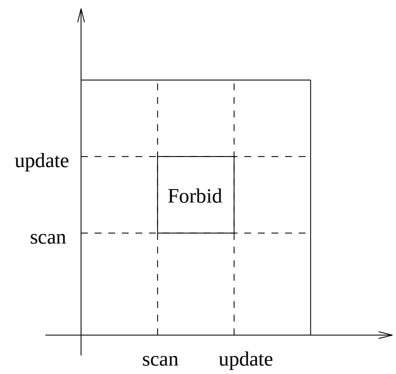

FIGURE 13. A P/V analogue to *scan/update*

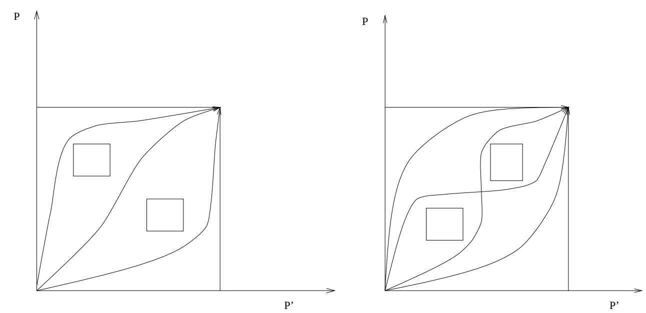

FIGURE 14. The two possible relative configurations of holes

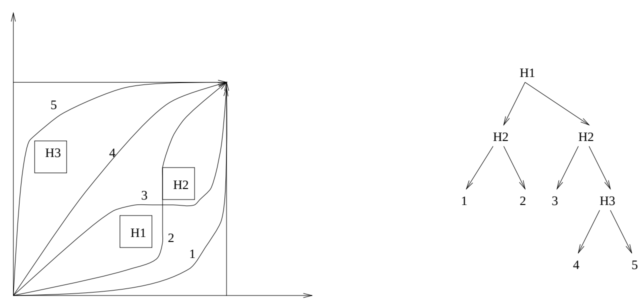

FIGURE 15. A more complex situation

As a matter of fact, we could easily describe a superset of the schedules.

Formally, two holes are comparable if there is a directed path from the end of one of the holes to the start of the other hole. Comparability is a partial order, and we can show that any linearisation of this partial order, by deforming the shape to have the holes in a linear configuration) gives a superset of the possible schedules. But a chain (under this order, compatible with the comparability partial order) of holes has exactly the "directed" homotopy type of a binary tree (Figure 16).

To any leaf of the tree we can associate a segment in the output graph as follows,

- a vertex is any local state, i.e. (P, x) or (P', x'),
- a segment between (P, x) and (P', x') represents the global state of the two processes,

A leaf of the tree corresponds to one of the possible schedules of execution, and can only lead to a unique global state, hence a segment.

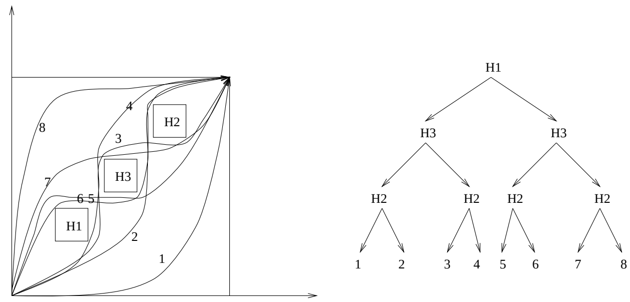

FIGURE 16. The homotopy type of a chain of holes

For instance the example program *Prog* does map a segment (a global state) to three segments as shown in Figure 12. This is a particular case of a more general phenomenon. In the semantics of the *scan/update* language, going from a leaf of the binary tree from the next one, we always share one vertex (a P vertex or a P' vertex), so the graph that is reached by the possible schedules is connected. We have made this sketch of proof for a superset of possible 1-schedules. This does not change the connectivity argument, which completes the proof of the following result; let $\{e_1, \ldots, e_k\}$ be the set of global states that one can find at the end of an execution of a *scan/update* program starting from of a global state $e = ((P, u), (P', v'))$. Then the graph generated (as mentionned above) by these global states is a connected graph (actually there is a converse to that, which is by iterating in some manner the example program above [29]).

This shows in particular that the consensus problem cannot be solved in our *scan/update* programming language: starting from a global state $((P, 0), (P', 1))$ the possible answers we wish to get i.e. $\{((P, 0), (P', 0)), ((P, 1), (P', 1))\}$ do not form a connected graph.

Many other results can be derived in that style, and one of the objectives of the theory presented in this paper is to be able to derive sufficient knowledge about "schedules" for complex programming languages so that we can understand what these languages can compute[4].

In order to give further and more intricate examples in the sequel, we slightly enhance our toy programming languages so that,

- shared objects can be "weakly-synchronizing", i.e., they can be shared by k processes but not $k+1$ at the same time, for any $k \geq 1$. Examples of such objects can be redundant functional units (for instance, in microprocessors, or in workshop modelisations), or communication buffers of fixed size (in the case of asynchronous message passing), or shared FIFO queues (in shared-memory systems). We choose to think of these objects `s` in the convenient form of "k-places buffers", on which we can do actions `push(x,s)` where `x` is any integer value and `read(y,s)` where `y` is any local (to the process executing the instruction) integer array variable.
- `read(y,s)` gives an atomic snapshot of the shared buffer in the local memory. Then any array operation like access at the `i`th element, `y[i]` can be performed locally.
- `push(x,s)` corresponds to asking to take one of the free places of the buffer (in FIFO order here): if the buffer is full then it pushes the values so that the first value entered is discarded. If two or more `push` instructions are executed right at the same time the semantics is not defined (anything can happen, in practice at the hardware level if no locks are used, this corresponds to some kind of short-circuit). In order to protect the integrity of the messages, we are using instructions `Ps` and `Vs` to acquire (respectively relinquish) one of the locks on the buffer. One place buffers are just the same as ordinary integer variables

[4]It must be noted that most of the approaches about this kind of problems make simplifying assumptions about the model of computation (making them more synchronous than they should) to be able to enumerate the schedules in a combinatorial manner.

3. Definitions

We start with elementary definitions and properties of pospaces that are the starting point for formalizing the intuitions of Section 2 (see [25]):

Definition 3.1.
1. A *partial order* $\leq$ on a set U is a reflexive, transitive and antisymmetric relation. We write $x < y$ for ($x \leq y$ and $x \neq y$).
2. A partial order $\leq$ on a topological space X is called *closed* if $\leq$ is a closed subset of $X \times X$ in the product topology. In that case, $(X, \leq)$ is called a *pospace*.

The idea is that a po-space is a topological space in which points are ordered globally through time. This is sometimes too strong an assumption and will lead us to local po-spaces (Definition 3.3).

Remark 3.2. Let $(X, \leq)$ denote a pospace.
1. For every $x \in X$, the sets $\downarrow x = \{y \in X | y \leq x\}$ and $\uparrow x = \{y \in X | y \geq x\}$ are closed.
2. For every pair of points $y_1, y_2 \in X$, the set $[y_1, y_2] = \{x \in X | y_1 \leq x \leq y_2\} = \downarrow y_2 \cap \uparrow y_1$ is closed.
3. A partially ordered topological space is a pospace if and only if whenever $a \not\leq b$, there exist open sets U and V with $a \in U$ and $b \in V$ such that for all $x \in U$ and $y \in V$ $x \not\leq y$. Hence a pospace is Hausdorff.[25].

Definition 3.3. Let X be a topological space.
1. A *local partial order* on X is a covering $\mathcal{U}$ of X by open partially ordered sets $(U, \leq_U)$ such that for all $(U, \leq_U)$ and $(V, \leq_V)$ in $\mathcal{U}$ we have
$$\forall x, y \in U \cap V : \quad x \leq_U y \Leftrightarrow x \leq_V y.$$
2. A refinement of the local partial order $(\mathcal{U}, \{\leq_U\}_{U \in \mathcal{U}})$ on X consists of a refinement $\mathcal{V}$ of the covering $\mathcal{U}$ with partial orders such that for every $V \subset U, V \in \mathcal{V}, U \in \mathcal{U}$ and $x, y \in V$
$$x \leq_V y \Leftrightarrow x \leq_U y.$$
3. Two local partial orders on X are said to be equivalent if they have a common refinement.
4. A topological space X with a local partial order $(\mathcal{U}, \{\leq_U\}_{U \in \mathcal{U}})$ is called a *local pospace* if there is a refinement $\mathcal{V}$ of the local partial order such that for every $V \in \mathcal{V}$, $(V, \leq_V)$ is a pospace.

What we gain here is the ability to consider loops and points which you can come across in a trace of execution (infinitely) many times.

Remark 3.4.
1. This is a sort of *germ* or *sheaf* type definition of a local partial order and in particular of the monotone functions (below).
2. The transitive hull of the partial orders given on subsets does not in general give rise to an interesting relation on X. If X is the circle (3.5) with local partial order given by a chosen direction, then the hull of the relation is the trivial relation $x \leq y$ for any pair x and y. The same is true for the torus.
3. By an abuse of notation, we will henceforth denote a locally partially ordered space $(X, \mathcal{U})$ without the $\leq_U$.
4. The equivalence of local partial orders is an equivalence relation. To prove transitivity, suppose $\mathcal{U}$ and $\mathcal{V}$ have a common refinement $\mathcal{W}_1$ and that $\mathcal{W}_2$ is a common refinement of $\mathcal{V}$ and $\mathcal{T}$. Then $\mathcal{W}_1 \cap \mathcal{W}_2 = \{W_1 \cap W_2 | W_i \in \mathcal{W}_i, i = 1, 2\}$ is a common refinement of $\mathcal{U}$ and $\mathcal{T}$. The partial orders are defined as follows: For all $W_1 \cap W_2$ there are $U \in \mathcal{U}$ and $V \in \mathcal{V}$ such that $W_1 \in U \cap V$ and there are $V' \in \mathcal{V}$ and $T \in \mathcal{T}$ such that $W_2 \in V' \cap T$. For $x, y \in W_1 \cap W_2$ we have $x \leq_U y \Leftrightarrow x \leq_V y \Leftrightarrow x \leq_{V'} y \Leftrightarrow x \leq_T y$.

Example 3.5.
1. The circle $S^1 = \{e^{i\theta} \in \mathbb{C}\}$ has a local partial order: $U_1 = \{e^{i\theta} \in S^1 | 0 < \theta < 2\pi\}$ has a (partial) order given by the order of the θ and $U_2 = \{e^{i\theta} \in S^1 | \pi < \theta < 3\pi\}$ is (partially) ordered by the order on the θ's.

2. The torus T^2 is $\mathbb{C}$ modulo a lattice $z \equiv z + ip + q \ \forall (p,q) \in \mathbb{Z} \times \mathbb{Z}$ and hence it inherits a local partial order from the standard partial order on $\mathbb{C} \cong \mathbb{R}^2$. This is equivalent to choosing a partial order on each of the two generators of the torus.
3. Let X be a disjoint union of four copies of the unit square I^2. We get inequivalent global partial orders on X by considering $X = [0,1]\times[0,1] \bigcup [0,1]\times[4,5] \bigcup [4,5]\times[0,1] \bigcup [4,5]\times[4,5]$ or $X = [0,1]\times[0,1] \bigcup [0,1]\times[4,5] \bigcup [4,5]\times[0,1] \bigcup [2,3]\times[2,3]$ both with the partial order induced from $\mathbb{R}^2$. Considered as local partial orders, these are equivalent. A common refinement is defined by letting all 4 copies of I^2 have the partial order induced from $\mathbb{R}^2$ and no further relations.

Definition 3.6. 1. Let $(X, \mathcal{U})$ and $(Y, \mathcal{V})$ be *locally partially ordered spaces*. A continuous map $f : X \to Y$ is called a *dimap* (directed map) if there are refinements $\mathcal{U}'$ of $\mathcal{U}$ and $\mathcal{V}'$ of $\mathcal{V}$ such that

$$\forall U \in \mathcal{U}', V \in \mathcal{V}', x, y \in U \cap f^{-1}(V) : \quad x \leq_U y \Rightarrow f(x) \leq_V f(y).$$

2. A *dipath* in X is a dimap $f : I \to X$ from the unit interval I with the natural (global) order $\leq$.

What we get here is the mathematical definition of traces of executions as we showed in Figures 3, 6, 14, 15 and 16.

Remark 3.7. 1. Let $f_1, f_2 : I \to X$ denote two dipaths with $f_1(1) = f_2(0)$. Their *concatenation* $f_1 * f_2$ is again a dipath. ($(f_1 * f_2)(t) = \begin{cases} f_1(t) & t \leq 0.5; \\ f_2(2t-1) & t \geq 0.5. \end{cases}$)
2. One might look at maps from *arbitrary* intervals and allow equivalence classes with respect to strictly increasing homeomorphisms between intervals.

Definition 3.8. (Compare [48].) Let X be a locally partially ordered space. We define a new relation $\prec$ on X by $x \prec y$ if there is a dipath from x to y in X.

Lemma 3.9. *If X has a global partial order, $\leq$ the relation $\prec$ is a new partial order.*

Proof. The relation $\prec$ is coarser then the relation $\leq$, i.e., $x \prec y \Rightarrow x \leq y$. Hence, $\prec$ is antisymmetric. Concatenation of dipaths shows the transitivity of $\prec$. $\square$ This is some kind of "reachability" relation which underlines most proofs in semantics.

Remark 3.10. If X is locally partially ordered, the relation $\prec$ is still transitive, but it is not necessarily antisymmetric as one can see from the example 3.5 with the oriented circle.

Definition 3.11. Let $y \in X, S \subset X$.

1. The set $J^+(y) := \{x \in X | \quad y \prec x\}$ is called the *future* of y; likewise, one defines the *past* $J^-(y)$. The set $J(y) := J^-(y) \cup J^+(y)$ is called the *history*.
2. $J^+(S) := \bigcup_{x\in S} J^+(x)$, $J^-(S) := \bigcup_{x \in S} J^-(x)$, $J(S) := J^-(S) \cup J^+(S)$ are called the *future, past, history* of S.
3. x is called an *initial* point if $J^-(x) = \{x\}$; x is called a *final* point if $J^+(x) = \{x\}$.

Remark 3.12. Initial points, resp. final points are local maxima, resp. minima with respect to the partial order $\prec$. An initial point is *unreachable* from any other initial point. A final point is unreachable from any other final point. Hence a *deadlock* is a final point, which is not among the final points representing succesful outcomes of the computations.

4. Diconnected spaces, diconnected components, di-1-connected spaces

Now we want to formalize the deformations of paths we have been "using" in the introduction, i.e. directed homotopy, and characterize the local po-spaces up to dihomotopy. First, we should have a new notion of connectivity and of connected components since we have to take the flow of time into account.

To define *di-connected components* we have to study the dipaths up to deformation. This seems very different from the non-directed topology case. It is however what one would expect, since we

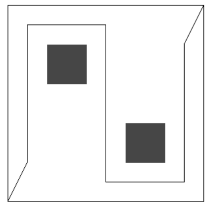

FIGURE 17. "A path around two non-ordered holes".

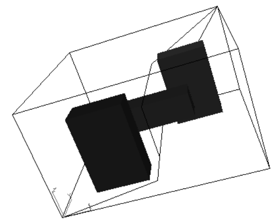

FIGURE 18. "Room with three barriers" and 2 non-dihomotopic dipaths.

