Labelled cubical sets and asynchronous transition systems: an adjunction

Eric Goubault

LIST/DTSI/SLA, CEA Saclay, F-91191 Gif-sur-Yvette, France
Email: Eric.Goubault@cea.fr, Phone: +33 1 69 08 94 72, Fax: +33 1 69 08 83 95

Abstract. We show in this article that "labelled" cubical sets (or Higher-Dimensional Automata) are a natural generalization of transition systems and asynchronous transition systems. This generalizes an older result of [19] which was only holding with precubical sets and subcategories of the classical (see [37]) categories of transition systems and asynchronous transition systems. This opens up new promises on the actual use of geometric methods (such as [10]) and on comparisons with other methods for verification of concurrent programs.

keywords Models for concurrency, semantics, category theory.

1 Introduction

There is a great variety of models for concurrency, as witnessed in [37] for instance. Most of the relationships between these models are known, but the newer "geometric" models for concurrency, such as cubical sets (HDA in [30] or in [21]) or local po-spaces [11] have not been so well formally linked with older models, such as transition systems or transition systems with independence. In fact, cubical sets have a notion of generalized transition in their very definition. The idea of relating these in the style of G. Winskel et al. [37] with operational models for concurrency dates back to [19], but this was done only between fairly restricted categories. In this paper we greatly improve previous work by extending it to the full categories of transition systems (operational model of "interleaving" concurrency) and of transition systems with independence (operational model of "true" concurrency). The main idea is that by relating these models, we can compare the semantics of concurrent languages given in different formalisms. Moreover, it is hoped that specific methods for statically analysing concurrent programs (such as the deadlock detection algorithm of [10] in the case of cubical sets) in one model can be re-used in the other, giving some nice cross-fertilisations, some of these being hinted in Section 9.

This paper represents the first step towards formally linking "geometric" models with other models for concurrency. The links might appear as "intuitive", but the formal step we are making unravels interesting phenoma (besides being necessary for being able to relate semantics given in different styles) such as the fact that a certain category of labelled cubical sets forms an elementary topos, or such as the fact that persistent set types of methods for tackling the state-space explosion problem can be seen as searching for retracts of the state space, in the algebraic topological sense. We end this article by making some hypotheses on further relationships, with event structures and Petri nets in particular.

2 Transition systems

Transition systems are one of the oldest semantic models, both for sequential and concurrent systems. There is a convenient categorical treatment of this model, that we use in the sequel, taken from [37].

Definition 1. *A transition system is a structure (S,i,L,Tran) where,*

- *S is a set of states with initial state i*
- *L is a set of labels, and*
- *$Tran \subseteq S \times L \times S$ is the transition relation*

Transition systems are made into a category by defining morphisms to be some kind of simulation (for then being able to discuss about properties modulo [weak/strong] bisimulation, see [24]). The idea is that a transition system T_1 simulates a transition system T_0 if as soon as T_0 can fire some action a in some context, then T_1 can fire a as well in some related context. A morphism $f : T_0 \rightarrow T_1$ defines the way states and transitions of T_0 are related to states and transitions of T_1 making transition systems into a category TS.

Definition 2. *Let $T_0 = (S_0, i_0, L_0, Tran_0)$ and $T_1 = (S_1, i_1, L_1, Tran_1)$ be two transition systems. A partial morphism (or morphism in [37]) $f : T_0 \rightarrow T_1$ is a pair $f = (\sigma, \lambda)$ where,*

- *$\sigma : S_0 \rightarrow S_1$,*
- *$\lambda : L_0 \rightarrow L_1$ is a* partial *function. (σ, λ) are such that*
  - *$\sigma(i_0) = i_1$,*
  - *$(s, a, s') \in Tran_0$ and $\lambda(a)$ is defined implies $(\sigma(s), \lambda(a), \sigma(s')) \in Tran_1$. Otherwise, if $\lambda(a)$ is undefined then $\sigma(s) = \sigma(s')$.*

As in [37], we can restrict to "total morphisms" i.e. the ones for which λ is a total function by suitably completing transition systems. Just add "idle" transitions to transition systems, very similar in spirit to the lifting of domains in denotational semantics [23, 29], where partial functions from D to D are considered total (and strict) from $D_\perp$ to $D_\perp$ ($\perp$ is a new element such that $\forall x, \perp \leq x$). An idle (or "$\perp$-") transition is a transition $*$ such that $*$ goes from a state s to the same state s. Consider the following completion $T_* = (S_*, i_*, L_*, Tran_*)$ of a transition system $T = (S, i, L, Tran)$, by setting $S_* = S$, $i_* = i$, $L_* = L \cup \{*\}$ and $Tran_* = Tran \cup \{(s, *, s) \mid s \in S\}$. Now, a morphism $f = (\sigma, \lambda)$ (with λ a total function) from $(T_0)_*$ to $(T_1)_*$ such that $\lambda(*) = *$ is the same as a partial morphism f' from T_0 to T_1 by identifying $*$ with "undefined". Conversely, a partial morphism $f = (\sigma, \lambda)$ from T_0 to T_1 can be identified with $f_* = (\sigma, \lambda_*)$, $\lambda_*(x) = *$ if and only if $\lambda(x)$ is undefined.

3 Asynchronous Automata

Asynchronous Automata are a nice generalization of Mazurkiewicz traces, and have influenced a lot other models for concurrency (like transition systems with independence etc.). They have been independently introduced in [34] and [3]. The idea is to decorate transition systems with an "independence" relation (between actions) that will allow us to distinguish between true-concurrency and mutual exclusion (or non-determinism) of two actions. We actually use a slight modification for our purposes, due to [7], and called "automata with concurrency relations":

Definition 3. *An automaton with concurrency relations is a quintuple $(S, i, E, Tran, I)$ where,*

(1) S and E are disjoint sets; $i \in S$ is a distinguished element (the start state); $Tran$ is a subset of $S \times E \times S$,
(2) $Tran$ is such that whenever (s, e, s'), $(s, e, s'') \in Tran$, then $s = s''$; we require that for each $e \in E$, there are $s, s' \in S$ with $(s, e, s') \in Tran$;
(3) $I = (I_s)_{s \in S}$ is a family of irreflexive, symmetric binary relations I_s on E; it is required that whenever $e_1 I_s e_2$ $(e_1, e_2 \in E)$, there exist transitions (s, e_1, s_1), (s, e_2, s_2), (s_1, e_2, r) and (s_2, e_1, r) in $Tran$.

In the sequel, we relax condition (2). A morphism is now a morphism $f = (\sigma, \lambda)$ of the underlying transition systems such that $aI_s b$ implies $\lambda(a) I'_{\sigma(s)} \lambda(b)$ (when $\lambda(a)$ and $\lambda(b)$ are both defined). This makes automata with concurrency relations into a category, written ACR. The category of automata with concurrency relations over an alphabet E is named ACR_E.

Similarly to Section 2, we can equivalently consider ACR (and ATS) to be built using $*$ transitions and total morphisms. The condition on the independence relation is then $aI_s b \Rightarrow \lambda(a) I'_{\sigma(s)} \lambda(b)$ when $\lambda(a) \neq *$ and $\lambda(b) \neq *$.

4 Cubical sets

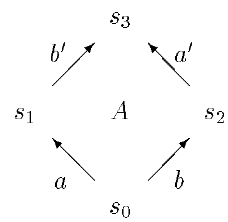
Cubical sets, which are classical objects in combinatorial algebraic topology, see for instance [33], have been used as an alternative "truly-concurrent" model for concurrency, in particular since the seminal paper [30] and [36]. More recently they have been used (in particular the "precubical" ones) in [10] and [11] for deriving new and interesting deadlock detection algorithms. More algorithms have been designed since then, see for instance [31], [9] and [12]. Notice that there is an alternative presentation of HDA [32], which is fairly close to what would be a labelled version of precubical sets, but not quite; our interest here is to link concurrent models with standard notions from combinatorial algebraic topology, hence the use of a different formalism, which moreover gives us a lot of categorical structure for free.

4.1 "Precubical" sets

Definition 4. *A precubical set K is a family of sets $\{K_n \mid n \geq 0\}$ with face maps $\partial_i^\alpha : K_n \to K_{n-1}$ $(0 \leq i \leq n-1,\ \alpha = 0,1)$ satisfying the following commutation rules:*

$$\partial_i^\alpha \partial_j^\beta = \partial_{j-1}^\beta \partial_i^\alpha \ (i < j)$$

Elements of K_n are called n-transitions. An simple example of a 2-dimensional pre-cubical set (which should represent a in parallel with b) is given below:



where A is a 2-transition, a, b, a', b' are 1-transitions and s_0, s_1, s_2 and s_3 are all 0-transitions (or states). We have $\partial_0^0(A) = a$, $\partial_0^1(A) = a'$, $\partial_1^0(A) = b$, $\partial_1^1(A) = b'$, $\partial_0^0(a) = \partial_0^0(b) = s_0$, $\partial_0^1(a) = s_1 = \partial_0^0(b')$, $\partial_0^1(b) = \partial_0^0(a') = s_2$ and $\partial_1^1(b') = \partial_1^1(a') = s_3$. One can readily check the commutation rules of the definition, for instance,

$$\partial_0^0 \partial_1^1(A) = \partial_0^0(b') = s_1 = \partial_0^1(a) = \partial_0^1 \partial_0^0(A)$$

Let K and L be two precubical sets. Then $f = (f_n)_{n \in \mathbb{N}}$ is a morphism of precubical sets from K to L if for all $n \in \mathbb{N}$, f_n is a function from K_n to L_n such that $f_n \circ \partial_i^\alpha = \partial_i^\alpha \circ f_{n+1}$ (for all i, $0 \leq i \leq n$).

This forms a category called Υ^S. It is a presheaf category as follows. Let $\Box^S$ be the free category whose objects are $[n]$, where $n \in \mathbb{N}$, and whose morphisms are generated by $[n-1] \overset{\delta_i^0}{\underset{\delta_j^1}{\rightrightarrows}} [n]$ for all $n \in \mathbb{N} \backslash \{0\}$ and $0 \leq i, j \leq n-1$, such that $\delta_j^k \delta_i^l = \delta_{i-1}^l \delta_j^k \ (0 \leq i < j)$. Now, the presheaf category $\mathrm{Set}^{(\Box^S)^{op}}$ of *contravariant* functors from $\Box^S$ to Set (morphisms are natural transformations) is isomorphic to the category of precubical sets. This implies, by general theorems ([25] and [27]), that Υ^S is an elementary topos. Moreover it is complete and co-complete because Set is complete and co-complete. Also, we will use the general fact in the sequel that in all categories of presheaves $\mathrm{Set}^{\mathcal{D}^{op}}$ like this one, all elements (which are contravariant functors) are direct limits of so-called representable functors $h^{\mathcal{D}}$ (Yoneda embedding) which to every $d \in \mathcal{D}$ associate $(x \to Hom_{\mathcal{D}}(x,d)) \in \mathrm{Set}^{\mathcal{D}^{op}}$[1].

[1] This is for instance classical in the categorical presentation of simplicial sets, see for instance [14].

4.2 Cubical sets

Precubical sets are a bit like the category of transition systems with no idle transitions: paths are transformed by morphisms into paths of the same length. This is far too strict to be really useful. For instance, simulations (hence bisimulations) cannot be morphisms (respectively spans of open morphisms as in [24]) in general. Also, it is impossible to describe the restriction to some subset of transitions (projection, restriction in CCS for instance) as a morphism. This needs a generalization of idle transitions to higher dimensions. There is in fact a close notion in cubical sets:

Definition 5. *A cubical set K is a precubical set together with degeneracy maps $\epsilon_i : K_{n-1} \rightarrow K_n$ $(0 \leq i \leq n-1)$ satisfying the extra cubical relations:*

$$\epsilon_i \epsilon_j = \epsilon_{j+1} \epsilon_i \qquad (i \leq j)$$

$$\partial_i^\alpha \epsilon_j = \begin{cases} \epsilon_{j-1} \partial_i^\alpha & (i < j) \\ \epsilon_j \partial_{i-1}^\alpha & (i > j) \\ Id & (i = j) \end{cases}$$

Let K and L be two cubical sets. Then f is a morphism of cubical sets from K to L if it is a morphism of precubical sets from the underlying precubical sets, and $f_{n+1} \circ \epsilon_j = \epsilon_j \circ f_n$ (for all $n \in \mathbb{N}$, $0 \leq j \leq n$).

