Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 357 (2006) 241 – 278
Theoretical Computer Science
www.elsevier.com/locate/tcs

Algebraic topology and concurrency

Lisbeth Fajstrup[a,*], Martin Raußen[a], Eric Goubault[b]

[a] *Department of Mathematical Sciences, Aalborg University, Fredrik Bajers Vej 7G, DK-9220 Aalborg Ø, Denmark*
[b] *LIST (CEA-Technologies Avancées), DTSI-SLA, CEA F91191 Gif-sur-Yvette Cedex, France*

Abstract

We show in this article that some concepts from homotopy theory, in algebraic topology, are relevant for studying concurrent programs. We exhibit a natural semantics of semaphore programs, based on partially ordered topological spaces, which are studied up to "elastic deformation" or homotopy, giving information about important properties of the program, such as deadlocks, unreachables, serializability, essential schedules, etc. In fact, it is not quite ordinary homotopy that has to be used, but rather a "directed homotopy" that does not reverse the flow of time. We show some of the essential differences between ordinary and directed homotopy through examples. We also relate the topological view to a combinatorial view of concurrent programs closer to transition systems, through the notion of a cubical set. Finally we apply some of these concepts to the proof of the safeness of a two-phase protocol, well-known and used in concurrent database theory. We end up with a list of problems from both a mathematical and a computer-scientific point of view.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Homotopy; Concurrency; Cubical set; Higher-dimensional automata; Partial order; Partially ordered space

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 twofold: 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 a partial ordering describing the time flow. This gives rise to new interesting mathematical problems and it provides common grounds to computer-scientific problems that have not been precisely related otherwise in the past.

The organization of the paper is as follows.

We first explain the interest of using geometric ideas for semantical reasons, in Section 2, pointing to some examples that will be given throughout the paper, and in Section 10.

In Section 3, we give the first few definitions needed for modelling the topological spaces already arising in Section 2. Basically, we need to define a topological space containing all traces of executions of the concurrent systems we want to

* Corresponding author.
E-mail addresses: fajstrup@math.aau.dk (L. Fajstrup), raussen@math.aau.dk (M. Raußen), Eric.Goubault@cea.fr (E. Goubault).

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.03.022

characterize plus the information about how time flows. This is the main difference from 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. In Section 6, it will be pointed out that cubical sets (or higher-dimensional automata, [34,65]) give rise to such spaces in a natural way, hence most "combinatorial" concurrency models are instances of these local po-spaces. It is worth noting that some models in general relativity [64] consider timed spaces, and the authors were inspired by these physical concepts when developing the theory.

Section 4 gives the first definitions of the new homotopy theory we need in order to define equivalence of paths in accordance with the intuitive explanations from 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. We give examples that show that this directed homotopy is finer than usual homotopy in the sense that it can distinguish homotopy equivalent (in the standard sense) topological spaces.

We study in Section 5 a particular subcategory of local po-spaces: those which are locally Euclidean, i.e. the local partial order is inherited from $\mathbb{R}^n$ (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 in Section 6 by investigating cubical sets (or higher-dimensional automata, HDA) and show that they are in some sense a combinatorial counterpart of these local po-spaces (at least, of a large subcategory). We refer the reader to [34] or to the more recent [21] for actual semantics of some concurrent systems using these cubical sets. A "combinatorial" deformation theory in cubical sets is developed and related to the directed homotopy in the continuous case, using in particular the notion of subdivision, in Section 6.8.

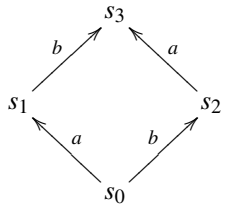
A major application, serializability in concurrent databases, is fully treated in Section 7. It is an application of the preceding theory and a refinement and extension of the result in [48]. There are also similar problems in scheduling microinstructions in chips, as will be briefly explained in Section 10.

Some mathematical directions are given in Section 8, and related computer-scientific perspectives are listed in Section 9. Finally we refer the reader to related work where algebraic topology is at the centre of computer-scientific modelling and proofs, and to application areas which should be examined in detail in future work, in Section 10.

Part of this was presented by two of the co-authors at the 14th conference on the Mathematical Foundations of Programming Semantics (London, May 1998). It has taken a long time to assemble this full version of our work, and the development has gone on since then. This is why we have to give some forward references, some of them based on earlier versions of this paper, both as footnotes and in the concluding sections 8–10.

2. Motivation and examples of applications

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 they 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 as in the figure below:



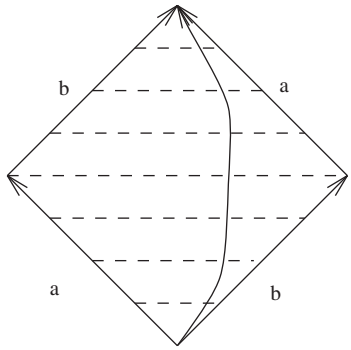


Fig. 1. New traces.

Let us take a closer look at the geometry of transition systems used in concurrency. 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 as 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 Fig. 1, 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.