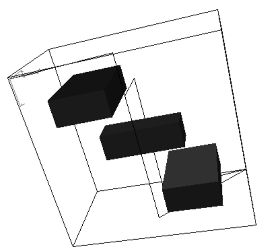

FIGURE 19. "Room with three barriers" (another view).

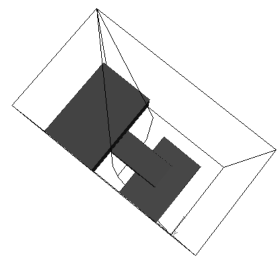

FIGURE 20. "Room with three barriers" (yet another view).

have to take the interplay between the partial order and the topology into consideration, and this is of course reflected in the dipaths. Throughout this section, $(X, \leq)$ is supposed to be a locally partially ordered topological space.

Definition 4.1. A dipath $\alpha : I \to X$ is called *inextendible*, if there is no dipath $\beta : J \to X$ such that $\alpha(I) \subsetneq \beta(J)$. The set of all inextendible dipaths in X is denoted as $\vec{P}_1(X)$.

In other words, since dipaths are compact, an inextendible path α starts at an initial point and ends at a final point.

Definition 4.2. Let I denote the unit-interval, and let J denote another interval.

1. A continuous map $H : J \times I \to X$ is called a *dihomotopy* if every partial map $H_t : J \to X$, $t \in I$, is an inextendible dipath.
2. Two inextendible dipaths $\alpha, \beta : J \to X$ are called *dihomotopic* if there is a dihomotopy $H : J \times I \to X$ with $H_0 = \alpha$ and $H_1 = \beta$. We write: $\alpha \overset{\rightarrow}{\simeq} \beta$.
3. The set of dihomotopy classes of inextendible dipaths in X is denoted as $\vec{\pi}_1(X)$.

Remark 4.3. 1. If H_0 starts in the initial point x, then (by inextendibility and continuity), all H_t have to start in the same point x. Analogously, they have to end in the same final point.
2. The local partial order need *not* be preserved wrt. to the variables in I.
3. Dihomotopy is obviously an equivalence relation.

Example 4.4. 1. Figure 17 represents a path from an initial to a final point that cannot be homotoped to a dipath.
2. Figures 18, 19 and 20 are different views of an example of two dipaths that are homotopic to each other (relative to the boundary), but *not dihomotopic* to each other:

   To explain the meaning of these two paths computer-scientifically, consider the following three terms.

- $T_1 = Pa.\mathtt{push}(a,1).Va.Pb.\mathtt{read}(b,u).\mathtt{read}(a,v).\mathtt{push}(b,u[1]+v).Vb.Pc.\mathtt{read}(b,u).\mathtt{push}(c,u[1]+u[2]).Vc$,
- $T_2 = Pb.\mathtt{read}(a,v).\mathtt{push}(b,v+1).Vb$,
- $T_3 = Pa.\mathtt{push}(a,3).Pb.\mathtt{read}(a,v).\mathtt{push}(b,0).Va.Pc.\mathtt{read}(b,u).\mathtt{push}(b,u[1]*v).\mathtt{push}(c,u[1]+u[2]).Vb.Vc$.

Then the first dipath (the one below the central hole in Figure 18) corresponds to the following schedule (where T_3 gets a before T_1 and T_2 gets into b after (T_1,T_3)),

T_1	T_2	T_3	$Values$
Pa	$-$	Pa	$a=3$
$-$	$-$	Pb	$b=(\square 0,0)$
$-$	$-$	Va	
Va	$-$	$-$	$a=1$
Pb	$-$	$-$	
Vb	$-$	$-$	$b=(1,\square 0)$
Pc	$-$	$-$	
Vc	$-$	$-$	$c=1$
$-$	$-$	Pc	
$-$	$-$	Vb	$b=(1,3)$
$-$	$-$	Vc	$\underline{c=4}$
$-$	Pb	$-$	
$-$	Vb	$-$	$b=(2,1)$

where "boxed" values (for b) are the places of this buffer which are holding a lock. The second dipath corresponds to (where T_3 gets a before T_1 and T_1 gets into b after (T_2,T_3)),

T_1	T_2	T_3	$Values$
Pa	$-$	Pa	$a=3$
$-$	$-$	Pb	$b=(\square 0,0)$
$-$	$-$	Va	
$-$	Pb	$-$	
$-$	Vb	$-$	$b=(4,\square 0)$
Va	$-$	$-$	$a=1$
Pb	$-$	$-$	
Vb	$-$	$-$	$b=(5,\square 0)$
Pc	$-$	$-$	
Vc	$-$	$-$	$c=5$
$-$	$-$	Pc	
$-$	$-$	Vb	$b=(5,15)$
$-$	$-$	Vc	$\underline{c=20}$

Let us assume the purpose of this program was to give a value for c then we see that these two homotopic (in the classical sense) but not dihomotopic dipaths give different results.

Definition 4.5. 1. The homotopy history of an inextendible dipath $\alpha : I \to X$ is defined as

$$h\alpha := \{y \in X | \exists \text{ a dipath } \beta \text{ through } y \text{ and } \alpha \simeq \beta\}.$$

2. Two points are *homotopy history equivalent* if

$$x \in h\alpha \Leftrightarrow y \in h\alpha \text{ for all } \alpha \in \vec{P}_1(X).$$

3. The *diconnected components* of X consists of the connected components (in the classical sense) of the homotopy history equivalence classes of X.

Remark 4.6. 1. Inextendible paths in the same dihomotopy class have the same homotopy history.

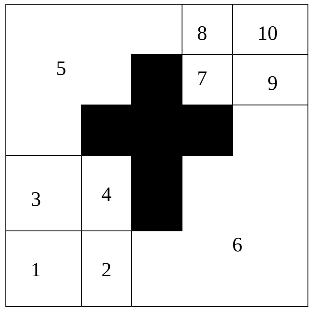

FIGURE 21. The "Swiss flag" example.

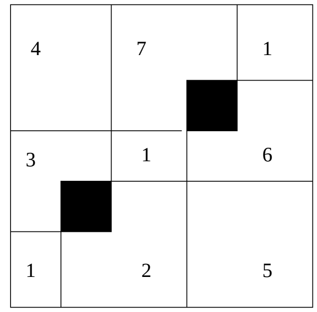

FIGURE 22. "Two partially ordered holes".

2. Two points $x, y \in X$ are history equivalent if and only if *every* dipath through x is dihomotopic to one through y and *every* dipath through y is dihomotopic to one through x.
3. If X is compact, the Boolean algebra generated by the homotopy histories is atomic and a homotopy history equivalence class C is an atom, i.e., it contains the points that are contained in the homotopy histories of certain dihomotopy classes but not in others, i.e., there is a subset $K \subset \vec{\pi}_1(X)$ such that

$$C = \bigcup_{\alpha \in K} h\alpha \setminus \bigcup_{\beta \notin K} h\beta.$$

Point 3 of the remark above is of primary importance for program analysis. Each $h\alpha$ corresponds to some property of accesses of shared ressources in the PV model. The decomposition of C shows that there are elementary regions of execution which are separated out by the properties paths going through them can have in the future (and in the past). We give examples below.

Example 4.7. 1. The complement of the "Swiss flag" in I^2 (see Figure 21) has 10 homotopy history components. All of them are pathwise connected. This gives the semantics of the program having process $T_1 = Pa.Pb.Vb.Va$ in parallel with $T_2 = Pb.Pa.Va.Vb$ (where a and b are 1-semaphores). In region 1, we still have all possible futures (all possible access histories to a and b). In region 2, we can only go to 4 or to 6, meaning that we are going to deadlock in the future or T_2 will get a and b before T_1. In region 6, we can only come from 2 and go to 9: T_2 has got a and b before T_1. In region 9, we can "come" from the unreachable region 7 or from 6. In region 10, we might have come from any history in the past.
2. The complement of "two partially ordered holes" in I^2 (see Figure 22) has 7 homotopy history components. One of them contains both the initial point $\mathbf{0}$, the final point $\mathbf{1}$, and an area in the middle. This homotopy history class decomposes into three diconnected components, all the others are pathwise connected. This pictures gives the semantics of the term $Pa.Va.Pb.Vb \mid Pa.Va.Pb.Vb$.
3. The "room with 3 barriers" in I^3 from Ex. 4.4 has 8 homotopy history components. The middle region from 2. decomposes into two spacial components.

Let again $(X, \mathcal{U})$ be a locally partially ordered topological space, and let $A, B \subset X$.

Definition 4.8. 1. A dipath from A to B is a dipath γ in X with $\gamma(0) \in A$ and $\gamma(1) \in B$.
2. Two dipaths γ, δ in X from A to B are *dihomotopic from A to B* if there is a homotopy $H : J \times I \to X$ with $H_0 = \gamma, H_1 = \delta$ and such that every map H_t is a dipath from A to B.
3. Dihomotopy from A to B is obviously an equivalence relation. The equivalence classes (*dihomotopy classes*) constitute the *dihomotopy set* $\vec{\pi}_1(X; A, B)$.

Remark 4.9. 1. In many relevant cases, A, resp. B will consist of a single point or a finite number of "initial", resp. "final" points. If I, resp. F consists of the initial, resp. final points of X, then $\vec{\pi}_1(X) = \vec{\pi}_1(X; I, F)$.

2. Concatenation of dipaths factors to yield concatenation on the level of dihomotopy classes $\vec{\pi}_1(X;A,B)\times\vec{\pi}_1(X;B,C)\to\vec{\pi}_1(X;A,C)$.

Definition 4.10. 1. S is called *di-1-connected from A to B* if $\vec{\pi}_1(S;A,B)$ consists of a single dihomotopy class.

5. Parameterized and Euclidean partial orders

In this section, we look at a particular subcategory of local po-spaces, where locally, the time ordering is the component-wise ordering in $\mathbb{R}^n$. These spaces are a special case of parameterized spaces in which it is possible to cut transversally to time to determine, using ordinary homotopy theory, obstructions to directed homotopy.

Definition 5.1. Let U be a set with a partial order $\leq$. A subset $V\subset U$ is called *achronal* if for all $x,y\in V: x\leq y\Rightarrow x=y$; compare [48].

Definition 5.2. Let $(X,\leq)$ denote a pospace.

1. We call $(X,\leq)$ *parameterized* if there is a (parameter) dimap $F: X\to\mathbb{R}$ such that $X_t := F^{-1}(t)$ is achronal for every $t\in\mathbb{R}$.
2. We call $\leq$ *Euclidean* if there are finitely many dimaps $f_i: X\to\mathbb{R}$ such that
$$\forall x,y\in X:\quad x<y\Leftrightarrow\forall i:\quad f_i(x)\leq f_i(y);\quad \exists i:\quad f_i(x)<f_i(y).$$
3. A local partial order on a topological space X is called *parameterized*, resp. *Euclidean* if it (or a refinement of it) consists of parameterized, resp. Euclidean partial orders.

Remark 5.3. For a Euclidean partial order the relation is given by comparison with the natural partial order on Euclidean space $\mathbb{R}^n$, defined for $\mathbf{x}=[x_1,\dots,x_n],\mathbf{y}=[y_1,\dots,y_n]\in\mathbb{R}^n$:
$$\mathbf{x}\leq\mathbf{y}\Leftrightarrow\forall i:\quad x_i\leq y_i.$$

Lemma 5.4. 1. *A parameterized po-space gives rise to a new partial order $\leq'$ defined by: $x\leq' y\Leftrightarrow F(x)\leq F(y)\in\mathbb{R}$.*
2. *A Euclidean partial order is parameterized.*

Proof.

1. Antisymmetry follows from the achronality of the X_t.
2. Let X and $f_i: X\to\mathbb{R}$ be given as in Def. 5.2.2. The function $F=\sum f_i:\quad U\to\mathbb{R}$ is a dimap, and for every $t\in\mathbb{R}$, the preimage $X_t := F^{-1}(t)\subset X$ is achronal.

□

For the rest of this section, let $(X,\leq,F)$ denote a parameterized pospace *foliating* X into "cuts" X_t, $t\in\mathbb{R}$. We shall moreover use F to reparameterize dipaths and dihomotopies to yield new parameterizations matching with that foliation:

Definition 5.5. 1. A dipath $\alpha: I\to X$ is called *well-parameterized* if $F(\alpha(t))=t$ for every $t\in I$.
2. A dihomotopy $H: J\times I\to X$ is called well-parameterized if every dipath $H_s: I\to X$, $s\in J$ is well-parameterized.
3. A dipath $\beta: I'\to X$ is called a *reparameterization* of another dipath α if there is a *strictly monotonic* map $h: I'\to I$ such that $\beta=\alpha\circ h$.

Almost as in any course on elementary differential geometry, we get easily:

Proposition 5.6. 1. *To any dipath $\alpha: I\to X$, there is (exactly one) well-parameterized reparameterization $\beta: I'\to X$.*
2. *To any dihomotopy $H: J\times I\to X$ there is (exactly one) well-parameterized reparameterization $\bar{H}: J\times I'\to X$.*

Proof. The map $s:=F\circ\alpha: I\to\mathbb{R}$ is strictly increasing and continuous with image an interval $I'\subset\mathbb{R}$. Let $t: I'\to I$ denote the (strictly monotonic and continuous) map inverse to s. Obviously, $\beta:=\alpha\circ t: I'\to X$ is (the only) well-parameterized reparameterization of α. The same argument goes through (levelwise) for a dihomotopy. □

Remark 5.7. 1. In fact, the proof for Prop. 5.6 needs to start with a *regular* dipath: $t_1 < t_2 \Rightarrow \alpha(t_1) \neq \alpha(t_2)$. But if a dipath is not regular, (i.e., constant on subintervals), one may first reparameterize it such that it gets regular by shortening the interval of definition.
2. In 2., one might have to use that X is a Hausdorff space. This is alright by Rem. 3.2.4.

The cuts $X_t := F^{-1}(t)$ will in general decompose into (classical) connected components. These can be used to obtain obstructions to dihomotopy:

Proposition 5.8. *Let $H : J \times I \to X$ denote a well-parameterized dihomotopy between two well-parameterized dipaths $\alpha, \alpha' : I \to X$. For every $t \in I$, $\alpha(t)$ and $\alpha'(t)$ are contained in* the same *component of X_t.*

Proof. The map $H(-,t) : J \to X_t$ is a path from $\alpha(t)$ to $\alpha'(t)$. □

Unfortunately, this criterion is not enough to describe, even in this restricted subcategory, what directed homotopy is, as we show in next example.

Example 5.9. Let X be the subset $[0,3] \times [0,3] \times [0,3] \setminus [1,2] \times [1,2] \times [0,3]$ in $\mathbb{R}^3$ with the standard partial order. There are two dihomotopy classes of paths from $(0,0,0)$ to $(3,3,3)$, but the cuts induced by $F(x,y,z) = x+y+z$ are all connected. Hence to get full information about dihomotopy classes, it does not suffice to study just one family of cuts. We conjecture that for cubical complexes (treated in next section), it suffices to know *all* families of cuts and their connected components, i.e., that cuts give all the obstructions to dihomotopy equivalence.

6. Cubical complexes as local po-spaces

Cubical complexes were defined by J.-P.Serre [54] and a theoretical framework is developped by R. Brown and P. J. Higgins [8], see also [27] for their use as models for higher dimensional automata (HDA). We show here that they are the natural combinatorial counterpart of local po-spaces (the centre of this is Theorem 6.24 and Proposition 6.37). This makes the link with more standard combinatorial techniques for reasoning about concurrent systems (interleaving ones or truly-concurrent ones like HDA).