The corresponding category of cubical sets, Υ, is isomorphic to the category of presheaves $\text{Set}^{\Box^{op}}$ over a small category $\Box$ (containing generating morphisms $e_i : [k+1] \rightarrow [k]$, $0 \leq i \leq k$, generating the degeneracies ϵ_i, besides the $\delta_i^k : [k-1] \rightarrow [k]$). This latter can be described in a nice way, see [6]. Therefore, similarly to the case of the category of precubical sets, the category of cubical sets is an elementary topos, which is complete and co-complete. We do not talk about cubical sets with connections and compositions here [4], but they have a great interest for our purposes, see for instance [15].

4.3 Some useful functors

There again, we need two interesting (and quite classical in spirit) functors. Let Υ_n be the category of Υ, whose objects are the n-dimensional cubical sets, i.e. the "cubical sets M with $M_k = \emptyset$ for all $k > n$". This category can be seen as the presheaf category $\text{Set}^{(\Box^{\leq n})^{op}}$ where $\Box^{\leq n}$ is the full subcategory of $\Box$ where objects are $[p]$ with $p \leq n$. Similarly, we define Υ_n^S, the category of n-dimensional precubical sets, seen as the presheaf category $\text{Set}^{\left((\Box^S)^{\leq n}\right)^{op}}$.

Lemma 1. *Let T_n (respectively T_n^S) be the function from Υ (respectively Υ^S) to Υ_n (respectively Υ_n^S), which to every $M \in \Upsilon$ (respectively $M \in \Upsilon^S$) associates $N \in \Upsilon_n$ (respectively $N \in \Upsilon_n^S$) with, $N([k]) = M([k])$ if $k \leq n$, $N(e_i : [k+1] \rightarrow [k]) = M(e_i)$ for $k < n$ and $N(\delta_i^\alpha : [k-1] \rightarrow [k]) = M(\delta_i^\alpha)$ for $k < n$. It defines a functor, called the n-truncation functor.*

The second functor is one which permits to build a natural cubical set from a precubical set:

Lemma 2. *There is a functor "free cubical set from a precubical set" $F : \Upsilon^S \rightarrow \Upsilon$ which is left-adjoint to the (obvious) forgetful functor K from Υ to Υ^S. Similarly, there is a functor "free cubical set of dimension less or equal than n from a precubical set of dimension less or equal than n", $F_n : \Upsilon_n^S \rightarrow \Upsilon_n$ which is left-adjoint to the (obvious) forgetful functor K_n from Υ_n to Υ_n^S.*

The proof uses a special form of Freyd's special adjoint functor theorem (which is also some form of Kan extension in presheaf categories), which is Proposition 1.3. of [14] (see Appendix A).

4.4 Labelled Cubical Sets

One remaining problem now, is that we do not have labels on transitions. This is easily taken care of by the following trick. Consider the category Υ^L of labelled cubical sets consisting of morphisms $l : M \rightarrow E$, where M is the underlying "unlabelled" cubical set and E is a "labelling" cubical set.

The morphisms in this category are as usual $f = (g, h) : (l : M \rightarrow E) \rightarrow (l' : M' \rightarrow E')$ with $g : M \rightarrow M'$ and $h : E \rightarrow E'$ such that the diagram

$$\begin{array}{ccc} M & \xrightarrow{g} & M' \\ {\scriptstyle l}\downarrow & & \downarrow{\scriptstyle l'} \\ E & \xrightarrow{h} & E' \end{array}$$

is commutative. By abuse of notation, we will sometimes identify f, g and h in the following. Of course, Υ^L is the comma category (see [26]) $(Id_\Upsilon \downarrow Id_\Upsilon)$. We will also consider in the following the category Υ_*^L of "pointed" labelled cubical sets, i.e. pairs $(l : M \rightarrow L, s)$ with $l \in \Upsilon^L$ and $s \in M_0$ (the "initial" state) and morphisms preserving initial states. We call this category, the category of Higher-Dimensional Transition Systems.

Given an alphabet ("of actions") Σ, we can construct a "labelling" *precubical set* $!\Sigma$ as follows. Suppose first that we have a total order $\prec$ on Σ; we then set:

- $(!\Sigma)_n$ is the set of increasing sequences of length n of letters of Σ, (or "well-ordered" words of length n written on Σ). For instance $(!\Sigma)_0$ consists of the empty word and $(!\Sigma)_1$ is Σ,
- $\partial_i^0(a_1, \cdots, a_n) = \partial_i^1(a_1, \cdots, a_n) = (a_1, \cdots, \hat{a_i}, \cdots, a_n)$ i.e., the word where the ith letter has been forgotten,

Then we set $!\Sigma$ again, by an abuse of notation, to be the free cubical set generated by the previous cubical set. Geometrically, $!\Sigma$ is in dimension one the wedge of a set of loops, one for each $\sigma \in \Sigma$ (giving the labels for 1-transitions). In dimension two, it is a wedge of a set of tori, one for each pair $(\sigma, \tau) \in \Sigma \times \Sigma$, now seen as a set (giving the labels for 2-transitions) etc. As an example, consider

Figure 1. On the left hand side, the "unlabelled" cubical set is the one taken as an example in Section 4.1. On the right-hand side is pictured a torus (this is a cubical set indeed) in which label a is the small circle, label b is the big circle, and label $a \mid b$ is the surface itself. The labelling morphism associates a and a' with a, b and b' with b and A with $a \mid b$.

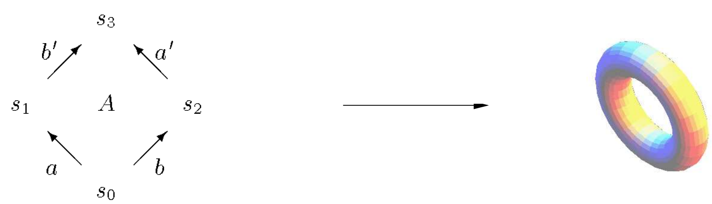


Fig. 1. A labelled cubical set

5 Some adjunctions

5.1 With transition systems

We prove that some suitable full subcategory of $(\Upsilon^L_*)_1$ is isomorphic to TS. Consider HTS to be the category whose objects are the pointed labelled cubical sets $(M, l : M \to E, i)$ such that,

- they are freely generated by precubical sets, i.e. $M = F(N)$, $l = F(l')$ with $l' : N \to F$ morphism of precubical sets,
- they are "strongly-labelled"[2], i.e. $\forall x, x' \in M_k$ $(k \geq 1)$,

$$\left(\partial^0_i(x) = \partial^0_i(x'), \partial^1_i(x) = \partial^1_i(x')\ (\forall 0 \leq i < k)\ , l(x) = l(x')\right) \Longrightarrow x = x'$$

and whose morphisms are all morphisms of pointed labelled cubical sets. HTS_1 is the full sub-category of HTS consisting of pointed labelled cubical sets of dimension at most one.

As a matter of fact, the categories are defined in quite similar terms. States of ordinary transition systems are of the same nature as states of labelled cubical sets and source and target representation of transitions is nothing but a functional interpretation of the relation $Tran$. This is done formally by constructing two functors $\mathcal{U} : TS \to HTS_1$ and $\mathcal{V} : HTS_1 \to TS$ inverse of each other, with,

[2] This technical condition is necessary for having a smooth relation between the labelled graph kind of presentation of a transition system, with a transition relation kind of presentation; see the proof in the Appendix.

- $(F(M), F(l) : F(M) \rightarrow F(E), i) = \mathcal{U}(S, A, Tran, j)$ with,
  - $M_0 = S$,
  - $M_1 = \{a_{s,s'} \mid a \in A, s \xrightarrow{a} s' \in Tran\}$,
  - $i = j$,
  - $\partial_0^0(a_{s,s'}) = s$, $\partial_0^1(a_{s,s'}) = s'$,
  - $E = K_1(T_1(!A))$,
  - $l(a_{s,s'}) = a$, $l(s) = 1$.
- $(S, A, Tran, j) = \mathcal{V}(F(M), F(l) : F(M) \rightarrow F(E), i)$ with,
  - $S = M_0$,
  - $j = i$,
  - $A = E_1$,
  - $s \xrightarrow{a} s' \in Tran$ if $\exists x \in M_1$, such that $l(x) = a$, $\partial_0^0(x) = s$ and $\partial_0^1(x) = s'$ (then this x is unique because $(F(M), F(l), i)$ is strongly-labelled).

Action of the functors on morphisms is as follows,

- if $f = (\sigma, \lambda) : (S_0, A_0, Tran_0, j_0) \rightarrow (S_1, A_1, Tran_1, j_1)$ is a morphism of transition systems then we define $\mathcal{U}(f) = (\mathcal{U}(f)^1, \mathcal{U}(f)^2)$ where $\mathcal{U}(f)^1 : F(M_0) \rightarrow F(M_1)$ and $\mathcal{U}(f)^2 : F(E_0) \rightarrow F(E_1)$ are the two components of the morphism, where $\mathcal{U}(S_0, A_0, Tran_0, j_0) = (F(M_0), F(l_0) : F(M_0) \rightarrow F(E_0), j_0)$, $\mathcal{U}(S_1, A_1, Tran_1, j_1) = (F(M_1), F(l_1) : F(M_1) \rightarrow F(E_1), j_1)$.
  - $\mathcal{U}(f)^1(a_{s,s'}) = \begin{cases} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } \lambda(a) \neq * \\ \epsilon_0(\sigma(s)) & \text{otherwise} \end{cases}$,
  - $\mathcal{U}(f)^1(s) = \sigma(s)$ $(s \in M_0)$,
  - $\mathcal{U}(f)^2(a_{s,s'}) = \begin{cases} \lambda(a) & \text{if } \lambda(a) \neq * \\ \epsilon_0(1) & \text{otherwise} \end{cases}$,
  - $\mathcal{U}(f)^2(s) = 1$ $(s \in M_0)$.
- if $f = (f^1, f^2) : (l_0 : M_0 \rightarrow E_0, i_0) \rightarrow (l_1 : M_1 \rightarrow L, i_1)$ is a morphism in HTS_1, then $\mathcal{V}(f) = (\sigma, \lambda) : \mathcal{V}(l_0 : M_0 \rightarrow E_0, i_0) \rightarrow \mathcal{V}(l_1 : M_1 \rightarrow E_1, i_1)$ with
  - $\sigma(s) = f^1(s)$ (for all s state of $\mathcal{V}(l_0 : M_0 \rightarrow E_0, i_0)$),
  - $\lambda(a) = \begin{cases} f^2(a) & \text{if } f^2(a) \notin \text{Im } \epsilon_0 \\ * & \text{otherwise} \end{cases}$ (for all a label in $\mathcal{V}(l_0 : M_0 \rightarrow E_0, i_0)$)

In the sequel we will restrict functors and categories of models so that they have "fixed labellings". We call HTS the category of higher-dimensional transition systems labelled over a fixed cubical set $!E$ for a given (fixed once and for all in all the following arguments) set of labels E. We will no longer mention these labelling sets. Given this restriction,

Theorem 1. *$\mathcal{U}$ and $\mathcal{V}$ are inverse functors.*

Now, in order to compare the category of higher-dimensional transition systems with ordinary transition systems we only have to look at how to retract HTS onto its sub-category HTS_1. This boils down to looking at the different adjunctions we have between Υ and Υ_1 because of the few next lemmas. The first one tells us that we can lift adjunctions from unlabelled to labelled cases, and the second one tells us that we can restrict adjunctions (this is useful for dealing with the "strong labelling condition" of labelled cubical sets).

Lemma 3. *Let $\mathcal{C}$ and $\mathcal{D}$ be two categories and $S_{\mathcal{C},\mathcal{D}}$ be the set of all pairs of functors (F,G) with $F:\mathcal{C}\to\mathcal{D}$ left adjoint to $G:\mathcal{D}\to\mathcal{C}$. Then all elements of $S_{\mathcal{C},\mathcal{D}}$ induce elements of $S_{(Id_\mathcal{C}\downarrow Id_\mathcal{C}),(Id_\mathcal{D}\downarrow Id_\mathcal{D})}$.*

Lemma 4. *Let $\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$ be a pair of adjoint functors, $\mathcal{C}'$ (respectively $\mathcal{D}'$) a full sub-category of $\mathcal{C}$ (respectively of $\mathcal{D}$). Suppose that $F(\mathcal{C}')\subseteq\mathcal{D}'$ and $G(\mathcal{D}')\subseteq\mathcal{C}'$, then $\mathcal{C}' \underset{G_{|\mathcal{D}'}}{\overset{F_{|\mathcal{C}'}}{\rightleftarrows}} \mathcal{D}'$ is a pair of adjoint functors.*

We have mainly two different adjunctions between Υ and Υ_1 using T_1 (to keep the underlying ordinary transitions unchanged in the interpretation) among all the possible ones.

Proposition 1. *There are pairs of adjoint functors as follows (for $n\geq 1$):*

- *There is a functor $\mathcal{I}_n:\Upsilon_n\to\Upsilon$ left-adjoint to the truncation functor $T_n:\Upsilon\to\Upsilon_n$. Similarly, there is a functor $\mathcal{I}_n^S:\Upsilon_n^S\to\Upsilon^S$ left-adjoint to the truncation functor $T_n:\Upsilon^S\to\Upsilon_n^S$. Moreover, $\mathcal{I}_n$ and T_n commute with the free functor.*
- *The truncation functor $T_n:\Upsilon\to\Upsilon_n$ (respectively $T_n^S:\Upsilon^S\to\Upsilon_n^S$) is left-adjoint to a functor $\mathcal{G}_n:\Upsilon_n\to\Upsilon$ (respectively $\mathcal{G}_n^S:\Upsilon_n^S\to\Upsilon^S$).*

Proof. These are direct applications of Proposition 1.3. of [14] (see Appendix A).

The intuition about these functors is as follows. $\mathcal{I}_n$ is just some kind of inclusion functor; it takes a n-dimensional cubical set and forms a cubical set with exactly the same non-degenerated elements (i.e. those elements which are not in some Im ϵ_i); in fact, exactly the same elements in dimension less or equal than n, but only degenerated elements in dimension strictly bigger than n. Seen as some kind of abstraction (in the sense of abstract interpretation [5]), it is a "minimal allocation strategy" abstraction. A n-dimensional cubical set only prescribes what can happen for degrees of concurrency less or equal than n. $\mathcal{I}_n$ interprets this as being exactly with no (interesting) actions with more than n processes busy at the same time. On the contrary $\mathcal{G}_n$ tries to interpret a n-dimensional cubical set with "maximal allocation strategy" i.e. tries to fill in all $(n+1)$-dimensional holes in a n-dimensional cubical set as imposing that this should be filled in by a $(n+1)$-transition, and up and up in all dimensions. There are "dihomotopy" properties that should be proven about this "resolution" like functor. This is left for future work.

We notice now that the adjunction $(\Upsilon_*^L)_n \underset{T_n}{\overset{\mathcal{I}_n}{\rightleftarrows}} \Upsilon_*^L$ can be restricted using Lemma 4 to the full sub-categories of free objects generated by precubical sets, in, respectively, $(\Upsilon_*^L)_n$ and Υ_*^L. This is due to the fact that (see Proposition 1) $\mathcal{I}_n$ and T_n commute with the "free functors". We can restrict this adjunction furthermore, still using Lemma 4, to see that the adjunction still holds with $n\geq 1$

when we restrict to strongly-labelled automata. Hence we have the adjunction: $HTS_1 \underset{T_1}{\overset{\mathcal{I}_1}{\rightleftarrows}} HTS$. Given that HTS_1 and TS are isomorphic (see Theorem 1), we deduce that we have a pair of adjoint functors: $TS \underset{ht}{\overset{th}{\rightleftarrows}} HTS$. Unfortunately, we did not manage yet to "lift" the other adjunction of Proposition 1 to higher-dimensional transition systems.

5.2 With automata with concurrency relations

We first define functors $\mathcal{W}, \mathcal{Y}$, which will be proven to be inverse functors:

$$ACR \underset{\mathcal{Y}}{\overset{\mathcal{W}}{\leftrightarrows}} HTS_2$$

(HTS_2 is the full subcategory of Υ_*^L consisting of higher-dimensional transition systems of dimension less than or equal to two) by,

- $(F(P), F(l : P \to L), F(j))$=$\mathcal{Y}(S, i, E, I, Tran)$ with,
  - $j = i$,
  - $P_0 = S$,
  - $P_1 = \{t_{s,s'} \mid s \xrightarrow{t} s' \in Tran\}$,
  - $L = K_2(T_2(!E))$,
  - $\partial_0^0(t_{s,s'}) = s$, $\partial_0^1(t_{s,s'}) = s'$ and $l(t_{s,s'}) = t$,
  - $P_2 = \{ab_{s,s',s'',u} \mid aI_sb \wedge a_{s,s'} \in P_1 \wedge b_{s,s''} \in P_1 \wedge b_{s',u} \in P_1 \wedge a_{s'',u} \in P_1\}$,
  - $\partial_0^0(ab_{s,s',s'',u}) = a_{s,s'}$ (or $\partial_1^0(ab_{s,s',s'',u}) = a_{s,s'}$, depending on the way this is coded in $!E$), $\partial_1^0(ab_{s,s',s'',u}) = b_{s,s''}$ (or $\partial_0^0(\cdots) = \cdots$), $\partial_1^1(ab_{s,s',s'',u})$ $= b_{s',u}$, (respectively, or $\partial_0^1(\cdots) = \cdots$), $\partial_0^1(ab_{s,s',s'',u}) = a_{s'',u}$ (respectively $\cdots$) and $l(ab_{s,s',s'',u}) = (a, b)$ (respectively $\cdots$).
- $\mathcal{W}(P, P \xrightarrow{l} L, j) = (S, i, E, I, Tran)$ with,
  - $(S, i, E, Tran) = \mathcal{V}(T_1(P), T_1(l), j)$,
  - aI_sb if there exist $x, x', y, y' \in P_1$, $C \in P_2$ with $l(x) = a$, $l(x') = a$, $l(y) = b$, $l(y') = b$ and $\partial_0^0(x) = \partial_0^0(y) = s$, $\partial_0^1(x) = \partial_0^0(y')$, $\partial_0^1(y) = \partial_0^0(x')$, $\partial_1^1(y') = \partial_1^1(x')$, $l(C) = (a, b)$, $\partial_0^0(C) = x$, $\partial_1^0(C) = y$, $\partial_0^1(C) = y'$ and $\partial_1^1(C) = x'$ (or, respectively, $\partial_1^0(C) = x$, $\partial_0^0(C) = y$, $\partial_1^1(C) = y'$ and $\partial_0^1(C) = x'$).

$\mathcal{Y}$ has the same action on the underlying ordinary transition system of an asynchronous transition system as functor $\mathcal{U}$; we will identify $\mathcal{U}(S, i, E, Tran)$ with the underlying 1-dimensional skeleton of the higher-dimensional transition system $\mathcal{Y}(S, i, E, I, Tran)$. Similarly for $\mathcal{W}$ which acts as $\mathcal{V}$ on the underlying ordinary transition systems, thus we will identify $\mathcal{V}(P, l : P \to L, j)$ as the underlying transition system of the asynchronous transition system $\mathcal{W}(P, l : P \to L, j)$. $\mathcal{Y}$ fills in all interleavings of two independent actions by 2-transitions $\mathcal{W}$ imposes two actions to be independent if and only if there exists a truly concurrent execution of them in the higher-dimensional transition system. The action on morphisms is again easy to define. Let $f = (\sigma, \lambda) : (S, i, E, I, Tran) \rightarrow (S', i', E', I', Tran')$ be a morphism of asynchronous transition systems. Then $g = \mathcal{Y}(f) : \mathcal{Y}(S, i, E, I, Tran) \rightarrow \mathcal{Y}(S', i', E', I', Tran')$ is defined by,

– $T_1(g) = \mathcal{U}(f)$ (by the identification made above),

– $g_2(ab_{s,s',s'',u}) = \begin{cases} \lambda(a)\lambda(b)_{\sigma(s),\sigma(s'),\sigma(s''),\sigma(u)} & \text{if } \lambda(a) \neq * \text{ and } \lambda(b) \neq * \\ \epsilon_0\left(\lambda(a)_{\sigma(s),\sigma(s')}\right) & \text{if } \lambda(a) \neq * \text{ and } \lambda(b) = * \\ \epsilon_1\left(\lambda(b)_{\sigma(s),\sigma(s'')}\right) & \text{if } \lambda(b) \neq * \text{ and } \lambda(a) = * \\ \epsilon_0\epsilon_0(\sigma(s)) & \text{if } \lambda(a) = * \text{ and } \lambda(b) = * \end{cases}$

for $ab_{s,s',s'',u} \in \mathcal{Y}(S, i, E, I, Tran)_2$.

Finally, for $g : (P, P \xrightarrow{l} L, j) \rightarrow (P', P' \xrightarrow{l'} L', j')$ a morphism of $(\Upsilon_*^L)_2$ we define $f = (\sigma, \lambda) : \mathcal{W}(P, P \xrightarrow{l} L, j) \rightarrow \mathcal{W}(P', P' \xrightarrow{l'} L', j')$ simply by (using the previous identification) $f = \mathcal{V}(T_1(g) : T_1(P), T_1(l), j) \rightarrow (T_1(P'), T_1(l'), j')$.

In the sequel we will again fix once and for all the labelling cubical set used in our higher dimensional transition systems, to be $!E$ (where E is a set of labels fixed once and for all). Then again,

Theorem 2. *$\mathcal{W}$ and $\mathcal{Y}$ are well-defined functors. Moreover, $\mathcal{Y}$ and $\mathcal{W}$ are inverse of each other.*

Proof. The only difficulty, is to show that the action of these functions on morphisms are well-defined. For $\mathcal{Y}$, the only thing to check is that the definition in dimension 2 of the underlying precubical set is coherent. We only check one of the necessary equalities: (taking the same notations as above), for $ab_{s,s',s'',u} \in \mathcal{Y}(S', i', E', I', Tran')$ with $\lambda(a) \neq *$ and $\lambda(b) = *$ (notice that we have then $\sigma(s'') = \sigma(s)$ and $\sigma(s') = \sigma(u)$),

$$\begin{aligned} \partial_l^k(g_2(ab_{s,s',s'',u})) &= \partial_l^k(\epsilon_0(\lambda(a)_{\sigma(s),\sigma(s')})) \\ &= \begin{cases} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k = 0,\ l = 0 \\ \epsilon_0(\partial_0^0(\lambda(b)_{\sigma(s),\sigma(s'')})) = \epsilon_0(\sigma(s)) & \text{if } k = 0,\ l = 1 \\ \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k = 1,\ l = 0 \\ \epsilon_0(\partial_0^1(\lambda(b)_{\sigma(s'),\sigma(u)})) = \epsilon_0(\sigma(u)) & \text{if } k = 1,\ l = 1 \end{cases} \\ g_1(\partial_l^k(ab_{s,s',s'',u})) &= \begin{cases} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k = 0,\ l = 0 \\ \epsilon_0(\sigma(s)) & \text{if } k = 0,\ l = 1 \\ \lambda(a)_{\sigma(s''),\sigma(u)} = \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k = 1,\ l = 0 \\ \epsilon_0(\sigma(u)) & \text{if } k = 1,\ l = 1 \end{cases} \end{aligned}$$

which are equal. The rest of the proof goes along the same lines (see the rest in Appendix A).