6.1. **Cubical complexes.**

We start here with elementary definitions of cubical complexes, as found in e.g. Brown et al. [7] and [8]. We have added up a notion of "semi-cubical" complex (analogous to "semi-simplicial" complexes) which is the common part of all cubical complexes introduced by many authors (for a different category of cubical complexes, see [18]).

Definition 6.1. A semi-cubical complex M is a family of sets $\{M_n | n \geq 0\}$ with face maps $\partial_i^k : M_n \to M_{n-1}$ $(1 \leq i \leq n,\ k = 0,1)$ satisfying the semi-cubical relations:

$$\partial_i^k \partial_j^l \quad = \quad \partial_{j-1}^l \partial_i^k \quad (i < j)$$

Definition 6.2. A cubical complex K is a semi-cubical complex with degeneracy maps $\epsilon_i : K_{n-1} \to K_n$ $(1 \leq i \leq n)$ satisfying the cubical relations:

$$\begin{array}{rcll} \epsilon_i \epsilon_j & = & \epsilon_{j+1}\epsilon_i & (i \leq j) \\ \partial_i^k \epsilon_j & = & \left\{ \begin{array}{ll} \epsilon_{j-1}\partial_i^k & (i < j) \\ \epsilon_j \partial_{i-1}^k & (i > j) \\ Id & (i = j) \end{array} \right. & \end{array}$$

Definition 6.3. Let M and N be two semi-cubical sets, and f a family $f_n : M_n \to N_n$ of functions. f is a morphism of semi-cubical sets if

$$f_n \circ \partial_i^0 = \partial_i^0 \circ f_{n+1}$$
$$f_n \circ \partial_i^1 = \partial_i^1 \circ f_{n+1}$$

for all $n \in \mathbb{N}$ and $1 \leq i \leq n+1$.

This defines the category Υ_{sr} of semi-cubical sets.

We write Υ_{sr}^n for the full subcategory of Υ_{sr} consisting of semi-cubical sets whose elements are cubes of dimension less than or equal to n.

6.2. **The geometric realization of a semi-cubical set.** Let $\square_n$ be the standard cube in $\mathbb{R}^n$ $(n \geq 0)$,

$$\square_n = \{(t_1, \dots, t_n) | \forall i, 0 \leq t_i \leq 1\}$$
$$\square_0 = \{0\}$$

and let $\delta_i^k : \square_{n-1} \to \square_n$, $1 \leq i \leq n$, $k = 1, 2$, be the continuous functions $(n \geq 1)$,

$$\begin{tikzcd}
\square_n & \square_{n-1} \arrow[l, "\delta_i^0"'] \\
\square_{n-1} \arrow[u, "\delta_i^1"] &
\end{tikzcd}$$

defined by,

$$\delta_i^k(t_1, \dots, t_{n-1}) \quad = \quad (t_1, \dots, t_{i-1}, k, t_i, \dots, t_{n-1})$$

Then,

Lemma 6.4.

$$\delta_i^k \delta_j^l \quad = \quad \delta_{j+1}^l \delta_i^k \qquad (i \leq j)$$

Proof. Let $i \leq j$, and $(t_1, \dots, t_n) \in \square_n$. Then,

$$\begin{aligned}
\delta_i^k(\delta_j^l(t_1, \dots, t_n)) &= \delta_i^k(t_1, \dots, t_{j-1}, l, t_j, \dots, t_n) \\
&= (t_1, \dots, t_{i-1}, k, \dots, l, t_j, \dots, t_n) \\
&= \delta_{j+1}^l(\delta_i^k(t_1, \dots, t_n))
\end{aligned}$$

$\square$

We notice that δ^k verify the dual equations that ∂^k verify in all semi-cubical sets.

Consider now, for a semi-cubical set M, the set $\mathbf{R}(M) = \coprod_n M_n \times \square_n$. The sets M_n have the discrete topology and $\square_n$ is topologized as a subset of $\mathbb{R}^n$ with the standard topology thus $\mathbf{R}(M)$ is a topological space with the disjoint sum topology.
Let $\equiv$ be the equivalence relation induced by the identities:

$$\forall k, i, n, \forall x \in M_{n+1}, \forall t \in \square_n, n \geq 0, (\partial_i^k(x), t) \equiv (x, \delta_i^k(t))$$

Let $\mid M \mid = \mathbf{R}(M)/ \equiv$ have the quotient topology. The topological space $\mid M \mid$ is called the *geometric realization* of M. And moreover, M can be thought of as a labelling of a subdivision of $\mid M \mid$ into cubes. An element $y_m \in M_m$ is the label of an m-cube in $\mid M \mid$. Let $p \in \mid M \mid$, then there is a minimal cube in the subdivision of $\mid M \mid$ containing p, namely the unique cube $x \times \square_k$ which has p in the interior. We call x the *carrier* of p.

Definition 6.5. The open star of a point $p \in \mid M \mid$ with respect to the subdivision M is

$$\mathrm{St}(p, M) = \{q \in \mid M \mid \,|\, \text{carrier } (p) \text{ is a face of carrier } (q)\}$$

For a cube $x \in M_n$ we define the open star

$$\mathrm{St}(x, M) = \{y \in M | \exists (k_1, l_1), \dots, (k_i, l_i), \partial_{l_1}^{k_1} \dots \partial_{l_i}^{k_i}(y) = x\}$$

The upper star of x is

$$\mathrm{St}^+(x, M) = \{y \in M | \exists l_1, \dots, l_i, \partial_{l_1}^0 \dots \partial_{l_i}^0(y) = x\}$$

The lower star is

$$\mathrm{St}^-(x, M) = \{y \in M | \exists l_1, \dots, l_i, \partial_{l_1}^1 \dots \partial_{l_i}^1(y) = x\}$$

Remark 6.6. 1. If $x \in M_n$ is the carrier of $p \in \mid M \mid$, then $\mathrm{St}(p, M)$ is the union of the interiors of cubes in $\mid M \mid$ which are labelled by an element of $\mathrm{St}(x, M)$. We define the upper and lower star of p as the union of the interior of cubes in $\mid M \mid$ labeled by elements from respectively the upper and lower star of x.
2. By an abuse of notation, we will omit M and write $\mathrm{St}(p)$ and $\mathrm{St}(x)$if there is no risk of confusion.
3. If p and q in $\mid M \mid$ have the same carrier, i.e., if they are in the interior of the same cube, then $\mathrm{St}(p) = \mathrm{St}(q)$

6.3. **The geometric realization functor and locally po-spaces.** First we have to define some technical intermediary steps.

Definition 6.7. Let M be a semi-cubical complex and y an element of M. Then $x \in M$ is a face of y if there exists a collection of indices $k_1, \cdots, k_1$ being 0 or 1 and $l_1, \cdots, l_i$ integers such that $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} y$. This is equivalent to saying that y is in the star of x ($y \in \mathrm{St}(x, M)$).

To take advantage of the geometrical intuition, we will consider the realization $|M|$. Hence given $y \in M_n$ and its tree of boundaries $\partial_{l_1}^{k_1} \ldots \partial_{l_m}^{k_m} y$ we may think of this as the n-cube $\square_n$ in $|M|$ labelled y and its iterated boundaries. To be precise, $x = \partial_i^k y$ means that the $n-1$-cube $\square_{n-1}$ labelled x is identified with $\{(t_1, \ldots, t_n) \in \square_n | t_i = k\}$. The commutator rules for the boundaries ensures that this works for iterated boundaries:

Lemma 6.8. *Let $x = \partial_{l_1}^{k_1} \ldots \partial_{l_m}^{k_m} y$. Then there is a (not necessarily unique) canonical form $\partial_{l'_1}^{k'_1} \ldots \partial_{l'_m}^{k'_m} y$ with $l'_1 < l'_2 < \ldots < l'_m$, and x is identified with $\{(t_1, \ldots, t_n) \in \square_n | t_{l'_i} = k_i$ for $i = 1, \ldots, m\}$,*

Proof. Use the commutator relations. □

To support this view, we prove that the combinatorial model of the iterated boundaries is "right".

Definition 6.9. Let $D_{[n]}$ be the free semi-cubical complex generated by a unique n-cube $I_{[n]}$, i.e. the semi-cubical complex of faces of $I_{[n]}$ which is formally,

- $(D_{[n]})_j = \{\partial_{l_1}^{k_1} \cdots \partial_{l_{n-j}}^{k_{n-j}} (I_{[n]}) \mid k_i = 0, 1, l_1 < \cdots < l_{n-j}\}$,
- The boundary operations are concatenations of the operator and of the element of $D_{[n]}$ itself.

Then,

Lemma 6.10. • *$\mid D_{[n]} \mid$ is homeomorphic to the space $\square_n$ with the usual topology in $\mathbb{R}^n$,*
- *Any morphism of semi-cubical complexes $\sigma : D_{[n]} \to M$ (we call this a singular cube) induces a continuous map $\mid \sigma \mid : \square_n \to \mid M \mid$.*

Proof. Let us consider the following map:

$$\begin{array}{cccc} f_{I_{[n]}}: & \mid D_{[n]} \mid & \to & \square_n \\ & (\partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i}(I_{[n]}), v) & \to & \delta_{l_i}^{k_i} \cdots \delta_{l_1}^{k_1}(v) \end{array}$$

In $\mid D_{[n]} \mid$ we identify points of the form $(\partial_l^k \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i}(I_{[n]}), v)$ with $(\partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i}(I_{[n]}), \delta_l^k(v))$. By definition $f(\partial_l^k \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i}(I_{[n]}), v) = \delta_{l_i}^{k_i} \cdots \delta_{l_1}^{k_1} \delta_l^k(v) = f(\partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i}(I_{[n]}), \delta_l^k(v))$. This implies that f is well defined.

To see, that f is injective, suppose we have two points in $D_{[n]}$, $x = (\partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i}(I_{[n]}), v)$ and $y = (\partial_{l'_1}^{k'_1} \cdots \partial_{l'_{i'}}^{k'_{i'}}(I_{[n]}), v')$ such that $f_{I_{[n]}}(x) = f_{I_{[n]}}(y)$. We assume without lack of generality, that $v \in \mathring{\square}_{n-i}$ and $v' \in \mathring{\square}_{n-i'}$. Using the commutator rules (which is ok by the identifications made in the geometric realisation), we may also assume that $l_i < \cdots < l_1$ and $l'_{i'} < \cdots < l'_1$.

With these assumptions, $f(x) = f(y)$, i.e., $\delta_{l_i}^{k_i} \cdots \delta_{l_1}^{k_1}(v) = \delta_{l'_{i'}}^{k'_{i'}} \cdots \delta_{l'_1}^{k'_1}(v) = (t_1, \ldots, t_n) \in \square_n$ means $t_j = 1$ if and only if there is a μ and a μ' such that $l_\mu = l_{\mu'} = j$ and $k_\mu = k_{\mu'} = 1$ Similarly for $t_j = 0$. Thus $i' = i$ and $l_j = l'_j$ and $k_j = k'_j$ for $j = 1, \ldots, i$

f is also surjective in an easy manner, therefore f is a bijection. Furthermore, f is continuous on each face $(\partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i}(I_{[n]})), v)$ $(v \in \mathring{\Box}_{n-i})$ of $\mid D_{[n]} \mid$ with value in a face of $\Box_n$. This implies that f is continuous and it is easy to see that f^{-1} is also continuous: f is an homeomorphism.

Then, given σ, it suffices to see that $\sigma(I_{[n]}) = x \in M_n$ is an n-cube of M and that $\mid \sigma \mid$ is f_x, thus is a continuous map. □

The non uniqueness of the canonical form in Lemma 6.8 arises when M has other identifications of the faces of y than the ones induced by the commutator relations, and hence in $|M|$ we glue faces of the same cube to eachother. To avoid this, we define

Definition 6.11. Let M a semi-cubical complex. M is a non singular cubical complex if for all its n-cubes x, $\partial_l^k(x) = \partial_{l'}^{k'}(x)$ implies $k \neq k'$.

Remark 6.12. In the geometric realization $\mid M \mid$, the requirement that M is non singular ensures that two faces of an n-cube are identified only if one is an upper face and the other is a lower face, thus giving rise to a loop.

Definition 6.13. Let M a semi-cubical complex. M is a non self-linked cubical complex if for all its n-cubes x, $\partial_l^k(x) = \partial_{l'}^{k'}(x)$ implies $k = k'$ and $l = l'$.

Remark 6.14. One may still have loops in $\mid M \mid$, but they will always consist of more than one cube. Hence each $y \in M_n$ has a full subtree of iterated boundaries with $2\binom{n}{k}$ vertices in M_{n-k}, as does indeed an n-cube and its iterated boundaries.

Now the canonical form is unique:

Lemma 6.15. *Let M be a non self-linked cubical complex, and x, y be elements of M. Suppose x is a face of y. Then x can be written in a unique manner as,*

- $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} y$ *with* $k_1, \cdots, k_i = 0, 1$ *and* $l_1 < l_2 < \cdots < l_i$ *("canonical form").*
- $x = \partial_{l'_1}^{0} \cdots \partial_{l'_j}^{0} \partial_{l'_{j+1}}^{1} \cdots \partial_{l'_i}^{1} y$ *with* $l'_1 < l'_2 < \cdots < l'_j$ *and* $l'_{j+1} < l'_{j+2} < \cdots < l'_i$.

All other "decompositions" of x as a face of y, $x = \partial_{v_1}^{u_1} \cdots \partial_{v_i}^{u_i} y$ verify the following: let $K_0(x,y)$ (respectively $K_1(x,y)$) be the cardinal of the set $\{j/1 \leq j \leq i, k_j = 0\}$ (respectively $\{j/1 \leq j \leq i, k_j = 1\}$), then $K_0(x,y)$ (respectively $K_1(x,y)$) is also the cardinal of $\{j/1 \leq j \leq i, u_j = 0\}$ (respectively $\{j/1 \leq j \leq i, u_j = 1\}$).

Proof. By induction on i (the length of the decomposition). The statement about K_0 and K_1 follows from the fact that these are invariant under commutation following the commutator rules. □

Lemma 6.16. *Let M be a non self-linked cubical complex, x and y two of its elements. The relation x is a face of y ("$x <_F y$") is a partial order.*

Proof. It is reflexive indeed.

Now, if $x <_F y$ and $y <_F x$ then $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} y$ and $y = \partial_{l'_1}^{k'_1} \cdots \partial_{l'_j}^{k'_j} x$ by definition. So $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} \partial_{l'_1}^{k'_1} \cdots \partial_{l'_j}^{k'_j} x$ and $K_0(x,x) = K_0(x,y) + K_0(y,x)$ and $K_1(x,x) = K_1(x,y) + K_1(y,x)$ by Lemma 6.15. But $K_0(x,x) = K_1(x,x) = 0$ again by Lemma 6.15 so are $K_0(x,y)$, $K_1(x,y)$, $K_0(y,x)$ and $K_1(y,x)$. Therefore $x = y$.