For $\mathcal{W}$ we have to check that, for $f = (\sigma, \lambda) = \mathcal{Y}(g : (P, l : P \rightarrow L, i) \rightarrow (P', l' : P' \rightarrow L', i'))$, aI_sb and $\lambda(a) \neq *$, $\lambda(b) \neq *$ implies $\lambda(a)I'_{\sigma(s)}\lambda(b)$. Suppose aI_sb in $\mathcal{Y}(P, l : P \rightarrow L, i)$. Then there exist $x, x', y, y' \in P_1$ with $l(x) = a$, $l(x') = a$, $l(y) = b$, $l(y') = b$ and $\partial_0^0(x) = \partial_0^0(y) = s$, $\partial_0^1(x) = \partial_0^0(y')$, $\partial_0^1(y) = \partial_0^0(x')$, $\partial_1^1(y') = \partial_1^1(x')$, and we have a $C \in P_2$ with $l(C) = (a, b)$, $\partial_0^0(C) = x$, $\partial_1^0(C) = y$, $\partial_0^1(C) = y'$ and $\partial_1^1(C) = x'$. We know that $g(C) \in P'_2$ and that $l' \circ g(C) = (f(a), f(b))$ since $f(a) \neq *$ and $f(b) \neq *$. Similarly, $l'(g(x)) = f(a)$,

$l'(g(x')) = f(a)$, $l'(g(y)) = f(b)$, $l'(g(y')) = f(b)$. Furthermore, because g is a morphism of cubical sets, $\partial_0^0(g(x)) = \partial_0^0(g(y)) = \sigma(s)$, $\partial_0^1(g(x)) = \partial_0^0(g(y'))$, $\partial_0^1(g(y)) = \partial_0^0(g(x'))$, $\partial_1^1(g(y')) = \partial_1^1(g(x'))$, so $\lambda(a) I'_{\sigma(s)} \lambda(b)$.

The adjunctions of Proposition 1, in the particular case $n = 2$, together with the result of Theorem 2 imply that we have a pair of adjoint functors: $ACR \underset{ha}{\overset{ah}{\rightleftarrows}} HTS$.

6 Another formulation with scones

In fact, the results of previous section still hold if we restrict slightly the category of labelled cubical sets we are considering. What we have to notice is that labelled cubical sets look very much like *scones* or logical relations [28, 1] $(\Upsilon \downarrow !)$:

Definition 6. *Let $\Gamma : \mathcal{A} \to \mathcal{D}$ be a functor. Let $\mathcal{E} = (\mathcal{D} \downarrow \Gamma)$ be the scone (comma category) on Γ, i.e. the category which has:*

- *as objects: (d, a, f) where $d \in \mathcal{D}$, $a \in \mathcal{A}$ and $f : d \to \Gamma(a)$ is a morphism in $\mathcal{D}$,*
- *as morphisms: $(s, t) : (d, a, f) \to (d', a', f')$ with $s : d \to d'$ is a morphism in $\mathcal{D}$ and $t : a \to a'$ is a morphism in $\mathcal{A}$ such that the following diagram commutes:*

$$\begin{array}{ccc} d & \xrightarrow{\ s\ } & d' \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f'} \\ \Gamma(a) & \xrightarrow[\Gamma(t)]{} & \Gamma(a') \end{array}$$

Take $\mathcal{A} = FinOrd$, the category of finite ordinals and increasing maps (i.e. the base category Δ of simplicial sets!) $\mathcal{D} = \Upsilon$ and Γ the functor induced by ! as follows: $\Gamma(t)(a_1, \cdots, a_n) = (t(a_1), \cdots, t(a_n))$ where t is an increasing function. Then the objects of $(\mathcal{D} \downarrow \Gamma)$ are exactly labelled cubical sets whereas morphisms of $(\mathcal{D} \downarrow \Gamma)$ are those morphisms of labelled cubical sets which are generated by some "renaming function" between alphabets, i.e. are some $\Gamma(t)$ where t is a morphism in the category of finite ordinals.

It is easily seen that Γ commutes with all limits (in fact $\Gamma(\Sigma)$ is deduced from a nerve construction on Σ, i.e. is a right adjoint of some functor G as in Proposition 3). $FinOrd$ is a complete and co-complete topos [18], as is Υ. By [13], we know that $(\mathcal{D} \downarrow \Gamma)$ is a topos.

Notice as well that any Barr-Beck cotriple homology [26] on $FinOrd = \Delta$ can be lifted onto a Barr-Beck homology theory on $(\mathcal{D} \downarrow \Gamma)$:

Proposition 2. *Let $\mathcal{A}$ and $\mathcal{D}$ be two cartesian categories. Let $\Gamma : \mathcal{A} \rightarrow \mathcal{D}$ be a functor which commutes with cartesian products and terminal objects. Let $(L, \epsilon : L \rightarrow Id, \delta : L \rightarrow L^2)$ be a comonad on $\mathcal{A}$. Then this induces a comonad $(\tilde{L}, \tilde{\epsilon}, \tilde{\delta})$ on $(\mathcal{D} \downarrow \Gamma)$.*

Applying this with $\mathcal{A} = \Delta$, $\mathcal{D} = \Upsilon$ and $\Gamma =!$, this implies that any comonad on Δ will induce a homology theory on labelled cubical sets, hence on asynchronous transition systems and transition systems. This would allow us to lift homological reasoning from unlabelled (as in [11] and the beginning of [21]) nicely to labelled automata. We can think of several interesting comonads on Δ, for instance, the subdivision comonad:

Definition and lemma 3 – *let $f : \Delta \rightarrow \Delta$ be the functor which to any order $(S, \leq)$ associates the order $(S', \subseteq)$ with S' being the set of all linearly ordered finite and non-empty subsets of S and $\subseteq$ is the set-theoretic inclusion.*

- *For E in Δ, let $d_E : f(E) \rightarrow E$ be defined by $d_E(\{x_1 < \cdots x_k\}) = \sup\{x_1 < \cdots < x_k\} = x_k$.*
- *Let also $s_E : f(E) \rightarrow f^2(E)$ be defined by $s_E(\{x_1 < \cdots x_k\}) = \{\{x_1\}, \{x_1 < x_2\}, \cdots, \{x_1 < \cdots x_k\}\}$. $E \rightarrow d_E$ and $E \rightarrow s_E$ are respectively the co-unit and the comultiplication of a comonad (f, d, s) (see Appendix A for a proof).*

What do such homology theories classify? By lifting this comonad to simplicial sets, this gives the subdivision comonad, based on the barycentric subdivision functor: my current hope is that it will classify phenomena linked with obstructions to refinement of actions. This is left for future work.

7 Application to the state-space explosion problem

Stubborn sets [35], sleep sets and persistent sets [16] are methods used for diminishing the complexity of model-checking using transition systems. They are based on semantic observations using Petri nets in the first case and Mazurkiewicz trace theory in the other one. We believe that these are special forms of "homotopy retracts" when cast (using the adjunctions we have hinted) in the category of higher-dimensional transition systems. We hope to make this statement more formal, through these adjunctions, and use this to design new state-space reduction methods. Let me explain the intuition behind the scene.

Let T be a set of actions, $T \subseteq E$, and $p \in S$ be a state. We say that T is persistent [16] in state p if,

- T contains only actions which are enabled at p, and,
- for all traces t beginning at p containing only actions q out of T, qIT.

Suppose we have a set of persistent actions T_p for all states p in an asynchronous transition system. Then let us look at the following set of traces PT (identified with a series of states) in $(S, i, E, Tran, I)$ defined inductively [16] as follows:

- $(i) \in PT$,

– if $(p_1, \ldots, p_n) \in PT$, then $(p_1, \ldots, p_n, q) \in PT$ where $p_n \xrightarrow{t'} q \in Tran$ and $t' \notin T_{p_n}$.

Then checking deadlock detection can be made on this subset PT of traces instead of the full set of traces of $(S, i, E, Tran, I)$. Also, when $(S, i, E, Tran, I)$ is acyclic (but it can be modified so that the method works again), PT is enough for checking LTL temporal formulas.

We exemplify the method on the process $Pb.Pa.Vb.Va \mid Pa.Pb.Va.Vb$. A standard interleaving semantics would be as sketched in Figure 2, showing the presence of deadlocking state 19. One set of persistent sets is:

– $T_1 = \{Pa\}$, $T_2 = \{Pb\}$, $T_5 = \{Pa, Pb\}$, $T_6 = \{Pb, Va\}$, $T_8 = \{Pa, Va\}$, $T_{13} = \emptyset$,
– $T_9 = \{Vb\}$, $T_{12} = \{Va\}$, $T_{17} = \{Pb\}$, $T_{18} = \{Va\}$, $T_{22} = \{Vb\}$, $T_{23} = \emptyset$,
– $T_7 = \{Pb, Vb\}$, $T_{14} = \{Vb\}$, $T_{15} = \{Pb\}$, $T_{16} = \{Pa\}$, $T_{20} = \{Vb\}$, $T_{21} = \{Va\}$.

and we show the corresponding traces PT in Figure 3. We have not indicated the persistent sets corresponding to 3, 4 etc. since in a persistent set search, they will not be reached anyway, so their actual choice is uninteresting.

In Figure 2 there are 16 paths from 1 to be traversed if no selective search was used. Six of them lead to the deadlock 13, and 10 (5 above the hole, 5 below the hole) are going to the final point 23. In Figure 3, one can check that there are only 8 paths to be traversed if one uses the persistent sets selective search (3 to state 13, 1 to state 23 below the hole and 4 to state 23 above the hole).

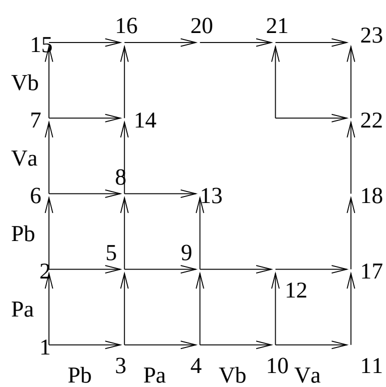


Fig. 2.

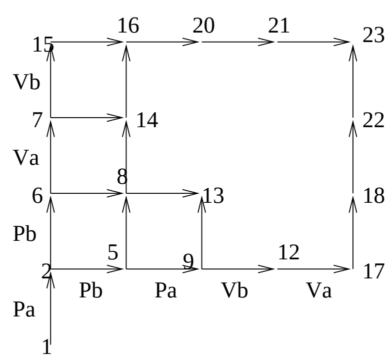


Fig. 3.

How did we find this set of persistent sets? In the PV case this can be done quite easily as follows. First the independence relation can be found out right away. Px and Py stand respectively for the query for a lock on x and y (nothing is committed yet) so they are two actions, whatever x and y are which are independent. But we should rather declare Px and Vy dependent in general since: if $x = y$ this is clear, and if $x \neq y$ this can come from the fact locks on x and y are causally related (precisely as in the case of Figure 2 with $x = a$ and $x = b$). This is slightly different from the more usual case of atomic reads and writes languages in which the independance relation can be safely determined as: actions are independent if and only if they act on distinct variables.

The most elaborated technique known in this framework is that of "stubborn sets" [35], which adapted to the presentation here, can be defined as (taken from [17]):

T_s is a stubborn set in a state s if T_s contains at least one enabled transition and for all t in T_s, the following two conditions hold:

- if t is disabled in s, and c_i is a necessary condition for t to be enabled which is false in s, then all transitions t' whose execution can make c_i true are also in T_s,
- if t is enabled in s, then all transitions t' such that t and t' are dependent are also in T_s.

The example of persistent set we gave in Figure 3 is clearly a stubborn set. As one can see as well, the persistent set approach here reduces the 5 paths below the hole into 1, which is a representant modulo dihomotopy of these 5 dipaths.

In the cubical sets approach, we have at hand a notion of diconnected components, first defined in [11], which characterize the "essential schedules" of executions of a parallel system. They form in fact a partition of the cubical set (or of its topological counterpart) together with a causal ordering (and an extra equivalence relation - which I will not detail here). It is shown in [22] to be a category of fractions of the fundamental category of the corresponding cubical set. Using this approach (there are some algorithms for determining these diconnected regions, see for instance [31]), one would find the set of 7 diconnected components and the corresponding graph of regions pictured in Figure 4.

There are 4 dipaths to be traversed in the graph of diconnected regions to determine the behaviour of this concurrent system (two of them leading to state 13 being dihomotopic); which looks better than with the persistent sets approach.

In fact, there are two explanations for this method of diconnected components to be better than the persistent set approach. First, in the persistent set approach, the independence relation does not in general depend on the current state (even if this might be changed by changing the set of labels), whereas our notion of independence is having a 2-transition, which depends on the current state. The second and more important reason is that the diconnected graph algorithm does determine regions because of global properties, whereas the persistent sets approach uses only (in general syntactic) local criteria for reducing the state-space.

Conversely, it is relatively easy to see the following. For all p state in our asynchronous transition system (or let us say, by the adjunction of Section 5.2,

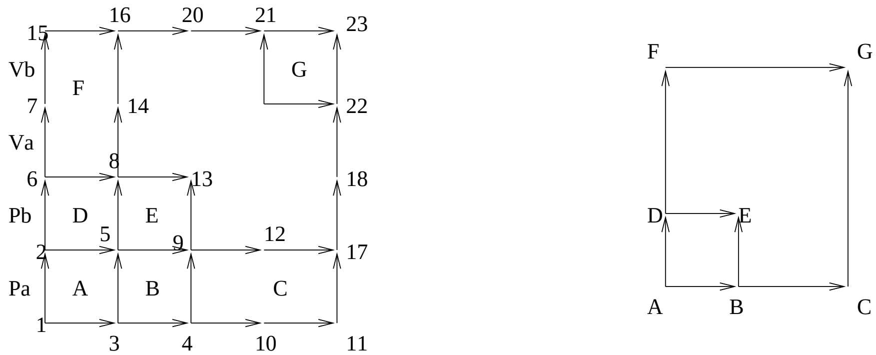


Fig. 4.

in a 2-dimensional cubical set), all traces t composed of actions out of T_p is such that all its actions are independent with T_p. So any trace from p made up of any action (those of T_p as well as those outside T_p) can be deformed (by dihomotopy, or "is equivalent to") into a trace firing first actions from T_p and then actions out of T_p. Therefore the selective search approach using only actions from T_p (for all p) is only traversing some representatives of the dihomotopy classes of paths. The persistent search approach is a particular (not optimal in general) case of dihomotopic deformation. We would like to understand better, through the adjunction of Section 5.2, the way one could see all these state-space reduction techniques as finding suitable dihomotopy retracts. This is left for future work.

8 Other adjunctions

In [37] some adjunctions are described between a variety of models for concurrency. We hope to be able to lift some of these functors to the case of labelled cubical sets. In particular, we believe that the equivalence between traces defined in the category TL of Mazurkiewicz traces should be mapped onto homotopy classes of traces in HTS, therefore the partially commutative monoid defined in Mazurkiewicz trace theory should be some analog of the fundamental category in cubical sets (defined for instance in [20]). This is left for future work. The domain of configurations of an event structure is a dI-domain (stable domain, à la Berry, see for instance [37]) and we believe that through adjunctions with HTS (and through the adjunctions between cubical sets and local po-spaces [11], using the geometrical realization functor), this is linked to the fact that partially ordered topological spaces are related to some particular forms of Scott domains (see again [20]). Finally, we believe that there is an equivalence of categories between some form of higher-dimensional transition systems and general Petri nets. One of the difficulties is in finding the right notion of independence between any number of transitions in Petri nets. One possible start is to use the adjunction between ACR and Petri nets in [7].

Nevertheless, it is not difficult to produce pairs of adjoint functors between the category of cubical sets (or precubical sets) and the category of Petri nets, or the category of "unlabeled" (prime) event structure as defined in [37], but the problem is to be sure that this is the translation we want, as we tried to show in this paper, in the case of transition systems, by using Proposition 3. For instance, let ES be the category of (unlabeled prime) event structures, i.e.,

- objects are prime event structures $(E, \leq, \#)$, where E is a set of *events* partially ordered by $\leq$ called the *causal dependency relation* and where $\# \subseteq E \times E$ is a symmetric irreflexive relation, the *conflict relation* satisfying,
  - $\{e'/e' \leq e\}$ is finite (axiom of "finite causes"),
  - $e\#e'$ and $e' \leq e''$ implies $e\#e''$ (conflict is hereditary).
- let $S = \{E, \leq, \#\}$ and $S' = \{E', \leq', \#'\}$ be event structures. A morphism of event structures from S to S' is a partial function $f : E \rightarrow E'$ such that,
  - if $f(e)$ is defined then $\{e'/e' \leq f(e)\} \subseteq f(\{e''/e'' \leq e\}$,
  - if $f(e_0)$ and $f(e_1)$ are both defined then $f(e_0)\#f(e_1)$ or $f(e_0) = f(e_1)$ implies $e_0\#e_1$ or $e_0 = e_1$.

To get an adjunction $ES \underset{R}{\overset{L}{\rightleftarrows}} \Upsilon$ we only have to find a functor $G : \square \rightarrow ES$. For instance, one can take:

- $G([0]) = \emptyset$ and for $n > 0$, $G([n])$ is the event structure with exactly n independent events $e_1, \cdots, e_n$,
- $G(\delta_i^k) : G([n-1]) \rightarrow G([n])$ associates with each e_j, $j \leq i$, $e_j \in G([n])$ and for $j > i$, it associates $e_{j+1} \in G([n])$.
- $G(e_i) : G([n]) \rightarrow G([n-1])$ associates with $e_j \in G([n])$, $j < i$, $e_j \in G([n-1])$, for $j > i$, it associates e_{j-1} and for $j = i$, it is not defined.

This looks reasonable as we associate with $[n]$ a set of n independent events, capturing the true-concurrency of the two models in the same way. The problem is prefixing, which is not preserved by this pair of adjoint functors. In fact, there is just no causal dependency relation generated at all by L! A solution is to set,

- $G([0]) = \{*\}$ and for $n > 0$, $G([n])$ is the event structure with exactly n independent events $e_1, \cdots, e_n$ after (in the $\leq$ order) event $*$,
- $G(\delta_i^k) : G([n-1]) \rightarrow G([n])$ associates with e_j, $j \leq i$, $e_j \in G([n])$, for $j > i$, it associates $e_{j+1} \in G([n+1])$, and with $*$ it associates $*$ if $k = 0$ otherwise it is undefined.
- $G(e_i) : G([n]) \rightarrow G([n-1])$ associates with $*$ itself and with each $e_j \in G([n])$, $j < i$, $e_j \in G([n-1])$, for $j > i$, it associates e_{j-1}, for $j = i$, it is not defined.

Now one can see that the functor preserves not only colimits but also prefixing. Yet another way of finding an adjunction between ES and Υ is as follows:

- $G([n])$ is the event structure $(\wp(\{0, \cdots, n\}, \subseteq, \emptyset)$,

– $G(\delta_i^k) : G([n-1]) \rightarrow G([n])$ associates with $\{a_1, \cdots, a_k\}$, $\{\delta_i^k(a_1), \cdots, \delta_i^k(a_k)\}$
– $G(e_i) : G([n]) \rightarrow G([n-1])$ associates with $\{a_1, \cdots, a_k\}$, $\{e_i(a_1), \cdots, e_i(a_k)\}$.

Now, colimits, products and prefixing are preserved. The further discussion of such adjunctions, together with the development of similar techniques to Petri nets is left for future work. One important gain would be to transport the methods used in some subcategories of the category of Petri nets, see for instance [8], for finding deadlocks, unreachable states etc., to HTS.

9 Conclusion and further work

We have seen that cubical sets form a complete and co-complete category. This means that the category of labelled cubical sets (with a fixed alphabet of the form $!E$) is complete and co-complete. Because it is related through left and right adjoints to transition systems (and asynchronous transition systems), there are some correspondences between limits and co-limits in these categories. For instance, products in higher-dimensional transition systems correspond to the parallel combination (with no interference) of the two higher-dimensional transition systems (as does the cartesian product of two partially ordered topological spaces); co-products correspond to non-deterministic choice. Fibred products, i.e. synchronized products as in the category of ordinary transition systems [2], allow for nice semantical definitions. This allows also for nice comparison of semantics through adjunctions.

Last but not least, in [24] is defined an abstract notion of bisimulation. Given a model for concurrency, i.e. a category of models **M** and a "path category", i.e. a subcategory of **M** which somehow represents what should be thought of as being paths in the models, then we can define two elements of **M** to be bisimilar if there exists a span of special morphisms linking them. These special morphisms have a path-lifting property that we believe would be in higher-dimensional transition systems a (geometric) fibration property. We thus hope that homotopy invariants could be useful for the study of a variety of bisimulation equivalences. Some work has been done in that direction in [32] (and in some sense also in [21]).

References

1. Moez Alimohamed. A characterization of lambda definability in categorical models of implicit polymorphism. *Theoretical Computer Science*, 146(1–2):5–23, July 1995.
2. A. Arnold. *Systèmes de transitions finis et sémantique des processus communicants*. Masson, 1992.
3. M. A. Bednarczyk. *Categories of asynchronous systems*. PhD thesis, University of Sussex, 1988.
4. R. Brown and P. J. Higgins. On the algebra of cubes. *Journal of Pure and Applied Algebra*, (21):233–260, 1981.
5. P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction of approximations of fixed points. *Principles of Programming Languages 4*, pages 238–252, 1977.

6. S. E. Crans. *Pasting Schemes for the monoidal biclosed structure on ω-cat.* PhD thesis, 1995.
7. M. Droste and R. M. Shortt. Petri nets and automata with concurrency relations – an adjunction. *Droste, Manfred and Gurevich, Yuri /(ed.) Semantics of Programming Languages and Model Theory, Gordon and Breach Science Publ.*, 1993.
8. J. Esparza. Model checking using net unfoldings. *Science of Computer Programming*, 23(2):151–195, 1994.
9. L. Fajstrup. Loops, ditopology, and deadlocks. *Mathematical Structures in Computer Science*, 2000.
10. L. Fajstrup, E. Goubault, and M. Raussen. Detecting deadlocks in concurrent systems. In *Proceedings of the 9th International Conference on Concurrency Theory, also available at http://www.dmi.ens.fr/˜goubault.* Springer-Verlag, 1998.
11. L. Fajstrup, E. Goubault, and M. Raussen. Algebraic topology and concurrency. *submitted to Theoretical Computer Science, also technical report, Aalborg University*, 1999.
12. L. Fajstrup and S. Sokolowski. Infinitely running processes with loops from a geometric view-point. *Electronic Notes in Theoretical Computer Science, Proceedings of GETCO'00*, 2000.
13. P. J. Freyd and A. Scedrov. Categories, allegories. In *North-Holland Mathematical Library*, volume 39. North-Holland, 1990.
14. P. Gabriel and M. Zisman. Calculus of fractions and homotopy theory. In *Ergebnisse der Mathematik und ihrer Grenzgebiete*, volume 35. Springer Verlag, 1967.
15. P. Gaucher. Homotopy invariants of higher-dimensional categories and concurrency in computer science (i). *Mathematical Structures in Computer Science*, to appear, 2000.
16. P. Godefroid and P. Wolper. Using partial orders for the efficient verification of deadlock freedom and safety properties. In *Proc. of the Third Workshop on Computer Aided Verification*, volume 575, pages 417–428. Springer-Verlag, Lecture Notes in Computer Science, July 1991.
17. P. Godefroid and P. Wolper. Partial-order methods for temporal verification. In *Proc. of CONCUR'93.* Springer-Verlag, LNCS, 1993.
18. R. Goldblatt. Topoi, the categorial analysis of logic. In *Studies in logic and the foundations of mathematics*, volume 98. North-Holland, 1984.
19. E. Goubault. *The Geometry of Concurrency.* PhD thesis, Ecole Normale Supérieure, 1995. to be published, 1998, also available at http://www.dmi.ens.fr/˜goubault.
20. E. Goubault. Géométrie du parallélisme, analyse statique et autres applications. Technical report, Habilitation à diriger les recherches, Paris Dauphine, to appear, 2001.
21. E. Goubault and T. P. Jensen. Homology of higher-dimensional automata. In *Proc. of CONCUR'92*, Stonybrook, New York, August 1992. Springer-Verlag.
22. E. Goubault and M. Raussen. Dihomotopy as a tool in state space analysis. In *Proceedings of LATIN'02*, 2002.
23. C.A. Gunther and D.S. Scott. Semantic domains. In *Handbook of Theoretical Computer Science.* Elsevier, 1990.
24. A. Joyal, M. Nielsen, and Winskel G. Bisimulation and open maps. In *Proc. of LICS'93.* ACM Press, 1993.
25. J. Lambek and P.J. Scott. Introduction to higher-order categorical logic. In *Cambridge studies in advanced mathematics*, volume 7. Cambridge University Press, 1986.