Finally, if $x <_F y$ and $y <_F z$ then $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} y$ and $y = \partial_{l'_1}^{k'_1} \cdots \partial_{l'_j}^{k'_j} z$. So $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} \partial_{l'_1}^{k'_1} \cdots \partial_{l'_j}^{k'_j} z$ and $x <_F z$. □

Lemma 6.17. *Let M be a non self-linked cubical complex, x, y and z three elements of M. Then $x <_F y <_F z$ implies $K_0(x,z) = K_0(x,y) + K_0(y,z)$ and $K_1(x,z) = K_1(x,y) + K_1(y,z)$.*

Proof. If $x <_F y$ and $y <_F z$ then $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} y$ and $y = \partial_{l'_1}^{k'_1} \cdots \partial_{l'_j}^{k'_j} z$. So $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} \partial_{l'_1}^{k'_1} \cdots \partial_{l'_j}^{k'_j} z$ and the number of k_m or k'_m equal to 0 (respectively 1) in the decomposition above is the number of k_m equal to 0 (respectively 1) plus the number of k'_m equal to 0 (respectively 1), hence the result. $\square$

The face ordering has nice properties that we will exploit later on.

Lemma 6.18. *Let M be a non-singular cubical complex and x, y be two elements of M such that $x <_F y$. Then there is a projection operator (which is a dimap) p^y_x of cube $\mid y \mid$ in $\mid M \mid$ whose carrier is y on the cube $\mid x \mid$ such that, if $x = \partial^{k_1}_{l_1} \cdots \partial^{k_j}_{l_j} y$, $p^y_x(y, \delta^{k_j}_{l_j} \cdots \delta^{k_1}_{l_1} v) = (x, v)$.*

Proof. In $\mid M \mid$, points (x,t) are identified with $(y, \delta^{k_j}_{l_j} \cdots \delta^{k_1}_{l_1}(t))$. Let p be the projection from $\mathbb{R}^n$ to $\mathbb{R}^m$ (if $y \in M_n$, $x \in M_m$, we have $t \in \mathbb{R}^m$ and $\delta^{k_j}_{l_j} \cdots \delta^{k_1}_{l_1}(t) \in R^n$) which projects out coordinates $l_1, \cdots, l_j$. Then set $p^y_x(y,u) = (x, p(u))$, then $p^y_x(y, \delta^{k_j}_{l_j} \cdots \delta^{k_1}_{l_1} v) = (x,v)$ thus $p^y_x(x,v) = (x,v)$. p is continuous and monotonic so p^y_x is a dimap. $\square$

Lemma 6.19. *Let M be a non self-linked cubical complex. Then for all $y \in M$, for all faces b and b' of y in M, we have only two possibilities,*

- *b and b' have no face in common (we write $b \cap b' = \emptyset$),*
- *or b and b' have a maximal (with respect to the partial order $<_F$) face in common (that we write $b \cap b'$), which is a face of y).*

Proof. Since M is non self-intersecting and we study iterated boundaries of $y \in M_n$, we can consider this a study of the n-cube $\square_n$ with no identifications of boundaries except the ones given by the geometry.

Two boundaries b and b' are then considered as subsets of $\{(t_1, \ldots, t_n) \in \square_n\}$. We write $b = \partial^{k_1}_{l_1} \cdots \partial^{k_j}_{l_j} y$ and $b' = \partial^{k'_1}_{l'_1} \cdots \partial^{k'_{j'}}_{l'_{j'}} y$ on canonical form. Then $b = \{(t_1, \ldots, t_n) \in \square_n | t_{l_i} = k_i$ *for* $i = 1 \ldots j\}$ and $b = \{(t_1, \ldots, t_n) \in \square_n | t_{l'_i} = k'_i$ for $i = 1 \ldots j'\}$ and the intersection $b \cap b' = \{(t_1, \ldots, t_n) \in \square_n | t_{l_i} = k_i$ for $i = 1 \ldots j$ and $t_{l'_i} = k'_i$ for $i = 1 \ldots j'\}$.

If for some $i \in \{1, \ldots, j\}$ and $i' \in \{1, \ldots j'\}$ $l_i = l'_{i'}$ and $k_i \neq k'_{i'}$, then $b \cap b' = \emptyset$. Otherwise, the description as a subset of $\square_n$ gives that $b \cap b'$ is a face.

$\square$

Let us fix M a non self-linked cubical complex now. Its geometric realization $\mid M \mid$ is made up of equivalence classes of points (x,t), $x \in M_n$, $t \in \square_n$ under the relations $(\partial^k_l(x), t) = (x, \delta^k_l(t))$.

We give a local partial order on $\mid M \mid$. Any point p in $\mid M \mid$ has a unique representative (x,t) where $x =$ carrier (p) and $t \in \mathring{\square}_n$ (for some n). We prove that there is a partial order $\leq_x$ on the open neighborhood $U^x = \mathrm{St}(p, M)$ of p whenever x is a vertex of M and that the covering by these partial orders gives rise to a local partial order. First, in the case where x is not necessarily a vertex, we can partially order any $(y,u) \in U^x$ (again, we choose a representative such that y is the carrier of the point) with any (x,t). We have $x = \partial^{k_1}_{l_1} \ldots \partial^{k_i}_{l_i}(y)$ because $y \in \mathrm{St}(x, M)$, so (x,t) is identified with $(y, \delta^{k_i}_{l_i} \ldots \delta^{k_1}_{l_1}(t))$.

Definition 6.20. We set,

$$(x,t) \leq_{U^x} (y,u) \text{ if } \delta^{k_i}_{l_i} \ldots \delta^{k_1}_{l_1}(t) \leq u \text{ in } \square_{n+i}$$

$$(y,u) \leq_{U^x} (x,t) \text{ if } \delta^{k_i}_{l_i} \ldots \delta^{k_1}_{l_1}(t) \geq u \text{ in } \square_{n+i}$$

This is well-defined since the decomposition of x as a boundary of y is unique, because M is non self-linked.

Lemma 6.21.
- *Suppose $(x,t) \leq_{U^x} (y,u)$ and $x \neq y$, then necessarily, $x = \partial^0_{l_1} \ldots \partial^0_{l_j} y$ (where $j \geq 0$). This implies that $K_0(x,y) > 0$ and $K_1(x,y) = 0$.*
- *Suppose $(y,u) \leq_{U^x} (x,t)$ and $x \neq y$, then necessarily, $x = \partial^1_{l_1} \ldots \partial^1_{l_j} y$ (where $j \geq 0$). This implies that $K_0(x,y) = 0$ and $K_1(x,y) > 0$.*

Proof. We only prove the first statement since the proof of the other is similar. $y \in \mathrm{St}(x, M)$ so there is a collection of indices such that $x = \partial^{k_0}_{l_0} \ldots \partial^{k_j}_{l_j} y$ $(j \geq -1)$. j cannot be equal to -1 since $x \neq y$. Suppose now that there is an index k_i $(0 \leq i \leq j)$ which is equal to one. Then $u \in \mathring{\square}_n$ ($n \geq 1$ since $n = 0$ is only possible when x is a vertex and $x = y$), therefore all coordinates u_i of u are strictly less than 1, so in particular $u_{l_i} < (\delta^{k_i}_{l_i}(t))_{l_i} = 1$ by definition of the operator $\delta^{k_i}_{l_i}$. This is a contradiction with the definition of $\leq_{U^x}$. □

We now define another useful relation on the points of the geometric realization of M.

Definition 6.22. Let x be a vertex of M and let (z, v) be a point in U^x with carrier z. We say $(z, v) \leq_x (y, u)$ if there exists b in the star of x and t such that $(z, v) \leq_{U^b} (b, t) \leq_{U^b} (y, u)$.

This relation subsumes the relation U^x in an obvious manner and can be characterized as follows,

Lemma 6.23. *Suppose $(z, v) \leq_x (y, u)$, that is, $\exists (b, t)$, $b \in St(x, M)$, $(z, v) \leq_{U^b} (b, t) \leq_{U^b} (y, u)$. Then we have the following cases,*

- **(a):** *$b = \partial^1_{l_1} \ldots \partial^1_{l_j} z$, for some collection of indices and $K_1(b, z) = j \geq 1$ ($K_0(b, z) = 0$); $b = \partial^0_{l'_1} \ldots \partial^0_{l'_{j'}} y$, for some collection of indices and $K_0(b, y) = j' \geq 1$ ($K_1(b, y') = 0$).*
- **(b):** *$b = z$, $b = \partial^0_{l'_1} \ldots \partial^0_{l'_j} y$, for some collection of indices and $K_0(b, y) = j \geq 1$ ($K_1(b, y') = 0$), and the relation above shrinks down to $(z, v) \leq_{U^z} (y, u)$.*
- **(c):** *$b = y$, $b = \partial^1_{l_1} \ldots \partial^1_{l_j} z$, for some collection of indices and $K_1(b, z) = j \geq 1$ ($K_0(b, z) = 0$), and the relation above shrinks down to $(z, v) \leq_{U^y} (y, u)$.*
- **(d):** *$y = z$ and the relation above shrinks down to $v \leq u$.*

Proof. This is entailed by Lemma 6.21. □

We will say in the sequel that (z, b, u) is in case (a), (b), (c) or (d) according to the criteria above. In order to simplify the proofs we will write in brief (for all cases (a), (b), (c) and (d)) $b = \partial^{\mathbf{1}}_* z$ and $b = \partial^{\mathbf{0}}_* y$ where * means a multiindex, $\mathbf{1} = (1, 1, \ldots, 1)$, $\mathbf{0} = (0, 0, \ldots, 0)$ and they may all be empty. Now we can state,

Theorem 6.24. *The geometric realization of a non self-linked cubical complex M defines a locally po-space with covering being $\{St(x, M)/x \in M_0\}$ and local partial order $\leq_x$ on $St(x, M)$.*

Proof. We check that $\leq_x$ is a partial order indeed for all x in M. First, the reflexivity is obvious.

Then we check the antisymmetry: suppose $(z, v) \leq_x (y, u)$ and $(y, u) \leq_x (z, v)$. This means there are (b, t) and (b', t') with $b \in \mathrm{St}(x, M)$ and $b' \in \mathrm{St}(x, M)$ such that,

$$(z, v) \leq_{U^b} (b, t) \leq_{U^b} (y, u) \leq_{U^{b'}} (b', t') \leq_{U^{b'}} (z, v)$$

$$b = \partial^{\mathbf{1}}_* z,\ b = \partial^{\mathbf{0}}_* y,\ b' = \partial^{\mathbf{1}}_* y,\ b' = \partial^{\mathbf{0}}_* z$$

and moreover

$$x = \partial^*_* b,\ x = \partial^*_* b'$$

where * means a multiindex, $\mathbf{1} = (1, 1, \ldots, 1)$, $\mathbf{0} = (0, 0, \ldots, 0)$ and they may all be empty. Hence composing boundary maps we see that the following equalities hold:

1. $K_0(x, z) = K_0(x, b)$ and $K_0(x, z) = K_0(x, b') + K_0(b', z)$
2. $K_0(x, y) = K_0(x, b) + K_0(b, y)$ and $K_0(x, y) = K_0(x, b')$
3. $K_1(x, y) = K_1(x, b)$ and $K_1(x, y) = K_1(x, b') + K_1(b', y)$
4. $K_1(x, z) = K_1(x, b) + K_1(b, z)$ and $K_1(x, z) = K_1(x, b')$

Now 1 and 2 imply $K_0(b, y) + K_0(b', z) = 0$ and thus both are 0 which give $b = y$ and $b' = z$. Similarly 2 and 3 imply that $K_1(b', y) + K_1(b, z) = 0$ and thus $b' = y$ and $b = z$. This means that (z, b, y) is in case (d) and (y, b', z) is in case (d). We have $y = z$, $v \leq u$ and $u \leq v$. Thus $(z, v) = (y, u)$.

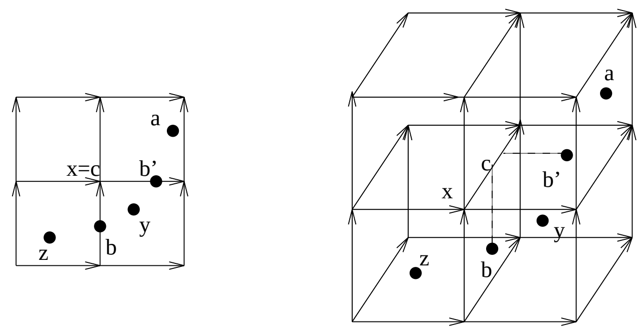

FIGURE 23. Illustration of the proof.

We check the transitivity now. Suppose $(z, v) \leq_x (y, u) \leq_x (a, w)$. This means there are (b, t) and (b', t') with $b \in \mathrm{St}(x, M)$ and $b' \in \mathrm{St}(x, M)$ such that,

$$(z, v) \leq_{U^b} (b, t) \leq_{U^b} (y, u) \leq_{U^{b'}} (b', t') \leq_{U^{b'}} (a, w)$$

with

$$b = \partial_*^{\mathbf{1}} z,\ b = \partial_*^{\mathbf{0}} y,\ b' = \partial_*^{\mathbf{1}} y,\ b' = \partial_*^{\mathbf{0}} a$$

and moreover

$$x = \partial_*^* b,\ x = \partial_*^* b'$$

where $*$ means a multiindex, $\mathbf{1} = (1, 1, \ldots, 1)$, $\mathbf{0} = (0, 0, \ldots, 0)$ and they may all be empty.

By Lemma 6.19, the intersection of b and b' is a face c of y containing x (see Figure 23). The inclusion of c into b and b' (which are compositions of [dual] boundary operators δ_l^k) define projections p and p' of points in b and b' onto points of c by Lemma 6.18. Define $d = p(t)$. Necessarily (looking at the coordinates in y, as p is a dimap), $p(t') \geq d$. Now it is enough to see that,

$$(z, v) \leq_{U^c} (c, d) \leq_{U^c} (a, w)$$

(hence the transitivity in that case). The first inequality is implied by the fact that $(b, t) \leq_{U^c} (c, d)$ since c is an end boundary of b and d is the corresponding projection of t. The second is implied by the fact that $(c, d) \leq_{U^c} (b', t')$ since c is a start boundary of b' and $p(t') \geq d$.

To see that these partial orders give rise to a local partial order, notice that for $p, q \in \mid M \mid$, the intersection $\mathrm{St}(p) \cap \mathrm{St}(q)$ is the interior of a cube $\Box_N$ which has the carrier x of p and the carrier y of q on the boundary. For any pair of points in this intersection, the partial orders$\leq_x$ and $\leq_y$ are both given by the partial order on the cube $\Box_N$.

$\Box$

To give an idea of some of the locally po-spaces we can construct consider this;

Example 6.25. Let M be the cubical complex

$$M_2 = A, B, C, D, M_1 = a, b, c, d, e, f, g, h, M_0 = p, q, r, s$$

$$d_1^0 A = d_1^0 C = a,\ d_2^0 A = d_2^0 B = b,\ d_1^0 B = d_1^0 D = c,\ d_2^0 D = d_2^0 C = d$$
$$d_1^1 A = d_1^1 D = e,\ d_2^1 A = d_2^1 D = f,\ d_1^1 B = d_1^1 C = g,\ d_2^1 B = d_2^1 C = h$$

$$d_1^0 a = d_1^0 b = d_1^0 c = d_1^0 d = p,\ d_1^1 a = d_1^1 c = d_1^0 f = d_1^0 h = q,$$
$$d_1^1 b = d_1^1 d = d_1^0 e = d_1^0 g = r,\ d_1^1 e = d_1^1 f = d_1^1 g = d_1^1 h = s$$

Then $\mid M \mid$ is the projective plane, and one can give cubical models for projective spaces of all dimensions in the same way.