26. S. Mac Lane. *Categories for the working mathematician.* Springer-Verlag, 1971.
27. S. Maclane and I. Moerdijk. *Sheaves in Geometry and Logic.* Springer-Verlag, 1992.
28. J. C. Mitchell and A. Scedrov. Notes on sconing and relators. In E. Boerger et al., editor, *Computer Science Logic '92, Selected Papers*, pages 352–378. Springer LNCS 702, 1993. Available by anonymous ftp from host ftp.cis.upenn.edu and the file pub/papers/scedrov/rel.dvi.
29. G. Plotkin. Domains. Technical report, Computer Science Department, Edinbourgh, 1984.
30. V. Pratt. Modeling concurrency with geometry. In *Proc. of the 18th ACM Symposium on Principles of Programming Languages.* ACM Press, 1991.
31. M. Raussen. On the classification of dipaths in geometric models for concurrency. *Mathematical Structures in Computer Science*, August 2000.
32. V. Sassone and G. L. Cattani. Higher-dimensional transition systems. In *Proceedings of LICS'96*, 1996.
33. J.P. Serre. *Homologie Singulière des Espaces Fibrés. Applications.* PhD thesis, Ecole Normale Supérieure, 1951.
34. M.W. Shields. Concurrent machines. *Computer Journal*, 28, 1985.
35. A. Valmari. A stubborn attack on state explosion. In *Proc. of CAV'90.* Springer Verlag, LNCS, 1990.
36. R. van Glabbeek. Bisimulation semantics for higher dimensional automata. Technical report, Stanford University, Manuscript available on the web as http://theory.stanford.edu/~rvg/hda, 1991.
37. G. Winskel and M. Nielsen. *Models for concurrency*, volume 3 of Handbook of Logic in Computer Science, pages 100–200. Oxford University Press, 1994.

A Proofs

Most proofs are based on a particular case of the existence of Kan extensions, taken here from [14] (Proposition 1. 3. Page 22):

Proposition 3. *Let $\mathcal{C}$ be a category with direct limits and $G : \text{Set}^{\mathcal{D}^{op}} \rightarrow \mathcal{C}$ a functor. Then the following statements are equivalent :*

(i) G commutes with direct limits.
(ii) G is left adjoint to a functor $D : \mathcal{C} \rightarrow \text{Set}^{\mathcal{D}^{op}}$. Moreover, the functor $G \rightarrow G \circ h^{\mathcal{D}}$ is an equivalence of the full subcategory of $Hom(\text{Set}^{\mathcal{D}^{op}}, \mathcal{C})$ formed by the functors G which commute with direct limits on $Hom(\mathcal{D}, \mathcal{C})$.

In fact, D is the functor which associates $h^{\mathcal{C}}(c) \circ G \circ h^{\mathcal{D}}$ with $c \in \mathcal{C}$.

Lemma 1 is obvious.
Lemma 2:

Proof. It suffices to use Proposition 3 with $\mathcal{D} = \square^S$, $\mathcal{C} = \text{Set}^{\square^{op}}$ and functor $w \in Hom(\mathcal{D}, \mathcal{C})$ with $w([p]) = h^{\square}([p])$. This defines F and its right-adjoint K. It is easy to see that the unit η of the adjunction is in fact the identity natural transformation $\eta : Id \rightarrow K \circ F$. This means that K induces an equivalence of categories between $F(\Upsilon^S)$ and Υ^S.

The case of cubical sets of dimension less or equal than n is treated in exactly the same manner.

Theorem 1:

Proof. We now forget about the given labelling set E, even in the definition of transition systems and labelled cubical sets. Thus, given a transition system $T = (S, i, Tran) \in TS$ we have, $\mathcal{U}(T) = N$ where $N = F(M, l, j)$ with,

- $M_0 = S$,
- $M_1 = \{a_{s,s'} \mid a \in E, s \xrightarrow{a} s' \in Tran\}$,
- $\partial_0^0(a_{s,s'}) = s$, $\partial_0^1(a_{s,s'}) = s'$,
- $l(a_{s,s'}) = a$, $l(s) = 1$.

Therefore, $\mathcal{V}(\mathcal{U}(T)) = (S', i', Tran')$ with,

- $S' = N_0 = M_0 = S$,
- $i' = j = i$,
- $s \xrightarrow{a} s' \in Tran'$ if $\exists x \in N_1$, such that $l(x) = a$, $\partial_0^0(x) = s$ and $\partial_0^1(x) = s'$. The only possible $x \in N_1$ such that $l(x) = a \in E$ is actually $x \in M_1$, and the only possible x satisfying all the conditions above is $a_{s,s'}$. Therefore, $s \xrightarrow{a} s' \in Tran'$ if and only if $s \xrightarrow{a} s' \in Tran$, hence $Tran' = Tran$.

Now, take $(M, l, j) \in HTS_1$, then $(S, i, Tran) = \mathcal{V}(M, l, j)$ with,

- $S = M_0$,
- $j = i$,

- $A = E_1 \backslash \text{Im } \epsilon_0$,
- $s \xrightarrow{a} s' \in Tran$ if $\exists x \in M_1$, such that $l(x) = a$, $\partial_0^0(x) = s$ and $\partial_0^1(x) = s'$.

And then, $F(M', l', j') = \mathcal{U}(S, i, Tran)$ with,

- $M'_0 = S = M_0$,
- $M'_1 = \{a_{s,s'} \mid a \in A, s \xrightarrow{a} s' \in Tran\} = M_1 \backslash \text{Im } \epsilon_0$ (because l is free),
- $j' = i = j$,
- $\partial_0^0(a_{s,s'}) = s$, $\partial_0^1(a_{s,s'}) = s'$,
- $l(a_{s,s'}) = a$, $l(s) = 1$.

Therefore, $F(M', l', j') = (M, l, j)$ because M and l are free.

This proof extends readily on morphisms: Let first $f : (l_0 : M_0 \to E_0, i_0) \to (l_1 : M_1 \to E_1, i_1)$ be a morphism of HTS, $f = (f_1, f_2)$. Then let $(\sigma, \lambda) = \mathcal{V}(f) : (S_0, i_0, Tran_0) \to (S_1, i_1, Tran_1)$. We have:

- $\sigma(s) = f^1(s)$ (for all s state of $\mathcal{V}(l_0 : M_0 \to E_0, i_0)$),
- $\lambda(a) = \left\{ \begin{array}{cc} f^2(a) & \text{if } f^2(a) \notin \text{Im } \epsilon_0 \\ * & \text{otherwise} \end{array} \right.$ (for all a label in $\mathcal{V}(l_0 : M_0 \to E_0, i_0)$)

Let now $(g_1, g_2) = \mathcal{U}(\sigma, \lambda)$. We have,

- $\mathcal{U}(\sigma, \lambda)^1(a_{s,s'}) = \left\{ \begin{array}{cc} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } \lambda(a) \neq * \\ \epsilon_0(\sigma(s)) & \text{otherwise} \end{array} \right.$,
- $\mathcal{U}(\sigma, \lambda)^1(s) = \sigma(s)$ $(s \in M_0)$,
- $\mathcal{U}(\sigma, \lambda)^2(a_{s,s'}) = \left\{ \begin{array}{cc} \lambda(a) & \text{if } \lambda(a) \neq * \\ \epsilon_0(1) & \text{otherwise} \end{array} \right.$,
- $\mathcal{U}(\sigma, \lambda)^2(s) = 1$ $(s \in M_0)$.

But,

- $\lambda(a)_{\sigma(s),\sigma(s')}$ is the unique x (because of the determinism condition in HTS_1) going from $\sigma(s) = f^1(s)$ to $\sigma(s') = f^1(s')$, with label $\lambda(a) = f^2(a)$, hence is equal to $f^1(a_{s,s'})$,
- when $\lambda(a) = *$, i.e. when $f^2(a) \in \text{Im } \epsilon_0$, $f^1(s)$ is necessarily in $\text{Im } \epsilon_0$: (f^1, f^2) being a morphism between l_0 and l_1, we have $l_1(f^1(a_{s,s'})) = f^2(l_0(a_{s,s'})) = f^2(a) \in \text{Im } \epsilon_0$; In order to have this, it is necessary that $f^1(a_{s,s'}) \in \text{Im } \epsilon_0$. Furthermore, $\partial_0^0(f^1(a_{s,s'})) = f^1(s) = \sigma(s)$ and $\partial_0^1(f^1(a_{s,s'})) = \sigma(s') = \sigma(s)$ therefore $f^1(a_{s,s'}) = \epsilon_0(\sigma(s)) = \mathcal{U}(\sigma, \lambda)^1(a_{s,s'})$.

Now let $f = (\sigma, \lambda) : (S_0, i_0, Tran_0) \to (S_1, i_1, Tran_1)$ be a morphism of labelled transition system and $g = \mathcal{U}(f)$. We have,

- $\mathcal{U}(\sigma, \lambda)^1(a_{s,s'}) = \left\{ \begin{array}{cc} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } \lambda(a) \neq * \\ \epsilon_0(\sigma(s)) & \text{otherwise} \end{array} \right.$,
- $\mathcal{U}(\sigma, \lambda)^1(s) = \sigma(s)$ $(s \in M_0)$,
- $\mathcal{U}(\sigma, \lambda)^2(a_{s,s'}) = \left\{ \begin{array}{cc} \lambda(a) & \text{if } \lambda(a) \neq * \\ \epsilon_0(1) & \text{otherwise} \end{array} \right.$,
- $\mathcal{U}(\sigma, \lambda)^2(s) = 1$ $(s \in M_0)$.

Then consider $f' = (\sigma', \lambda') = \mathcal{V}(g)$. We have,

– $\sigma'(s) = g_1(s)$ (for all s state),
– $\lambda'(a) = \begin{cases} g_2(a) & \text{if } g_2(a) \notin \text{Im } \epsilon_0 \\ * & \text{otherwise} \end{cases}$ (for all a label)

Therefore,

– if $g_2(a) \notin \text{Im } \epsilon_0$, i.e. if $\lambda(a) \neq *$, then $\lambda'(a) = g_2(a) = \lambda(a)$. If not, $\lambda'(a) = *$ and $\lambda(a) = *$ at the same time.
– $\sigma'(s) = g_1(s) = \sigma(s)$.

Lemma 3:

Proof. Let $(F, G) \in S_{\mathcal{C},\mathcal{D}}$, $l \in Hom_{\mathcal{C}}(M, N)$ and $l' \in Hom_{\mathcal{D}}(M', N')$. Let now $f \in Hom_{(Id_{\mathcal{D}} \downarrow Id_{\mathcal{D}})}(F(l), l')$; this means that $f = (f_1, f_2)$ where f_1 and f_2 are morphisms in $\mathcal{D}$ which make the following diagram commutative:

$$\begin{array}{ccc} F(M) & \xrightarrow{f_1} & M' \\ {\scriptstyle F(l)}\downarrow & & \downarrow{\scriptstyle l'} \\ F(N) & \xrightarrow[f_2]{} & N' \end{array}$$

So the following diagram is also commutative by functoriality of G:

$$\begin{array}{ccc} G \circ F(M) & \xrightarrow{G(f_1)} & G(M') \\ {\scriptstyle G \circ F(l)}\downarrow & & \downarrow{\scriptstyle G(l')} \\ G \circ F(N) & \xrightarrow[G(f_2)]{} & G(N') \end{array}$$

But the unit η of the adjunction between F and G is a natural transformation, thus the first square of the following diagram also commutes, entailing that the outer square itself is a commutative one:

$$\begin{array}{ccccc} M & \xrightarrow{\eta_M} & G \circ F(M) & \xrightarrow{G(f_1)} & G(M') \\ {\scriptstyle l}\downarrow & & {\scriptstyle G \circ F(l)}\downarrow & & \downarrow{\scriptstyle G(l')} \\ N & \xrightarrow[\eta_N]{} & G \circ F(N) & \xrightarrow[G(f_2)]{} & G(N') \end{array}$$

Hence we get naturally, a morphism in $Hom_{(Id_{\mathcal{C}} \downarrow Id_{\mathcal{C}})}(l, G(l'))$:

$$A_{l,l'}(f_1, f_2) = (G(f_1) \circ \eta_M, G(f_2) \circ \eta_N)$$

Similarly in the other direction, we get a morphism in $Hom_{(Id_{\mathcal{D}} \downarrow Id_{\mathcal{D}})}(F(l), l')$,

$$B_{l,l'}(g_1, g_2) = (\epsilon_{M'} \circ F(g_1), \epsilon_{N'} \circ F(g_2))$$

where ϵ is the co-unit of the adjunction(F, G).

We now prove that this is a natural bijection between $Hom_{(Id_{\mathcal{D}} \downarrow Id_{\mathcal{D}})}(F(l), l')$ and $Hom_{(Id_{\mathcal{C}} \downarrow Id_{\mathcal{C}})}(l, G(l'))$. The composite of $A_{l,l'}$ with $B_{l,l'}$ being the identity is a direct consequence of the (right) identity 8 page 80 of [26]:

$$F(M) \xrightarrow{F(\eta_M)} FGF(M) \xrightarrow{\epsilon_{F(M)}} F(M)$$

is the identity natural transformation on F. This means that the following diagram is commutative:

$$\begin{array}{ccccc} F(M) & \xrightarrow{F(\eta_M)} & FGF(M) & \xrightarrow{\epsilon_{F(M)}} & F(M) \\ {\scriptstyle f_1}\downarrow & & {\scriptstyle FG(f_1)}\downarrow & & {\scriptstyle f_1}\downarrow \\ M' & \xrightarrow{\eta_{M'}} & FG(M') & \xrightarrow{\epsilon_{M'}} & M' \end{array}$$

Hence,

$$F(M) \xrightarrow{F\eta_M} FGF(M) \xrightarrow{FG(f_1)} FG(M') \xrightarrow{\epsilon_{M'}} M' \quad = f_1$$

Similarly, the composite $B_{l,l'} \circ A_{l,l'} = Id$ because of (left) identity 8 page 80 of [26], so we have:

$$G(M') \xrightarrow{\eta_{M'}G} GFG(M') \xrightarrow{GF(f_2)} GF(M) \xrightarrow{\eta_M} M \quad = f_2$$

Thus (F, G) induces a pair of adjoint functors between $(Id_{\mathcal{C}} \downarrow Id_{\mathcal{C}})$ and $(Id_{\mathcal{D}} \downarrow Id_{\mathcal{D}})$.

Lemma 4:

Proof. The natural bijection between $Hom_{\mathcal{D}}(F(X), Y)$ and $Hom_{\mathcal{C}}(X, G(Y))$ naturally restricts to a bijection between $Hom_{\mathcal{D}}(F(X), Y) = Hom_{\mathcal{D}'}(F(X), Y)$ ($\mathcal{D}'$ is full in $\mathcal{D}$) and $Hom_{\mathcal{C}}(X, G(Y)) = Hom_{\mathcal{C}'}(X, G(Y))$ ($\mathcal{C}'$ is full in $\mathcal{C}$) for $X \in \mathcal{D}'$ and $Y \in \mathcal{C}'$.

Proposition 1:

Proof. Take as a first instance of Proposition 3 $\mathcal{D} = \square^{\leq n}$ and $\mathcal{C} = \text{Set}^{\square^{op}}$. We define functor $u \in Hom(\mathcal{D}, \mathcal{C})$ as follows :

$$u([p]) = h^{\square}([p])$$

Then functor G of Proposition 3 is the functor which commutes with direct limits and which is such that,

$$G(h^{\square^{\leq n}}([p])) = h^{\square}([p])$$

$\mathcal{I}_n$ of the proposition is therefore this functor G. Its right adjoint D given by the same proposition is such that (see [14]),

$$D(c) : a \to Hom_{\mathcal{C}}(G(h^{\mathcal{D}}(a)), c)$$

i.e. in our case, for $p \leq n$,

$$\begin{aligned} D(c)([p]) &= Hom_{\Upsilon}(h^{\square}([p]), c) \\ &= c([p]) \end{aligned}$$

the last equality holding because of Yoneda's lemma [26]. We recognize D as being the truncation functor.

Restricting the adjunction to the categories of cubical sets with morphisms respecting the initial states is obvious. The adjunction $(\Upsilon_*^L)_n \underset{T_n}{\overset{\mathcal{I}_n}{\rightleftarrows}} \Upsilon_*^L$ is a direct consequence of Lemma 3.

We proceed in a similar manner for the adjunction $\mathcal{I}_n^S$, T_n^S. We define again by Proposition 3 $\mathcal{I}_n^S(h^{\square^{S \leq n}}[p]) = h^{\square^S}[p]$. Notice that $h^{\square^{\leq n}}[p] = F_n(h^{S \leq n}[p])$ and $h^{\square}[p] = F(h^{\square^S}[p])$, therefore $\mathcal{I}_n(F(h^{\square^{S \leq n}}[p])) = F(\mathcal{I}_n^S(h^{\square^{S \leq n}}[p]))$, hence the commuting diagram, by taking the direct limit. The proof for the commutation of the diagram involving T_n is similar.

The last part of the proposition is by taking $\mathcal{D} = \square$, $\mathcal{C} = \mathbf{Set}^{(\square^{\leq n})^{op}}$ and functor $v \in Hom(\mathcal{D}, \mathcal{C})$ as follows,

$$v([p])([q]) = Hom_{\square}([q], [p])$$

which gives as G functor T_n. Now, its right adjoint is functor D with (for $N \in \mathrm{Set}^{(\square^{\leq n})^{op}}$ and $[p] \in \square$),

$$D(N)([p]) = Hom_{\Upsilon_n}(T_n(h^{\square}([p])), N)$$

<u>Theorem 2</u>:

Proof. The only difficulty in the first part, is to show that the action of these functions on morphisms are well-defined. For $\mathcal{Y}$, the only thing to check is that the definition in dimension 2 of the underlying precubical set is coherent. We compute first (taking the same notations as above), for $ab_{s,s',s'',u} \in \mathcal{Y}(S', i', E', I', Tran')$:

– if $\lambda(a) \neq *$ and $\lambda(b) \neq *$,

$$\begin{aligned} \partial_l^k(g_2(ab_{s,s',s'',u})) &= \partial_l^k\left(\lambda(a)\lambda(b)_{\sigma(s),\sigma(s'),\sigma(s''),\sigma(u)}\right) \\ &= \begin{cases} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k=0,\, l=0 \\ \lambda(b)_{\sigma(s),\sigma(s'')} & \text{if } k=0,\, l=1 \\ \lambda(a)_{\sigma(s''),\sigma(u)} & \text{if } k=1,\, l=0 \\ \lambda(b)_{\sigma(s'),\sigma(u)} & \text{if } k=1,\, l=1 \end{cases} \end{aligned}$$

We also have,

$$g_1(\partial_l^k(ab_{s,s',s'',u})) = \begin{cases} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k=0,\, l=0 \\ \lambda(b)_{\sigma(s),\sigma(s'')} & \text{if } k=0,\, l=1 \\ \lambda(a)_{\sigma(s''),\sigma(u)} & \text{if } k=1,\, l=0 \\ \lambda(b)_{\sigma(s'),\sigma(u)} & \text{if } k=1,\, l=1 \end{cases}$$

which are equal.

- if $\lambda(a) \neq *$ and $\lambda(b) = *$ (notice that we have then $\sigma(s'') = \sigma(s)$ and $\sigma(s') = \sigma(u)$),

$$\begin{aligned} \partial_l^k(g_2(ab_{s,s',s'',u})) &= \partial_l^k(\epsilon_0(\lambda(a)_{\sigma(s),\sigma(s')})) \\ &= \begin{cases} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k=0,\, l=0 \\ \epsilon_0(\partial_0^0(\lambda(b)_{\sigma(s),\sigma(s'')})) = \epsilon_0(\sigma(s)) & \text{if } k=0,\, l=1 \\ \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k=1,\, l=0 \\ \epsilon_0(\partial_0^1(\lambda(b)_{\sigma(s'),\sigma(u)})) = \epsilon_0(\sigma(u)) & \text{if } k=1,\, l=1 \end{cases} \end{aligned}$$

We also have,

$$g_1(\partial_l^k(ab_{s,s',s'',u})) = \begin{cases} \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k=0,\, l=0 \\ \epsilon_0(\sigma(s)) & \text{if } k=0,\, l=1 \\ \lambda(a)_{\sigma(s''),\sigma(u)} = \lambda(a)_{\sigma(s),\sigma(s')} & \text{if } k=1,\, l=0 \\ \epsilon_0(\sigma(u)) & \text{if } k=1,\, l=1 \end{cases}$$

which are equal.

- if $\lambda(b) \neq *$ and $\lambda(a) = *$ (notice then that we have $\sigma(s') = \sigma(s)$),

$$\begin{aligned} \partial_l^k(g_2(ab_{s,s',s'',u})) &= \partial_l^k(\epsilon_1(\lambda(b)_{\sigma(s),\sigma(s'')})) \\ &= \begin{cases} \epsilon_0(\sigma(s)) & \text{if } k=0,\, l=0 \\ \lambda(b)_{\sigma(s),\sigma(s'')} & \text{if } k=0,\, l=1 \\ \epsilon_0(\sigma(s'')) & \text{if } k=1,\, l=0 \\ \lambda(b)_{\sigma(s),\sigma(s'')} & \text{if } k=1,\, l=1 \end{cases} \end{aligned}$$

We also have,

$$g_1(\partial_l^k(ab_{s,s',s'',u})) = \begin{cases} g_1(a_{s,s'}) = \epsilon_0(\sigma(s)) & \text{if } k=0,\, l=0 \\ g_1(b_{s,s''}) = \lambda(b)_{\sigma(s),\sigma(s'')} & \text{if } k=0,\, l=1 \\ g_1(a_{s'',u}) = \epsilon_0(\sigma(s'')) & \text{if } k=1,\, l=0 \\ g_1(b_{s',u}) = \lambda(b)_{\sigma(s'),\sigma(u)} = \lambda(b)_{\sigma(s),\sigma(s'')} & \text{if } k=1,\, l=1 \end{cases}$$

which are equal.

- if $\lambda(a) = *$ and $\lambda(b) = *$ (notice that then we have $\sigma(s) = \sigma(s') = \sigma(s'') = \sigma(u)$),

$$\begin{aligned} \partial_l^k(g_2(ab_{s,s',s'',u})) &= \partial_l^k(\epsilon_0\epsilon_0(\sigma(s))) \\ &= \begin{cases} \epsilon_0(\sigma(s)) & \text{if } k=0,\, l=0 \\ \epsilon_0(\partial_0^0(\epsilon_0(\sigma(s)))) = \epsilon_0(\sigma(s)) & \text{if } k=0,\, l=1 \\ \epsilon_0(\sigma(s)) & \text{if } k=1,\, l=0 \\ \epsilon_0(\partial_0^1(\epsilon_0(\sigma(s)))) = \epsilon_0(\sigma(s)) & \text{if } k=1,\, l=1 \end{cases} \end{aligned}$$

We also have,

$$g_1(\partial_l^k(ab_{s,s',s'',u})) = \begin{cases} g_1(a_{s,s'}) = \epsilon_0(\sigma(s)) & \text{if } k=0,\ l=0 \\ g_1(b_{s,s''}) = \epsilon_0(\sigma(s)) & \text{if } k=0,\ l=1 \\ g_1(a_{s'',u}) = \epsilon_0(\sigma(s'')) = \epsilon_0(\sigma(s)) & \text{if } k=1,\ l=0 \\ g_1(b_{s',u}) = \epsilon_0(\sigma(s')) = \epsilon_0(\sigma(s)) & \text{if } k=1,\ l=1 \end{cases}$$

which are equal.