Remark 6.26. Notice that we only need a covering with opens of the form $\mathrm{St}(x, M)$ for x vertices of M and that $\leq_x$ is defined also for x vertices, whereas the relation $\leq_{U^b}$ has to be defined for b being any n-cube of M.

These results actually applies to more cubical spaces since we have the following lemmas, using the concept of subdivision.

Definition 6.27. Let K be a cubical complex and K' be another cubical complex. Then K' is a subdivision of K if there is a dihomeomorphim $f :\mid K' \mid \rightarrow \mid K \mid$ (meaning that f and f^{-1} are dimaps) such that,

- $\forall x \in K'_n$, $\exists y \in K_n$, $f(x, \square_n) \subseteq (y, \square_m)$,
- $\forall y \in K$, $\exists x_1, \cdots, x_k \in K'$, $(y, \square_m) = \bigcup_{i=1,\cdots,k} f(x_i, \square_{n_i})$.

Definition 6.28. We call standard n-dicube the topological space $\square_n$ with the covering $\mathbf{U} = \{\square_n\}$ and local partial order $\leq_{\square_n}$ being the partial order induced by the pointwise ordering in $\mathbb{R}^n$. The n-dicube is then a locally po-space.

Definition 6.29. Let M be a locally po-space. A singular n-dicube is any dimap from the standard n-dicube to M.

Lemma 6.30. *Let K be a cubical complex. The "barycentric subdivision" of K is defined as follows. Consider the singular n-dicubes of $\mid K \mid$, $\sigma_x : \square_n \rightarrow \mid K \mid$, $\sigma_x(t) = (x, t)$, and the 2^n functions,*

$$s_{b_1,\cdots,b_n} : \square_n \rightarrow \square_n$$

for $(b_1, \cdots, b_n) \in \{0,1\}^n$ with

$$s_{b_1,\cdots,b_n}(t_1, \cdots, t_n) = \left(\frac{t_1 + b_1}{2}, \cdots, \frac{t_n + b_n}{2}\right)$$

Then the subcomplex SdK of $\mid K \mid$ with,

$$(SdK)_n = \{\sigma_x \circ s_{b_1,\cdots,b_n} / x \in K, (b_1, \cdots, b_n) \in \{0,1\}^n\}$$

is a subdivision of K, called the barycentric subdivision of K.

Proof. Let $f :\mid SdK \mid \rightarrow \mid K \mid$ defined as follows. Elements of $\mid SdK \mid$ are of the form $x = (u, v)$ with $u \in (SdK)_n$, i.e. $u = \sigma_y \circ s_{b_1,\cdots,b_n}$ ($b_i = 0, 1$, $y \in K_n$), and $v \in \mathring{\square}_n$. We set $f(x) = (y, w)$ with $w \in \mathring{\square}_n$ and $w_i = \frac{v_i + b_i}{2}$ ($i = 1, \cdots, n$).

For all such $x = (u, v) \in SdK$, $f(x) \in (y, \square_n)$ with the y defined above. Also for the same y, $(y, \square_n) = \bigcup_{b_1,\cdots,b_n=0,1} f(\sigma_y \circ s_{b_1,\cdots,b_n}, \square_n)$.

Let now $g :\mid K \mid \rightarrow \mid SdK \mid$ defined as follows. Points of $\mid K \mid$ are of the form $z = (y, w)$ with $y \in K_n$ and $w \in \mathring{\square}_n$. Set $g(z) = (\sigma_y \circ s_{b_1,\cdots,b_n}, 2w_i - b_i)$ with $b_i = \begin{cases} 0 & \text{if } 0 \leq w_i < \frac{1}{2} \\ 1 & \text{if } \frac{1}{2} \leq w_i \leq 1 \end{cases}$ ($i = 1, \cdots, n$). Then f and g are continuous maps, $f \circ g = Id = g \circ f$. $\square$

Lemma 6.31. *Let M be a non singular cubical complex. Then $Sd(M)$ is non self-linked.*

Proof. Let $\mid M \mid$ be the realization of the cubical complex $\mid M \mid$. Then for $x \in M_n$ the map $\sigma_x : \square_n \rightarrow \mid M \mid$ is injective on the interior of $\square_n$ by construction of $\mid M \mid$.

On the boundary of $\square_n$ there may be identifications corresponding to $\partial_i^k(x) = \partial_j^l(x)$. When $k \neq l$, this will identify boundaries of *different* cubes in the barycentric subdivision: $\partial_i^k(\sigma_x \circ s_{b_1,\ldots,b_{k-1},k,b_{k+1},\ldots,b_n}) = \partial_j^l(\sigma_x \circ s_{b_1,\ldots,b_{k-1},l,b_{k+1}\ldots,b_n})$ $\square$

Now using both Lemma 6.31 and Theorem 6.24 we can give a local po-space structure to any non-singular cubical complex.

6.4. **The singular cube functor and locally po-spaces.** Notice first that geometric realization is a functor. We are going to construct a right-adjoint to it in this section.

Lemma 6.32. *Let $f : X \longrightarrow Y$ be a morphism between the two semi-cubical sets X and Y. Then f induces a continuous map $\mid f \mid$ from $\mid X \mid$ to $\mid Y \mid$.*

Proof. Define $\mathbf{R}(f) : \mathbf{R}(X) \longrightarrow \mathbf{R}(Y)$ by: $\mathbf{R}(f)((x, t)) = (f(x), t)$. It is obviously a continuous map.

Suppose $(x, t) \equiv (y, s)$. Then there exists $(y_1, s_1), \ldots, (y_u, s_u)$ such that $(y_1, s_1) = (x, t)$, $(y_u, s_u) = (y, s)$ and $\forall g$, $\exists k, j$, $d_j^k(y_g) = y_h$ and $s_g = \delta_j^k(s_h)$ with $h = g + 1$ or $h + 1 = g$.

We show by induction on u that $\mathbf{R}(f)((x,t)) \equiv \mathbf{R}(f)((y,s))$, thus inducing a map from $\mid X \mid$ to $\mid Y \mid$. It holds trivially for $u=1$. To prove the inductionstep it suffices to see that $\mathbf{R}(f)((x,t)) \equiv \mathbf{R}(f)((y_2,s_2))$.

Suppose $\exists k,j$, $\partial_j^k(x) = y_2$ and $t = \delta_j^k(s_2)$. But $\partial_j^k(f(x)) = f(\partial_j^k(x))$. Thus, $\partial_j^k(f(x)) = f(y_2)$ and t=$\delta_j^k(s_2)$, which proves the result.

□

Then the geometric realization is still a functor when used with value in locally po-spaces.

Proposition 6.33. *Let $f : M \rightarrow N$ be a morphism of semi-cubical sets. Then $\mid f \mid : \mid M \mid \rightarrow \mid N \mid$ is a dimap.*

Proof. Recall that $\mid f \mid (x,t) = (f(x),t)$ for all $x \in M_n$ and $t \in \square_n$ (and for all n). Consider the "partial order" $\leq_{U^x}$ first. Suppose $(x,t) \leq_{U^x} (y,u)$. This means that $x = \partial_{l_1}^{k_1} \ldots \partial_{l_i}^{k_i}(y)$ and $\delta_{l_i}^{k_i} \ldots \delta_{l_1}^{k_1}(t) \leq u$. But f is a morphism of cubical complexes so $f(x) = \partial_{l_1}^{k_1} \ldots \partial_{l_i}^{k_i}(f(y))$. So $(f(x),t) \leq_{U^{f(x)}} (f(y),u)$. Then it is easy to check that more generally $z \leq_x y$ implies $f(z) \leq_{f(x)} f(y)$. □

Not surprisingly, the geometric realization represents n-cubes by the standard n-dicube.

Proposition 6.34.
- *$\mid D_{[n]} \mid$ is in fact (a refinement of) the po-space $\square_n$ with the componentwise partial order in $\mathbb{R}^n$,*
- *any singular cube $\sigma_x : D_{[n]} \rightarrow M$ induces a dimap $\mid \sigma_x \mid : \square_n \rightarrow \mid M \mid$ (i.e. a singular n-dicube).*

Proof. The unique n-cube $I_{[n]}$ of $D_{[n]}$ is geometrically realized as the interior of $\square_n$ with the right partial order since inside the n-cube the local partial-order is defined by case (d) of Lemma 6.23. Let us consider again the map

$$\begin{array}{llll} f_{I_{[n]}} : & \mid D_{[n]} \mid & \rightarrow & \square_n \\ & (\partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i}(I_{[n]}), v) & \rightarrow & \delta_{l_i}^{k_i} \cdots \delta_{l_1}^{k_1}(v) \end{array}$$

We know by Lemma 6.10 that f is an homeomorphism. We now have to see that f and f^{-1} are dimaps as well.

Let y be a face of $I_{[n]}$, i.e. any element of $D_{[n]}$ and x a face of y, $f_{I_{[n]}}(x,t) \leq f_{[n]}(y,u)$ (respectively $f_{I_{[n]}}(y,u) \leq f_{I_{[n]}}(x,t)$) is equivalent to $(x,t) \leq_{U^x} (y,u)$ (respectively $(y,u) \leq_{U^x} (x,t)$). To see this, let $y = \partial_{v_1}^{u_1} \cdots \partial_{v_j}^{u_j}(I_{[n]})$ and $x = \partial_{l_1}^{k_1} \ldots \partial_{l_i}^{k_i}(y)$. $(x,t) \leq_{U^x} (y,u)$ is equivalent to $\delta_{l_i}^{k_i} \ldots \delta_{l_1}^{k_1}(t) \leq u$ then to $f_{I_{[n]}}(y, \delta_{l_i}^{k_i} \cdots \delta_{l_1}^{k_1}(t)) \leq f_{I_{[n]}}(y,u)$ by monotony of f. But $f_{I_{[n]}}(x,t) = f_{I_{[n]}}(y, \delta_{l_i}^{k_i} \cdots \delta_{l_1}^{k_1}(t))$ so this is equivalent to $f_{I_{[n]}}(x,t) \leq f_{I_{[n]}}(y,u)$.

Let us consider now (z,v) and (y,u) be any element of $\mid D_{[n]} \mid$. Suppose $(z,v) \leq_x (y,u)$ for some vertex x of $D_{[n]}$. Then there exists $(b,t) \in \mid D_{[n]} \mid$ such that $(z,v) \leq_{U^b} (b,t) \leq_{U^b} (y,u)$, hence $f_{I_{[n]}}(z,v) \leq f_{I_{[n]}}(b,t) \leq f_{I_{[n]}}(y,u)$. Inversely, suppose that we have (z,v) and (y,u) such that $f_{I_{[n]}}(z,v) \leq f_{I_{[n]}}(y,u)$ and z and y belong to $\mathrm{St}(x, D_{[n]})$ for some vertex x of $D_{[n]}$. Then by Lemma 6.19 there exists a maximal common face b between y and z since they are both faces of $I_{[n]}$. Furthermore this maximal face is such that $K_0(b,y) = 0$ and $K_1(b,z) = 0$, so b can be decomposed as $b = \partial_{f_1}^1 \cdots \partial_{f_k}^1(y)$ and $b = \partial_{f'_1}^0 \cdots \partial_{f'_{k'}}^0(z)$. So we have $u \leq \delta_{f_k}^1 \cdots \delta_{f_1}^1 p_b^y(u)$, i.e. $(y,u) \leq_{U^b} (b, p_b^y(u))$ (using the projection defined in Lemma 6.18). We also have $\delta_{f'_{k'}}^0 \cdots \delta_{f'_1}^0 p_b^z v \leq v$ thus $(b, p_b^z(v)) \leq_{U^b} (z,v)$. Now, $p_b^y(u) \leq p_b^z(v)$ because otherwise, looking at the coordinates of $f(y,u)$ and $f(z,v)$ we cannot have $f(y,u) \leq f(z,v)$. Therefore $(y,u) \leq_{U^b} (b, p_b^y u) \leq_{U^b} (z,v)$ hence $(y,u) \leq_x (z,v)$.

The second statement is obvious since $\mid \sigma_x \mid$ is f_x. □

We are now ready for the definition of the singular cube functor.

Lemma 6.35. *Let $(M, \leq)$ be a locally partially ordered topological space. Define $S(M)$ to be the following graded set. For $n \in \mathbb{N}$, $S(M)_n$ is the set of singular n-dicubes of M together with the operators ∂_l^k such that $\partial_l^k(f) = f \circ \delta_l^k$. This gives $S(M)$ the structure of a semi-cubical complex.*

Proof. This is a standard proof [27]. $\square$

Similarly,

Lemma 6.36. *Let $(M, \leq)$ and $(N, \leq)$ be two locally partially ordered topological spaces and let $f : M \rightarrow N$ be a dimap. Then $S(f) : S(M) \rightarrow S(N)$ defined by, for all $x : \square_n \rightarrow M \in S$, $f(x) = f \circ x : \square_n \rightarrow N$, is a map of semi-cubical complexes. S defines a functor from the category of locally partially-ordered topological spaces to the category of semi-cubical complexes.*

Proof. This is obvious (the composition of dimaps is a dimap). $\square$

Proposition 6.37. *$|\ .\ |$ is left-adjoint to S.*

Proof. We prove that there exist two natural transformations

$$\eta : Id \rightarrow S(|\ .\ |)$$

$$\epsilon : | S | \rightarrow Id$$

(respectively the unit and counit of the adjunction) such that

$$S \xrightarrow{\eta S} S(| S |) \xrightarrow{S\epsilon} S$$

$$| \cdot | \xrightarrow{| \cdot | \eta} | S(| \cdot |) | \xrightarrow{\epsilon | \cdot |} | \cdot |$$

are the identity.

We can first show that:

$$(A) : \quad M \hookrightarrow S(| M |)$$

$$(B) : \quad | S(X) | \hookrightarrow X$$

in a natural manner for all M semi-cubical complex and X any local po-space. We begin by (A). For all n, we have the identity arrows on $\square_n$ which induce the isomorphisms: for all x, $Id : \square_n \rightarrow (x, \square_n)$. These in turn induce injective morphisms $f_x : \square_n \rightarrow | M |$, because M is an amalgamated sum of the $(x, \square_n)$. The $(f_x)_x$ form a subset N of $S(| M |)$. It is an easy exercise to show that N is closed under the action of the δ_i^k. Thus N is a sub-semi-cubical complex of $S(| M |)$. The naturality of the inclusion arrow $M \hookrightarrow S(| M |)$ is most obvious. This defines what is to be the unit of the adjunction.

Now, we come to (B). Elements of $S(X)_n$ are $f : \square_n \rightarrow X$. Now, $| S(X) |$ is an amalgamated sum of $(x, \square_n)$, $x \in S(X)_n$. The x induce on $\coprod\limits_x (x, \square_n)$ and then on $| S(X) |$ an injective morphism in the category of local po-spaces. It is an easy exercice to show that these arrows are natural in X. This defines what is to be the counit of the adjunction.

Then, we have to verify that two compositions of natural transformations are the identity. This is easy verification. $\square$

7. Combinatorial dihomotopy

In this section, we prove that dihomotopy can be studied combinatorially or geometrically whenever most convenient, at least when it comes to dipaths.

Definition 7.1. Let N be a cubical complex. A dipath in N is any sequence $p = (p_1, \cdots, p_k)$ of elements of N_1 such that for all i, $1 \leq i < k$, $\partial_1^1(p_i) = \partial_1^0(p_{i+1})$. $\partial_1^0(p_1)$ is the initial point of p. $\partial_1^1(p_k)$ is the final point of p.

Definition 7.2. Let N be a cubical complex and p, q two dipaths in N with the same initial and final points. We say that p and q are elementary dihomotopic if there exists A in N_2, k and j in $\mathbb{N}$ such that,

(1) $p = (p_1, \cdots, p_k)$ and $q = (q_1, \cdots, q_k)$,
(2) for all i, $1 \leq i < j$, $p_i = q_i$, and for all i, $j + 1 < i \leq k$, $p_i = q_i$,
(3) $p_j = \partial_1^0(A)$, $p_{j+1} = \partial_2^1(A)$, $q_j = \partial_2^0(A)$, $q_{j+1} = \partial_1^1(A)$.

Dihomotopy of (cubical) dipaths is the reflexive and transitive closure of elementary dihomotopy.

Using the local po-space structure defined in the previous theorems, we have also the following link with the combinatorial structure of M,

Proposition 7.3.
- *Any combinatorial dipath p in M induces a (topological) dipath $\mid p \mid$ in $\mid M \mid$,*
- *Any combinatorial dihomotopy between two paths p and q in M induces a (topological) dihomotopy between $\mid p \mid$ and $\mid q \mid$.*

Proof. Let $p = (p_1, \cdots, p_k)$ be a dipath in N. Let $x : I \rightarrow \mid N \mid$ defined by, $\forall i$, $0 \leq i \leq k-1$, $\forall t \in I$, $\frac{i}{k} \leq t \leq \frac{i+1}{k}$, $x(t) = \left(p_{i+1}, k\left(t - \frac{i}{k}\right)\right) \in \mid N \mid$.

Then x is a dimap (the local partial order in $\mid N \mid$ being defined as in Theorem 6.24),

- around each point in $(p_i, \mathring{\Box}_1)$ (for some i), the local partial order is the same as the one in I,
- around each "glueing point" $\partial_0^1(p_i)$, the partial order is the same as the one in I since $\partial_0^0(p_{i+1}) = \partial_0^1(p_i)$.
- also, for all t such that $\frac{i}{k} < t < \frac{i+2}{k}$, $x(t) \in \mathrm{St}(\partial_1^0(p_{i+1}), \Box_0) = \mathrm{St}(\partial_1^1(p_i), \Box_0)$.

We set $\mid p \mid = x$.

It is enough to prove now that if p and q are elementary dihomotopic, then $\mid p \mid$ and $\mid q \mid$ are dihomotopic. We suppose that we have $A \in N_2$, k, $l \in \mathbb{N}$ such that $p = (p_1, \cdots, p_k)$, $q = (p_1, \cdots, p_{l-1}, q_l, q_{l+1}, p_{l+2}, \cdots, p_k)$ and $\partial_0^0(A) = p_l$, $\partial_1^0(A) = q_l$, $\partial_0^1(A) = q_{l+1}$, $\partial_1^1(A) = p_{l+1}$. Now, define $H(\alpha, t)$ for $\alpha \in I$, $t \in I$ to be the map,

- for $0 \leq t \leq \frac{l-1}{k}$ and $\frac{l+1}{k} \leq t \leq 1$, $H(\alpha, t) = \mid p \mid (t) = \mid q \mid (t)$,
- for $\frac{l-1}{k} \leq t \leq \frac{l}{k}$, $H(\alpha, t) = (A, (\alpha(kt - l + 1), (1-\alpha)(kt - l + 1))) \in \mid N \mid$,
- for $\frac{l}{k} \leq t \leq \frac{l+1}{k}$, $H(\alpha, t) = (A, ((1-\alpha)(kt - l) + \alpha, \alpha(kt - l) + 1 - \alpha)) \in \mid N \mid$.

Then H is the desired dihomotopy between $\mid p \mid$ and $\mid q \mid$. $\Box$

Now we would like to prove some converse of this Proposition. Can we transfer geometric proofs made in the local po-space framework to the combinatorial world (hence more relevant to computer-scientific applications)? We need the concept of "approximation" to answer positively to this question.

Definition 7.4. Let K and L be two semi-cubical complexes and let $h : \mid K \mid \rightarrow \mid L \mid$ be a dimap. $f : K \rightarrow L$ is called a semi-cubical approximation of h if,

- f is a map of semi-cubical complexes,
- for all $v \in K_0$, $h(\mathrm{St}v) \subseteq \mathrm{St}f(v)$.

The general approximation theorem is more complex than in the simplicial case. We choose to prove only a weak version that is enough for linking (topological) dipaths in $\mid M \mid$ with combinatorial dipaths in M.

First, we define general subdivisions of $D_{[1]}$.

Definition and lemma 7.5. *Any semi-cubical complex S^k of the form,*

- $S_0^k = \{v_0, \ldots, v_k\}$,
- $S_1^k = \{u_1, \ldots, u_k\}$ *and* $\partial_0^0(u_i) = v_{i-1}$, $\partial_0^1(u_i) = v_i$.

is a non self-linked cubical complex and is a subdivision of $D_{[1]}$.

Proof. It is non self-linked since all v_i are distinct. $\Box$

Definition 7.6. Let L be a semi-cubical complex. Let h be a dimap from $\mid S_k \mid$ to $\mid L \mid$. h satisfies the star condition if,

- for all $i = 0, \ldots, k$, there exists $w_i \in L_0$ such that $h(\mathrm{St}v_i) \subseteq \mathrm{St}w_i$,
- for all $i = 0, \ldots, k-1$, $w_i = w_{i+1}$ or there exists $z_i \in L_1$ with $\partial_0^0(z_i) = w_i$ and $\partial_0^1(z_i) = w_{i+1}$.

Remark 7.7. The usual star condition is not enough for semi-cubical complexes. Look at Figure 24: we have pictured a non self-linked cubical complex M, $h(v_0)$ to $h(v_3)$ where $h : S_3 \rightarrow M$ and corresponding w_i (when we forget about the second requirement), on the left, and the same with a finer subdivision, with $h : S_4 \rightarrow M$. We can see that the w_i on the left do not give rise to a convenient cubical approximation.

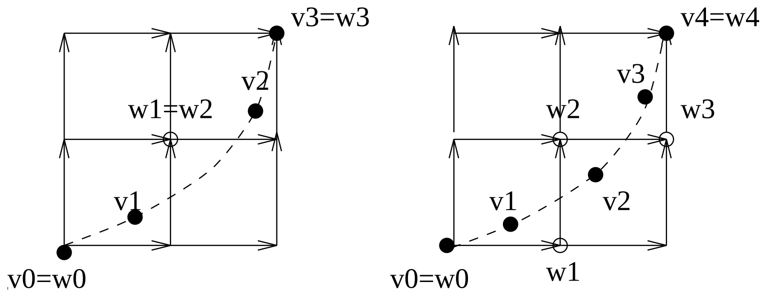

FIGURE 24. The "cubical star condition" illustrated.

Lemma 7.8. *Let $h :| S_k |\rightarrow L$ (L any semi-cubical complex). Then if h satisfies the star condition there h admits a semi-cubical approximation.*

Proof. Let $f : S_k \rightarrow L$ with,

- $f(v_i) = w_i$ (w_i is given by Definition 7.6),
- $f(u_i) = z_i$ (z_i is given by Definition 7.6).

Then f is obviously a map of semi-cubical complexes and is a semi-cubical approximation of h. □

Theorem 7.9. *Let L be a finite semi-cubical complex and h be a dipath in $\mid L \mid$ (i.e. a dimap from $\square_1$ to $\mid L \mid$). Then there exists a cubical approximation $f : S_k \rightarrow L$ of h (seen also as a dimap from $\mid S_k \mid \rightarrow \mid L \mid$ since $\mid S_k \mid$ is dihomeomorphic to $\square_1$). Moreover, f defines a (combinatorial) dipath $(f(u_1), \ldots, f(u_k))$ which we denote by $\tilde{f}$ and that we call the semi-cubical approximation of dipath h and, $\mid f \mid$ is homeomorphic to $\mid \tilde{f} \mid$,*

Proof. By Lemma 7.8 we only need to show that there exists a subdivision S_k of $D_{[1]}$ such that h seen as a dimap from $\mid S_k \mid$ to L satisfies the star condition.

Cover now $\mid L \mid$ by opens in A which are intersections of elements of $\{h^{-1}(\mathrm{St}(w, L)) \mid w \in L_0\}$ and of $\{h^{-1}(\mathrm{St}(a, L)) \mid a \in L_1\}$. $\mid L \mid$ is compact metric since L is a finite complex (whose realization is included into some $\mathbb{R}^n$). Therefore there exists $\lambda > 0$ (the Lebesgue number) such that any set of diameter less than λ lies in one of the elements of A.

Let us consider the d_∞ metric on $\mathbb{R}^n$. Then the diameter of a singular n-cube c is diam $c = max_{(x,y)\in c^2} d_\infty(x, y)$ and we have diam $(c \circ s_{b_1,\cdots,b_n}) = \frac{\text{diam } c}{2}$. Therefore, given $\epsilon > 0$, $\exists N$ such that $Sd^N D_{[1]}$ has all its n-cubes of diameter less than ϵ. Choose $\epsilon = \frac{\lambda}{2}$, and $Sd^N D_{[1]}$ is a subdivision S_k for some k. Then each star of a vertex in S_k has diameter strictly less that λ so is included in some $h^{-1}(\mathrm{St}(w, M))$ and also $\partial_0^0(h(u_i))$ is equal to $\partial_0^1(h(u_i))$ or there is a segment $a_i \in L_1$ such that these two points are boundaries of a_i (for all i in $1, \cdots, k$). This entails that $h :| S_k |\rightarrow| L |$ satisfies the star condition.

□

This entails that reasoning combinatorially on a cubical complex or geometrically on its topological realization is equivalent when it comes to dihomotopy. In particular, in dimension 2, this means that local commutation rules are the same as dihomotopy. This also makes a link with [27].

8. 2PHASE LOCKING IS SAFE; A MODIFICATION OF J. GUNAWARDENA'S PROOF

8.1. **Introduction.** This section gives an example of the applicability of our general approach. We obtain a new conceptual proof of safety for the "*2-phase-locking*" strategy in scheduling problems in data engineering from the study of specific dimaps and "dicontractions" : this scheduling strategy ensures that a concurrent program has the same effect as a serial execution of the individual programs as explained in the introduction. Consider several shared objects of memory that only a restricted number of processors can read and update concurrently (mutual exclusion, a generalisation of semaphore programs, cf. [15]). To avoid a situation where more processes work on one such object than allowable (at a given time), the transactions have to acquire locks to any object before working on it. The 2-phase locking strategy requires that every transaction must aquire all its locks before relinquishing any. In [35], J. Gunawardena gave geometric arguments (using dipaths and homotopies between them) to show that any execution in a 2-phase locked schedule is serializable. In a geometric language, this corresponds to the fact that any dipath in the associated "process graph" [15] is homotopic to a dipath on the 1-skeleton of the boundary of that model. See J. Gunawardena's very nice paper [35] explaining the connection with data engineering background and Dijkstra's process graph [15] in detail.

The aim of this section is twofold. First of all, we want to give a modification of Gunawardena's reasoning in the more general framework of the present paper. Our proof is certainly more technical, but it seems to have several advantages: First of all, we avoid J. Gunawardena's "wobbling" problems (cf. [35], p. 189): in his construction, he has to consider intermediate paths that are *not* dipaths – and to replace them by such. Secondly, our proof does not only work in the case of semaphore programs, but for general "mutual exclusion" programs – a fixed number $a \geq 1$ of transactions can acquire a lock to the same shared object at the same time. Finally, we hope that this proof can be a prototype of a more general van Kampen theorem for calculating π_1 of a dispace from the difundamental monoids of suitably chosen subspaces (compare with e.g. [33] in ordinary topology).

8.2. **Blockwise starshaped sets.** First, we modify the concept of a "star-shaped" set in a vector space (used in [35]) in the presence of a partial order:

Definition 8.1.
1. For $x, c \in \mathbf{R}$ let $I(x,c)$ denote the interval $[x,c] \cup [c,x]$.
2. For $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{c} = (c_1, \ldots, c_n) \in \mathbf{R}^n$ let $I(\mathbf{x},\mathbf{c}) = \prod I(x_i, c_i)$, cf. Fig. 25.
3. Let $\mathbf{c} \in F \subset \mathbf{R}^n$. The set F is called *blockwise starshaped with respect to* $\mathbf{c}$ if and only if $I(\mathbf{x},\mathbf{c}) \subset F$ for every $\mathbf{x} \in F$.

Since the block $I(\mathbf{x},\mathbf{c})$ is convex, a set that is blockwise starshaped with respect to $\mathbf{c}$ is also starshaped with respect to $\mathbf{c}$ in the classical sense.

Example 8.2.
1. An n-cube R (n-rectangle in [19]) is blockwise starshaped with respect to every point in R. A Euclidean ball is blockwise starshaped with respect to its center.
2. A *union* F of n-cubes is starshaped with respect to every point in their intersection. The forbidden region in a process graph is modelled by such a union of n-cubes. It has a nonempty "central" intersection if it is a model of a 2-phase locked transaction system.
3. The triangle $T = \{(x_1, x_2) \in \mathbf{R}^2 | x_i \geq 0, x_1 + x_2 \leq 1\}$ is starshaped with respect to every of its points. It is *not* blockwise starshaped with respect to any point $(c_1, c_2) \in T$ apart from $(c_1, c_2) = (0,0)$.

What are the properties of *complements* of blockwise starshaped sets? Let $I = [a,b]$, $F \subset I^n$ and $X = I^n \setminus F$.

Definition 8.3.
1. For $x, c \in I = [a,b]$ define the interval $J(x,c) \subset I$ by

$$J(x,c) = \begin{cases} [a,x], & x < c \\ [x,b], & x > c \\ [a,b], & x = c. \end{cases}$$

2. For $\mathbf{x} = [x_1, \ldots, x_n], \mathbf{c} = [c_1, \ldots, c_n] \in \mathbf{R}^n$ let $J(\mathbf{x},\mathbf{c}) = \prod J(x_i, c_i)$, cf. Fig. 25.