For $\mathcal{W}$ we have to check that, for $f = (\sigma, \lambda) = \mathcal{Y}(g : (P, l : P \to L, i) \to (P', l' : P' \to L', i'))$,

$$aI_s b \text{ and } \lambda(a) \neq *,\ \lambda(b) \neq * \text{ implies } \lambda(a) I'_{\sigma(s)} \lambda(b)$$

Suppose aI_sb in $\mathcal{Y}(P, l : P \to L, i)$. Then there exist $x, x', y, y' \in P_1$ with $l(x) = a$, $l(x') = a$, $l(y) = b$, $l(y') = b$ and $\partial_0^0(x) = \partial_0^0(y) = s$, $\partial_0^1(x) = \partial_0^0(y')$, $\partial_0^1(y) = \partial_0^0(x')$, $\partial_1^1(y') = \partial_1^1(x')$, and we have a $C \in P_2$ with $l(C) = (a, b)$, $\partial_0^0(C) = x$, $\partial_1^0(C) = y$, $\partial_0^1(C) = y'$ and $\partial_1^1(C) = x'$. We know that $g(C) \in P_2'$ and that $l' \circ g(C) = (f(a), f(b))$ since $f(a) \neq *$ and $f(b) \neq *$. Similarly, $l'(g(x)) = f(a)$, $l'(g(x')) = f(a)$, $l'(g(y)) = f(b)$, $l'(g(y')) = f(b)$. Furthermore, because g is a morphism of cubical sets, $\partial_0^0(g(x)) = \partial_0^0(g(y)) = \sigma(s)$, $\partial_0^1(g(x)) = \partial_0^0(g(y'))$, $\partial_0^1(g(y)) = \partial_0^0(g(x'))$, $\partial_1^1(g(y')) = \partial_1^1(g(x'))$, so $\lambda(a) I'_{\sigma(s)} \lambda(b)$.

It is easy to see that these functors restricted to the 1-skeleton are inverse of each other (this is the consequence of Theorem 1). Now more generally, it is easy to check that $\mathcal{W} \circ \mathcal{Y} = Id$.

Finally, for all free 2-dimensional cubical sets (from precubical sets) (P, l, j), $\mathcal{Y} \circ \mathcal{W}(P, l, j)$ is naturally equal to (P, l, j).

Proposition 2:

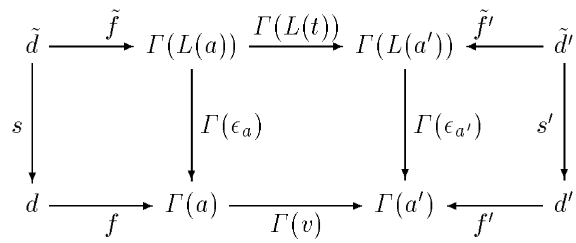
Proof. Let $(d, a, f : d \to \Gamma(a)) \in (\mathcal{D} \downarrow \Gamma)$. Define $\tilde{f}$ as in the following pullback diagram:

$$\begin{array}{ccc} \tilde{d} & \xrightarrow{\tilde{f}} & \Gamma(L(a)) \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle \Gamma(\epsilon_a)} \\ d & \xrightarrow[f]{} & \Gamma(a) \end{array}$$

and define $\tilde{L}(d, a, f : d \to \Gamma(a)) = (\tilde{d}, L(a), \tilde{f} : \tilde{d} \to \Gamma(L(a)))$. We first show that $\tilde{L}$ defines a functor from $(\mathcal{D} \downarrow \Gamma)$ to $(\mathcal{D} \downarrow \Gamma)$. Let (u, v) be a morphism from (d, a, f) to (d', a', f'). We have the following commutative diagram because of the naturality of ϵ:

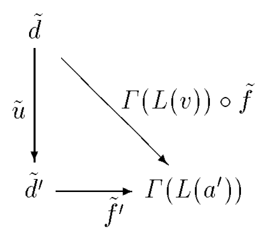
$$\begin{array}{ccc} L(a) & \xrightarrow{L(v)} & L(a') \\ {\scriptstyle \epsilon_a}\downarrow & & \downarrow{\scriptstyle \epsilon_{a'}} \\ a & \xrightarrow{v} & a' \end{array}$$

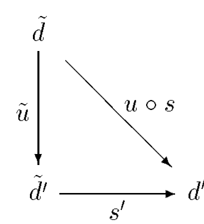
Hence applying functor Γ to last diagram and completing with the pullback diagrams defining respectively $\tilde{f}$ (on the left) and $\tilde{f'}$ (on the right), we get the following commutative diagram:



In particular, we can read off from this last diagram that we have a map $\Gamma(L(t)) \circ \tilde{f} : \tilde{d} \to \Gamma(L(a'))$. We also have a map $u \circ s : \tilde{d} \to d'$. These two maps are such that $f' \circ (u \circ s) = \Gamma(\epsilon_{a'}) \circ (\Gamma(L(t)) \circ \tilde{f})$ since by the left two commutative squares of the diagram above, we have $\Gamma(\epsilon_{a'}) \circ (\Gamma(L(v)) \circ \tilde{f}) = \Gamma(v) \circ f \circ s$ and because (u, v) is a morphism between (d, a, f) and (d', a', f') in $(\mathcal{D} \downarrow \Gamma)$, we have $f' \circ u = \Gamma(v) \circ f$ so $\Gamma(v) \circ f \circ s = f' \circ u \circ s$.

Therefore, by the universal property of pullbacks, applied to $\tilde{d'}$ being the pullback of f' and $\Gamma\epsilon_{a'}$, there is necessarily a unique $\tilde{u} : d \to d'$ such that the following diagrams are commutative:

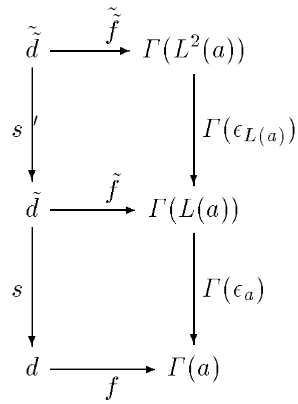




The diagram at the left precisely means that $(\tilde{u}, L(v))$ is a morphism from $\tilde{L}(d, a, f)$ to $\tilde{L}(d', a', f')$ in $\mathcal{D} \downarrow \Gamma$.

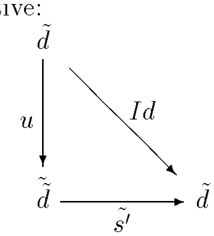
Now we define $\tilde{\epsilon} : \tilde{L} \to Id$. In fact, the pullback diagram defining $\tilde{L}(d, a, f)$ precisely defines a morphism (s, ϵ_a) from $\tilde{L}(d, a, f)$ to (d, a, f) in $\mathcal{D} \downarrow \Gamma$, which is natural (because it is the pullback diagram!) in a.

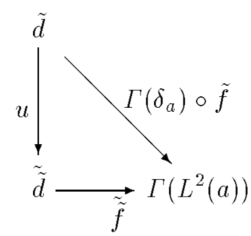
The definition of $\tilde{\delta} : \tilde{L} \to \tilde{L}^2$ is more intricate. The following commutative diagram is the concatenation of the two pullback diagrams, the topmost one defining $\tilde{L}^2(d,a,f)$ and the other one defining $\tilde{L}(d,a,f)$:



But we also have a map $\Gamma(\delta_a) \circ \tilde{f} : d' \to \Gamma(L^2(a))$ and a map $Id : d' \to d'$. Notice that these two maps are such that $\Gamma(\epsilon_{L(a)}) \circ \Gamma(\delta_a) \circ \tilde{f} = \tilde{f}$. This is precisely due to the "co-unit" equation of the comonad (L, ϵ, δ) which implies that $\Gamma(\epsilon_{L(a)}) \circ \Gamma(\delta_a) = \Gamma(\epsilon_{L(a)} \circ \delta_a) = \Gamma(Id) = Id$.

Thus, by the universal property of pullbacks, applied to $\tilde{\tilde{d}}$, we necessarily have a unique morphism $u : \tilde{d} \to \tilde{\tilde{d}}$ such that the following two diagrams are commutative:





This last right diagram shows precisely that (u, δ_a) is a morphism from $\tilde{L}(d,a,f)$ to $\tilde{L}^2(d,a,f)$ which is natural in a because it is given as the unique solution of a universal problem (the pullback diagram).

It is easy to see that the "co-associativity" of $\tilde{\delta}$ is a direct consequence of the "co-associativity" of δ, and similarly that $\tilde{\epsilon}$ is the "co-unit" for $\tilde{\delta}$.

Definition and Lemma 3:

Lemma 5. *(f, d, s) is a comonad.*

Proof. First, we have to show that for all $E \in \mathrm{Ord}$, $d_{f(E)} \circ s_E = f(d_E) \circ s_E$ meaning that the "co-multiplication" s with the "co-unit" d is the "co-unit".

Let $F = \{x_1 < \cdots < x_k\}$ $(k \geq 1)$ be a linearly ordered non-empty and finite subset of E, i.e. an element of $f(E)$. We have

$$s_E(F) = \{\{x_1\} \subseteq \{x_1 < x_2\} \subseteq \cdots \{x_1 < \cdots x_k\}\} \in f(f(E))$$

so $d_{f(E)}(s_E(F)) = \{x_1 < \cdots x_k\} = F$.

Now, $f(d_E) : f^2(E) \to f(E)$ associates to each linearly ordered finite and non-empty sequence $\{P_1 \subseteq \cdots P_k\}$ of linearly ordered finite and non-empty subsets P_i $(i = 1, \cdots, k)$ of E, the linearly ordered finiteand non-empty subset of E $f(d_E)(\{P_1 \subseteq \cdots P_k\}) = \{\sup P_1 < \cdots \sup P_k\}$. Therefore,

$$\begin{aligned} f(d_E)(s_E(\{x_1 < \cdots x_k\})) &= f(d_E)(\{\{x_1\} \subseteq \{x_1 < x_2\} \subseteq \cdots \{x_1 < \cdots x_k\}\}) \\ &= \{\sup(\{x_1\}) < \cdots \sup(\{x_1 < \cdots x_k\})\} \\ &= \{x_1 < \cdots x_k\} \\ &= F \\ &= d_{f(E)}(s_E(F)) \end{aligned}$$

The second identity we have to show is that the "co-multiplication" s is associative, i.e. for all $E \in \text{Ord}$,

$$s_{f(E)} \circ s_E = f(s_E) \circ s_E$$

Let $F = \{x_1 < \cdots < x_k\}$ $(k \geq 1)$ be an element of $f(E)$. We have

$$s_E(F) = \{\{x_1\} \subseteq \{x_1 < x_2\} \subseteq \cdots, \{x_1 < \cdots x_k\}\} \in f(f(E))$$

and also,

$$s_{f(E)}(s_E(F)) = \{\{\{x_1\}\} \subseteq \{\{x_1\} \subseteq \{x_1 < x_2\}\} \subseteq \cdots \{\{x_1\} \subseteq \cdots \{x_1 < \cdots x_k\}\}\}$$

Now, given $A = \{P_1 \subseteq \cdots P_k\} \in f^2(E)$ (where $P_i \in f(E)$, for $i = 1, \cdots, n$), we have,

$$f(s_E(A)) = \{s_E(P_1) \subseteq \cdots s_E(P_k)\}$$

therefore,

$$\begin{aligned} f(s_E)(s_E(F)) &= \{s_E(\{x_1\}) \subseteq \cdots s_E(\{x_1 < \cdots x_k\})\} \\ &= \{\{\{x_1\}\} \subseteq \{\{x_1\} \subseteq \{x_1 < x_2\}\} \subseteq \cdots \{\{x_1\} \subseteq \cdots \{x_1 < \cdots x_k\}\}\} \\ &= s_{f(E)}(s_E(F)) \end{aligned}$$