Lemma 8.4. *Let $F \subset I^n = [a,b]^n$ be blockwise starshaped with respect to $\mathbf{c} \in F$. Let $X = I^n \setminus F$. Then $J(\mathbf{y},\mathbf{c}) \subset X$ for every $\mathbf{y} \in X$.*

Proof. Assume $\mathbf{x} \in J(\mathbf{y},\mathbf{c}) \cap F$. Then $\mathbf{y} \in I(\mathbf{x},\mathbf{c}) \subset F$. Contradiction! □

8.3. **Partitions and Contractions.** Suppose $\mathbf{c} \in \overset{\circ}{F} \subset I^n$ and F blockwise starshaped with respect to $\mathbf{c}$. We want to study dipaths in $X = I^n \setminus F$ from $\mathbf{a} = (a, \ldots, a)$ to $\mathbf{b} = (b, \ldots, b)$. In order to get formulas that are easy to verify and to overlook, we apply a dihomeomorphism $\Psi : [a,b]^n \to [-1,1]^n$ with $\Psi(\mathbf{c}) = \mathbf{0}$. This dihomeomorphism should be chosen as a product of

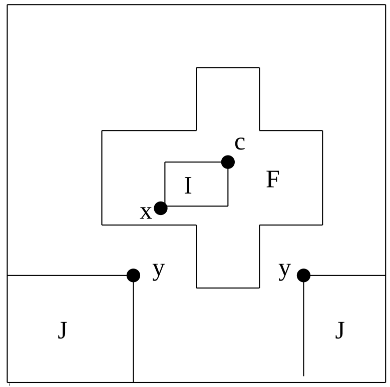

FIGURE 25. Sets F, $I(\mathbf{x},\mathbf{c})$, $I(\mathbf{y},\mathbf{c})$

maps that are increasing in each coordinate. This ensures that $\Psi(F)$ is blockwise starshaped with respect to $\mathbf{0}$. For $\mathbf{c}=\mathbf{0}$, we can describe the I- and J-set above as follows:

$$I(\mathbf{x},\mathbf{0}) = \{(y_1,\ldots,y_n)\in I^n | sgn(x_i) = sgn(y_i) = sgn(x_i - y_i)\};$$

$$J(\mathbf{x},\mathbf{0}) = \{(y_1,\ldots,y_n)\in I^n | x_i \neq 0 \Rightarrow sgn(x_i) = sgn(y_i) = sgn(y_i - x_i)\}.$$

From now on, let $I=[-1,1]$. We study the classification of dipaths using a subdivision of I^n with respect to $\mathbf{0}$. The decomposition of the interval $I^i = I^i_{-1}\cup I^i_1; = [-1,0]\cup[0,1]$ induces a decomposition of I^n into 2^n sub-n-rectangles $I_\Delta = I_{\delta_1\ldots\delta_n} = \prod I^i_{\delta_i}$. There is an obvious partial order between those subrectangles giving rise to $n!$ directed paths from $I_{-1\cdots-1}$ to $I_{1\ldots1}$. We need the following subsets of I_Δ:

Definition 8.5. Given $\Delta\in\{-1,1\}^n$ with $\delta_i = 1$ and δ_j= -1. Then

1. $I^{i,j}_\Delta = \{\mathbf{x}\in I_\Delta | (x_i,x_j)\neq(0,0)\}$.
2. $I^{i,j}_{\Delta0} = \{\mathbf{x}\in I^{i,j}_\Delta | x_i = 0\}$, $I^{i,j}_{\Delta1} = \{\mathbf{x}\in I^{i,j}_\Delta | x_j = 0\}$.
3. $J^{i,j}_\Delta = \{\mathbf{x}\in I_\Delta | x_k = \delta_k,\ k\neq i,j;\ x_j = -1 \text{ or } x_i = 1\}$.
4. The latter is a 1-complex with endpoints $\mathbf{p}^{i,j}_{\Delta0}$ and $\mathbf{p}^{i,j}_{\Delta1}$ with $(\mathbf{p}^{i,j}_{\Delta*})_k = \delta_k,\ k\neq i,j,\ (\mathbf{p}^{i,j}_{\Delta0})_i = 0,\ (\mathbf{p}^{i,j}_{\Delta0})_j = -1,; (\mathbf{p}^{i,j}_{\Delta1})_i = 1,\ (\mathbf{p}^{i,j}_{\Delta1})_j = 0.$

In fact, $I^{i,j}_\Delta$ is one of the 2^n sub-n-rectangles mentioned above with a 2-codimensional face removed; $I^{i,j}_{\Delta0}$ and $I^{i,j}_{\Delta1}$ represent two of its faces; $J^{i,j}_\Delta$ represents a (totally ordered) 1 complex in its boundary from $\mathbf{p}^{i,j}_{\Delta0}$ to $\mathbf{p}^{i,j}_{\Delta1}$.

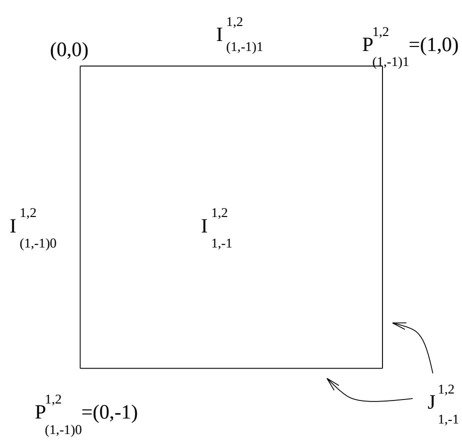

FIGURE 26. Subrectangle with faces

Next comes a definition of a *dimap*

$$\Phi^{\Delta,i,j} = (\Phi_1^{\Delta,i,j}, \dots \Phi_n^{\Delta,i,j}) : (I_\Delta^{i,j}; I_{\Delta 0}^{i,j}, I_{\Delta 1}^{i,j}) \to (J_\Delta^{i,j}, \mathbf{p}_{\Delta 0}^{i,j}, \mathbf{p}_{\Delta 1}^{i,j}) :$$

Definition 8.6. 1. $\Phi_k^{\Delta,i,j}(\mathbf{x}) = \delta_k$ for $k \neq i, j$.
2. $\Phi_i^{\Delta,i,j}(\mathbf{x}) = \frac{x_i}{-x_j}$ and $\Phi_j^{\Delta,i,j}(\mathbf{x}) = -1$ for $x_i \leq -x_j$.
3. $\Phi_i^{\Delta,i,j}(\mathbf{x}) = 1$ and $\Phi_j^{\Delta,i,j}(\mathbf{x}) = \frac{x_j}{x_i}$ for $x_i \geq -x_j$.

There is a directed version of deformation retracts (cf.[17]) in the dispace world, too:

Definition 8.7. Let $A \subset X$ denote an inclusion of two di-spaces. The subspace A is called a *strong deformation di-retract* of X if there exists a dimap $\Phi : X \to A$ restricting to the identity on A and a dihomotopy $H : X \times I \to X$ between Φ and the identity map on X which restricts to the trivial homotopy on A: $H_t|A = id,\ t \in I$.

Using the map $\Phi^{\Delta,i,j}$ above, we can then show:

Proposition 8.8. 1. *The subcomplex $J_\Delta^{i,j}$ is a strong deformation di-retract of $I_\Delta^{i,j}$. More precisely, $\Phi^{\Delta,i,j}$ is a dimap extending the identity on $J_\Delta^{i,j}$. It is dihomotopic to the identity map on $I_\Delta^{i,j}$ via a dihomotopy that fixes $J_\Delta^{i,j}$ pointwise.*
2. *If $\mathbf{0} \in \overset{\circ}{F}$ and F is blockwise starshaped with respect to $\mathbf{0}$, then $\Phi^{\Delta,i,j}(\mathbf{x}) \in J(\mathbf{x},\mathbf{c}) \subset X = I^n \backslash F$ for every $\mathbf{x} \in X$. Furthermore, $J_\Delta^{i,j} \setminus F$ is a strong deformation di-retract of $I_\Delta^{i,j} \setminus F$.*
3. *For $\Delta = (\delta_1, \dots, \delta_n)$ as above and $\Delta' = (\delta_1', \dots, \delta_n')$ with $\delta_k' = \delta_k$ for $k \neq j$, $\delta_j' = 1$, $\delta_l = \delta_l' = -1$, the maps $\Phi^{\Delta,i,j}$ and $\Phi^{\Delta',j,l}$ agree on the intersection of their domains of definition. The same is true for the homotopies from 2. above.*

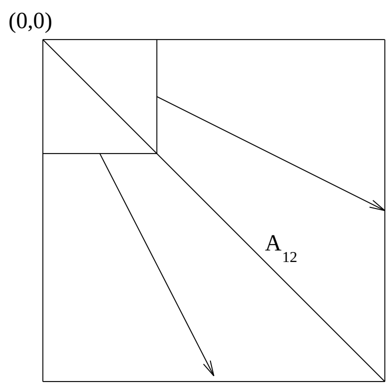

FIGURE 27. The dimap $\Phi^{\Delta,i,j}$

Proof.

1. The dimap property depends only on the (x_i, x_j)-coordinates. In that projection, the map $\Phi^{\Delta,i,j}$ “stretches” every wedge in Fig. 27 out on the boundary. In particular, points under/over the “antidiagonal” $\{x_i = -x_j\}$ are mapped to points in subsequent 1-simplices. It is elementary to see that $\Phi^{\Delta,i,j}$ is a dimap “under”, resp. ”over” that antidiagonal.
   The self-dihomotopy $H^{i,j}$ on $I_\Delta^{i,j}$: given by $H^{i,j}(\mathbf{x},t) = (1-t)\mathbf{x} + t\Phi^{\Delta,i,j}(x)$ connects $\mathbf{x}$ to $\Phi^{\Delta,i,j}(x)$ linearly; in particular, all of the maps H_t are di-maps.
2. We have to show that the dihomotopy $\Phi^{\Delta,i,j}$ restricts to a self-dihomotopy on $I_\Delta^{i,j} \setminus F$. By definition, $\Phi_k^{\Delta,i,j}(\mathbf{x}) \geq x_k$ for $\delta_k = 1$ and $\Phi_k^{\Delta,i,j}(\mathbf{x}) \leq x_k$ for $\delta_k = -1$. Similarly for H_t.
3. On the intersection $I_{\Delta 1}^{i,j} \cap I_{\Delta 0}^{j,k}$, the maps $\Phi^{\Delta,i,j}$ and $\Phi^{\Delta',j,l}$ are *constant* with value $\mathbf{p}_{\Delta 1}^{i,j} = \mathbf{p}_{\Delta 0}^{j,l}$.

□

We need special care for the minimal and maximal subrectangles I_{Δ_-}, resp. I_{Δ_+} corresponding to $\Delta_- = (-1,\dots,-1)$ and $\Delta_+ = (1,\dots,1)$. Let $I^i_{\Delta_\pm} = \{\mathbf{x} \in I_{\Delta_\pm} | x_k = \pm 1,\ k \neq i;\ x_i = 0\}$, $J^i_{\Delta_\pm} = \{\mathbf{x} \in I_{\Delta_\pm} | x_k = \pm 1,\ k \neq i\}$, and $(\mathbf{p}^i_{\Delta_\pm})_k = \pm 1,\ k \neq i,\ (\mathbf{p}^i_{\Delta_\pm})_i = 0$. Then, we define

$$\Phi^{\Delta_\pm, i} : (I_{\Delta_\pm}; I^i_{\Delta_\pm}) \to (J^i_{\Delta_\pm}, \mathbf{p}^i_{\Delta_\pm}), \quad \Phi^{\Delta_\pm, i}(\mathbf{x}) = \begin{cases} \pm 1, & k \neq i, \\ x_i, & k = i. \end{cases}$$

Proposition 8.9. *The analogue of Prop. 8.8 holds in these cases as well.* □

To formulate the corollary, we need notation about sequences of and unions of the subrectangles I_Δ above: Let $\sigma \in \Sigma_n$ denote a permutation of the integers $\{1,\dots,n\}$. Let $\Delta_\sigma(k) = (\delta_1,\dots,\delta_n)$ be given by $\delta_i = 1$ if $i \in \{\sigma(1),\dots\sigma(k)\}$ and $\delta_i = -1$ otherwise. Then $\Delta_\sigma(0),\dots,\Delta_\sigma(n)$ is an ascending chain of sub-rectangles from Δ_- to Δ_+. Let

$$I_\sigma = I_{\Delta-} \cup \bigcup_{k=1}^{n-1} I^{\sigma(k),\sigma(k+1)}_{\Delta_\sigma(k)} \cup I_{\Delta+},\ J_\sigma = J^{\sigma(1)}_{\Delta-} \cup \bigcup_{k=1}^{n-1} J^{\sigma(k),\sigma(k+1)}_{\Delta_\sigma(k)} \cup J^{\sigma(n)}_{\Delta+}$$

denote a union of (n+1)-subrectangles (without certain 2-dimensional subsets) in an ascending chain, resp. a *totally ordered* 1-dimensional subcomplex in the boundary. Glueing the dimaps and dihomotopies from Prop. 8.8 and Prop. 8.9 together, we obtain

Corollary 8.10. 1. *The 1-complex J_σ is a strong deformation di-retract of the complex I_σ.*

2. *If $\mathbf{0} \in \mathring{F}$ and F is blockwise starshaped with respect to $\mathbf{0}$, then J_σ is a strong deformation di-retract of the complex $I_\sigma \setminus F$.*

Remark 8.11. A dispace with a 1-subcomplex as a strong deformation di-retract is the analogue to a *contractable* space in ordinary topology. Hence, one might call such a dispace dicontractable. The strategy of the proof was thus to subdivide the underlying dispace into dicontractable pieces with control on the intersection. We hope that this strategy can be generalised to an adaption of van Kampen's theorem (see [33]) to the di-space category.

8.4. **Application to dipaths and serializability.** As an application, we obtain the following result about dipaths generalizing J. Gunawardena's result. Remark that the forbidden region in a process graph is blockwise starshaped with respect to a central point (see Ex. 8.2).

Theorem 8.12. *Let $\mathbf{0} \in \mathring{F}$ and F be blockwise starshaped with respect to $\mathbf{0}$. Every dipath in $X = I^n \setminus F$ from $-1 = (-1,\dots,-1)$ to $\mathbf{1} = (1,\dots,1)$ is dihomotopic to a dipath on the 1-skeleton $(\partial I^n)_1$ of the boundary $\partial I^n = \{(x_1,\dots,x_n) | \exists k : x_i = \pm 1,\ i \neq k\}$ of I^n.*

Proof. Let $\Sigma = \{(x_1,\dots x_n) \in I^n |\ x_i = x_j = 0 \text{ for } 1 \leq i < j \leq n\} \subset I^n$ denote the "singular set". Every dipath in $I^n \setminus F$ *avoiding* Σ is contained in one of the complexes I_σ for a permutation $\sigma \in \Sigma_n$ and thus dihomotopic in $X = I^n \setminus F$ to a dipath in $J_\sigma \subset (\partial I^n)_1$ by Cor. 8.10.

How can one handle dipaths intersecting Σ? We can apply a (locally linear) transversality argument to see that every dipath in X is dihomotopic to one avoiding Σ. Alternatively, We may give X the structure of a cubical complex in such a way that no 1-cube intersects Σ, and then argue that every dipath is dihomotopic to a dipath on the 1-skeleton of X. If X is the "forbidden region" corresponding to "mutual exclusion" in a process graph (cf.[35, 19]), the subdivision will have that non-intersection property by construction. □

9. Open Mathematical Problems

We tried in this paper to state only the definitions and theorems that should be the basis for the theory of directed homotopy. We believe that the following properties should be investigated more closely,

(M1) There is a natural order on the set of diconnected components, induced by the local partial order. How does this graph relate to the dihomotopy classes of dipaths? We believe that this gives the whole information about these dihomotopy classes.

(M2) What are the dihomotopy classes of dipaths of the union X (not necessarily disjoint!) of two locally po-spaces A and B when we know the dihomotopy classes of dipaths in A and in B? To solve this problem, we obviously need to have some information about how A and B are glued together. What we really want is an analogous of the Seifert/Van Kampen theorem of the usual homotopy theory. But we believe that the necessary glueing information is not in the form of the usual function induced by the homotopy functors by the inclusion morphisms of A and B into X, but would rather be in the form of the functions induced by the "dihomotopy functors" by some kind of restriction morphisms from X to A and B.

(M3) What is the convenient category of locally po-spaces? We would like to have cartesian closedness, so that we can define the higher-order homotopies in a simple inductive way.

(M4) What would be the structure on these higher-order homotopy sets then?

(M5) One of the aims of this theory is to study locally po-spaces up to dihomotopy equivalence. What would be the counterpart of the classification theorem that we have for surfaces in ordinary homotopy theory? We believe that the usual notion of orientability does not play a role. For instance, the Moebius strip can be given the structure of a locally po-space. The projective plane can also be given such structure (see 6.25). Also, looking at Figure 14 we see that the usual classification is refined in some ways by considering dihomotopy.

(M6) What is the relationship between dihomotopy and the homology theory defined in [27], constructed from a bicomplex naturally arising from cubical sets? We believe that this homology theory is an invariant for dihomotopy indeed but that it is not characterizing dihomotopy exactly. For instance the two dipaths in Figure 18 are (di-) homologous but not dihomotopic.

(M7) How could symmetry group actions on a locally po-space inform us about the possible dihomotopy classes of dipaths?

10. Open Computer-Scientific Problems

We tried in this paper to motivate the mathematics by some examples and concepts taken from several areas of computer science. Some new applications, or some new results could be derived from this theory in the following sense,

(CS1) How can we exploit (M1) so that we can derive the "essential schedules" of a concurrent systems? A nice application has already been made for a small subset of the set of diconnected components, namely the unsafe region and the unreachable region in simple cases, see [19]. This would be important since these schedules describe fine (safety) properties of concurrent systems (about the possible orderings on accesses to shared ressources for instance) that for instance encompass serializability issues. (M2) would make possible the difficult problem of reasoning about schedules of a system compositionally, i.e. inductively on the knowledge of its subparts. This would be of a great algorithmic value for program analysis for instance.

(CS2) (M3) and (M4) would make it possible to consider more refined properties of fault-tolerant systems and make the complete link with M. Herlihy, S. Rajsbaum and N. Shavit theories. Basically the aim is to give the semantic foundations to a computability and complexity theory for fault-tolerant distributed systems. (M5) would help understand what are the basic fundamental synchronisation models that one can imagine for concurrent systems.

(CS3) (M6) would make it possible to have good algorithms (from linear algebra) giving some information about schedules of concurrent programs. This was already hinted in [28].

(CS4) The whole theory should give some better invariants for the problem of knowing which monoids can be presented by finite canonical term rewriting systems [58]. More generally, we could ask ourself what can be computed in more general structures [47]. This is very much linked to (CS2).

(CS5) We think that better algorithms could be designed for distributed databases schedulers (like better "path-pushing" algorithms) and for micro-instructions schedulers using ideas from this theory. (M7) would also help simplify these algorithms in some specific cases.

11. Related work

Slightly different geometric models have been used in the work of people like M. Herlihy, N. Shavit and S. Rajsbaum on fault-tolerant protocols for distributed systems. This already gave numerous results in the field of distributed systems. It is proved for instance in [40] that in a shared-memory model with single reader/sin–gle writer registers providing atomic read and write operations, k-set agreement requires at least $\lfloor f/k \rfloor + 1$ rounds where f is the number of processes that can fail. General tests are also given for solving t-resilient problems. Not only impossibility results can be given but also constructive means for finding algorithms derive from this work (see for instance [41]). We refer the reader to other articles in this area, in particular, the book on distributed algorithms [46], other articles by Herlihy et al. [37], [38], [39], some slightly different methods, but still geometric in nature, in [2], [5], [6], [11], [22] (which originated this field of research, starting with graph theoretical arguments), [51]. Also some ideas about classifying data structures according to what protocols they manage to solve are described in [42], [43], [53]. This should be related to problem (CS2). Some links with directed homotopy have been hinted by one of the co-authors [29], [30], [31].

In concurrent databases, we refer the reader to the very nice introductory paper of J .Gunawardena [35]. An application of this theory to the problem of deadlock detection has been made by the authors in [19].

On the semantic side, there is a strong link between po-spaces and progress graphs [10], [16]. Then the model presented in [49], [62] originated part of this work, and most of the work of one of the co-authors, [26], [27], [32] and also [45]. Also there are links with the homological considerations of [23] and [24]. Some potentially related semantic models are the n-categorical formulations of [9] and some other combinatorial or categorical formulations in [21], [52], [55], [56], [57].

In program analysis, some proposals have been made to use the scheduling information that one can extract from the geometry of executions, to derive automatic parallelization algorithms. This has been hinted in [28] (where some other ideas for program analysis are exemplified) and also fully treated for CCS in [60], [61]. A prototype Parallel Pascal Analyser has been implemented by R. Cridlig (http://www.dmi.ens.fr/~cridlig) and its principles described in [12] . Another application on CML has been designed by the same author in [13] and to Linda-based languages in [14].

On the more mathematical side, homology of monoids has been studied in [20], [34] (with the specific use of cubical complexes), [44], [58]. An extension to homology of categories has been proposed in [47]. Also cubical complexes as such have been used in [54] and studied combinatorially and categorically in [7], [8].

References

1. A. Arnold, *Systèmes de transitions finis et sémantique des processus communicants*, Masson, 1992.
2. H. Attiya, A. Bar-Noy, D. Dolev, D. Peleg, and R. Reischuk, *Renaming in an asynchronous environment*, Journal of the ACM **37** (1990), no. 3, 524–548.
3. H. Attiya and R. Friedman, *A correctness condition for high-performance multiprocessors*, Proc. of the 24th STOC, ACM Press, 1992.
4. M. A. Bednarczyk, *Categories of asynchronous systems*, Ph.D. thesis, University of Sussex, 1988.
5. E. Borowsky, *Capturing the power of resiliency and set consensus in distributed systems*, Tech. report, University of California in Los Angeles, 1995.
6. E. Borowsky and E. Gafni, *Generalized FLP impossibility result for t-resilient asynchronous computations*, Proc. of the 25th STOC, ACM Press, 1993.
7. R. Brown and P. J. Higgins, *Colimit theorems for relative homotopy groups*, Journal of Pure and Applied Algebra (1981), no. 22, 11–41.
8. ———, *On the algebra of cubes*, Journal of Pure and Applied Algebra (1981), no. 21, 233–260.
9. R. Buckland and M. Johnson, *ECHIDNA: A system for manipulating explicit choice higher dimensional automata*, AMAST'96: Fifth Int. Conf. on Algebraic Methodology and Software Technology (Munich), 1996.
10. S. D. Carson and P. F. Reynolds Jr, *The geometry of semaphore programs*, ACM Transactions on Programming Languages and Systems **9** (1987), no. 1, 25–53.
11. S. Chaudhuri, *Agreement is harder than consensus: set consensus problems in totally asynchronous systems*, Proc. of the 9th Annual ACM Symposium on Principles of Distributed Computing, ACM Press, August 1990, pp. 311–334.

12. R. Cridlig, *Semantic analysis of shared-memory concurrent languages using abstract model-checking*, Proc. of PEPM'95 (La Jolla), ACM Press, June 1995.
13. ______, *Semantic analysis of Concurrent ML by abstract model-checking*, Proceedings of the LOMAPS Workshop, 1996.
14. R. Cridlig and E. Goubault, *Semantics and analyses of Linda-based languages*, Proc. of WSA'93, LNCS, no. 724, Springer-Verlag, 1993.
15. E.W. Dijkstra, *Co-operating sequential processes*, Programming Languages (F. Genuys, ed.), Academic Press, New York, 1968, pp. 43–110.
16. ______, *Cooperating sequential processes*, Academic Press, 1968.
17. A. Dold, *Lectures on algebraic topology*, 2 ed., Grundlehren der matematischen Wissenschaften, vol. 200, Springer-Verlag, 1980.
18. M. Evrard, *Homotopie des complexes simpliciaux et cubiques*, Tech. report, Université Paris 6.
19. L. Fajstrup, E. Goubault, and M. Raussen, *Detecting Deadlocks in Concurrent Systems*, CONCUR '98; Concurrency Theory (Nice, France) (D. Sangiorgi and R. de Simone, eds.), Lect. Notes Comp. Science, vol. 1466, Springer-Verlag, September 1998, 9th Int. Conf., Proceedings, pp. 332 – 347.
20. D. R. Farkas, *The Anick resolution*, Journal of Pure and Applied Algebra **79** (1992).
21. Marcelo Fiore, Gordon Plotkin, and John Power, *Complete cuboidal sets in axiomatic domain theory (extended abstract)*, Proceedings, Twelth Annual IEEE Symposium on Logic in Computer Science (Warsaw, Poland), IEEE Computer Society Press, 29 June–2 July 1997, pp. 268–279.
22. M. Fisher, N. A. Lynch, and M. S. Paterson, *Impossibility of distributed commit with one faulty process*, Journal of the ACM **32** (1985), no. 2, 374–382.
23. P. Gaucher, *Connexion de flux d'information en algèbre homologique*, Tech. report, IRMA, Strasbourg, available at http://irmasrv1.u-strasbg.fr/~gaucher/activite.html, 1997.
24. ______, *Itude homologique des chemins de dimension 1 d'un automate*, Tech. report, IRMA, Strasbourg, available at http://irmasrv1.u-strasbg.fr/~gaucher/activite. html, 1997.
25. G.Gierz, K.H.Hofmann, K. Keimel, J.D.Lawson, M.Mislove, and D.S.Scott, *A Compendium of Continuous Lattices*, Springer-Verlag, Berlin Heidelberg New York, 1980.
26. E. Goubault, *Domains of higher-dimensional automata*, Proc. of CONCUR'93 (Hildesheim), Springer-Verlag, August 1993.
27. ______, *The geometry of concurrency*, Ph.D. thesis, Ecole Normale Supérieure, 1995, to be published, 1998, also available at http://www.dmi.ens.fr/~goubault.
28. ______, *Schedulers as abstract interpretations of HDA*, Proc. of PEPM'95 (La Jolla), ACM Press, also available at http://www.dmi.ens.fr/~goubault, June 1995.
29. ______, *The dynamics of wait-free distributed computations*, Tech. report, Research Report LIENS-96-26, December 1996.
30. ______, *A semantic view on distributed computability and complexity*, Proceedings of the 3rd Theory and Formal Methods Section Workshop, Imperial College Press, also available at http://www.dmi.ens.fr/~goubault, 1996.
31. ______, *Optimal implementation of wait-free binary relations*, Proceedings of the 22nd CAAP, Springer Verlag, 1997.
32. E. Goubault and T. P. Jensen, *Homology of higher-dimensional automata*, Proc. of CONCUR'92 (Stonybrook, New York), Springer-Verlag, August 1992.
33. B. Gray, *Homotopy theory*, Pure and Applied Mathematics, vol. 64, Academic Press, New York, 1975.
34. J. R. J. Groves, *Rewriting systems and homology of groups*, Groups – Canberra 1989 (L. G. Kovacs, ed.), no. 1456, Lecture notes in Mathematics, Springer-Verlag, 1991, pp. 114–141.
35. J. Gunawardena, *Homotopy and concurrency*, Bulletin of the EATCS **54** (1994), 184–193.
36. E. Harcourt, J. Mauney, and T. Cook, *From processor timing specifications to static instruction scheduling*, Proc. of the Static Analysis Symposium'94, LNCS, Springer-Verlag, 1994.
37. M. Herlihy, *A Tutorial on Algebraic Topology and Distributed Computation*, Tech. report, presented at UCLA, 1994.
38. M. Herlihy and S. Rajsbaum, *Set consensus using arbitrary objects*, Proc. of the 13th Annual ACM Symposium on Principles of Distributed Computing, ACM Press, August 1994.
39. ______, *Algebraic topology and distributed computing, a primer*, Tech. report, Brown University, 1995.
40. M. Herlihy and N. Shavit, *The asynchronous computability theorem for t-resilient tasks*, Proc. of the 25th STOC, ACM Press, 1993.
41. ______, *A simple constructive computability theorem for wait-free computation*, Proceedings of STOC'94, ACM Press, 1994.
42. Prasad Jayanti, *On the robustness of Herlihy's hierarchy*, Proceedings of the Twelth Annual ACM Symposium on Principles of Distributed Computing (Ithaca, New York, USA), 15–18 August 1993, pp. 145–157.
43. ______, *Robust wait-free hierarchies*, Journal of the ACM **44** (1997), no. 4, 592–614.
44. Y. Kobayashi, *Complete rewriting systems and homology of monoid algebras*, Journal of Pure and Applied Algebra **65** (1990), 263–275.
45. E. Lanzmann, *Automates d'ordre supérieur*, Master's thesis, Université d'Orsay, 1993.
46. N. Lynch, *Distributed algorithms*, Morgan-Kaufmann, 1996.

47. F. Morace, *Finitely presented categories and homology*, Tech. report, Université Joseph Fourier, 1995.
48. R. Penrose, *Techniques of Differential Topology in Relativity*, Conference Board of the Mathematical Sciences, Regional Conference Series in Applied Mathematics, vol. 7, SIAM, Philadelphia, USA, 1972.
49. V. Pratt, *Modeling concurrency with geometry*, Proc. of the 18th ACM Symposium on Principles of Programming Languages, ACM Press, 1991.
50. T. A. Proebsting and C. W. Fraser, *Detecting pipeline structural hazards quickly*, Proc. of the Symposium on Principles of Programming Languages, ACM Press, 1994.
51. M. Saks and F. Zaharoglou, *Wait-free k-set agreement is impossible: The topology of public knowledge*, Proc. of the 25th STOC, ACM Press, 1993.
52. V. Sassone and G. L. Cattani, *Higher-dimensional transition systems*, Proceedings of LICS'96, 1996.
53. Eric Schenk, *The consensus hierarchy is not robust*, Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing (Santa Barbara, California), 21–24 August 1997, p. 279.
54. J.P. Serre, *Homologie singulière des espaces fibrés. applications*, Ph.D. thesis, Ecole Normale Supérieure, 1951.
55. S. Sokolowski, *Homotopy in concurrent processes*, Tech. report, Institute of Computer Science, Gdansk Division, 1998.
56. ———, *Investigation of concurrent processes by means of homotopy functors*, Tech. report, Institute of Computer Science, Gdansk Division, 1998.
57. ———, *Point glueing in cpo-s*, Tech. report, Institute of Computer Science, Gdansk Division, 1998.
58. C. C. Squier, F. Otto, and Y. Kobayashi, *A finiteness condition for rewriting systems*, Theoretical Computer Science **131** (1994), 271–294.
59. A. Stark, *Concurrent transition systems*, Theoretical Computer Science **64** (1989), 221–269.
60. Y. Takayama, *Cycle filling as parallelization with expansion law*, submitted to publication, 1995.
61. ———, *Extraction of concurrent processes from higher-dimensional automata*, Proceedings of CAAP'96, 1996, pp. 72–85.
62. R. van Glabbeek, *Bisimulation semantics for higher dimensional automata*, Tech. report, Stanford University, Manuscript available on the web as http://theory.stanford.edu/~rvg/hda, 1991.
63. R. van Glabbeek and U. Goltz, *Partial order semantics for refinement of actions*, Bulletin of the EATCS (1989), no. 34.
64. G. Winskel and M. Nielsen, Models for concurrency, vol. 3 of Handbook of Logic in Computer Science, 100–200, Oxford University Press, 1994, pp. 100–200.

Department of Mathematics, Institute for Electronic Systems, Aalborg University, Fredrik Bajers Vej 7E, DK - 9220 Aalborg Ø, and LETI (CEA - Technologies Avancées), DEIN-SLA, CEA F91191 Gif-sur-Yvette Cedex

E-mail address: {fajstrup,raussen}@math.auc.dk,goubault@aigle.saclay.cea.fr