Here we are confronted with two presentations of essentially the same phenomenon. The first one is the geometry of continuous paths (like in the study of mechanics). The second is a discretization of it, which in general comes first in the semantics applications (but not in others; see Section 9). Basically, abstract all paths inside an 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 (for describing, in a similar way as above, k actions executing asynchronously), as shown in Fig. 10. This is precisely what is called a cubical set in combinatorial algebraic topology (see Section 6) or a HDA [65] in computer science. But here, we are considering cubical sets with an orientation given by the time flow (by orienting the segments constituting it), whereas in ordinary algebraic topology, shapes have no preferred order. So the continuous counterpart of such a discretization is more than a topological space, it contains also (partial) order relations. This is developed in Section 3 under the name of po-spaces (and lpo-spaces) whose formalization and understanding is the main objective of this article.

The reader might wonder why we go from a typically discrete world (transition systems for instance) to a continuous world (topological subspaces of $\mathbb{R}^n$ for instance), especially when trying to tackle the state-space explosion problem! In some situations, going back and forth between discrete and continuous helps a lot: continuous situations are much easier to "deform and retract elastically" whereas it is combinatorially very demanding and less intuitive to do the same in a discrete world. On the converse, discrete situations make some arguments simpler, that otherwise have to be abstracted by hard topological properties. Even when the continuous image is questionable, since there is no immediate interpretation of infinitesimals for instance, it can help a lot, as in the case of quantum mechanics. Moreover, continuous models have a better chance of giving models to real-time (and hybrid) parallel systems, for this see for instance [36] (slightly outdated) or better, [17].

Last but not least, this way of looking at concurrent processes relies on well-established theories, which we have to transform, but still, quite a lot of the technical difficulties are handled by constructions in these theories (homotopy and homology theory). Certainly, the geometric intuition helped a lot in understanding essential phenomena of concurrency theory: for instance, there are new examples/counter-examples (Examples 4.4 (4) or 5.11) that could not have been obtained without these geometric ideas.

3. Partially ordered spaces and local po-spaces

The geometric model which is already implicitly used in the pictures in Section 2 is a topological space with a (time) direction. For codes without loops, the right model is a po-space, a topological space with a partial order Definition 3.1. When there are loops involved, we cannot get a global partial order: the notion of clockwise order on the circle is not a partial order—all points are both before and after all other points. Hence we need a *locally partially ordered space* or an *lpo-space*, cf. Definition 3.4.

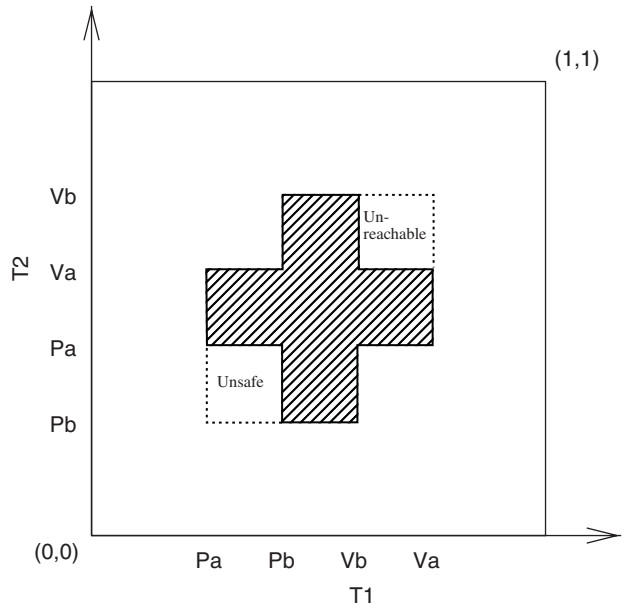


Fig. 2. The Swiss flag example—two processes sharing two resources.

In this geometric model, we can now give geometric characterizations of an *execution*, a *deadlock*, an *unreachable point* etc. As a class of examples throughout this text, we use a toy language manipulating semaphores. Using Dijkstra's notation [15]. Processes are sequences of locking operations *Pa* on semaphores *a* and unlocking operations *Va*. For the time being we consider only binary semaphores, ensuring mutual exclusion of accesses, but in further examples, we will also model and use counting semaphores, or k-semaphores ($k > 1$) which can be accessed concurrently by up to k processes.

In the example where two processes share two resources a and b:

$$T1 = Pa.Pb.Vb.Va,$$

$$T2 = Pb.Pa.Va.Vb$$

the geometric model is the "Swiss flag", Fig. 2. This is a subset of $\mathbb{R}^2$ with partial order $(x_1, y_1) \leqslant (x_2, y_2)$ if $x_1 \leqslant y_1$ and $x_2 \leqslant y_2$. As a matter of fact, the (interior of the) horizontal dashed rectangle comprises global states that are such that T_1 and T_2 would hold a lock on a: this is impossible by the very definition of a binary semaphore. Similarly, the (interior of the) vertical rectangle is a path from the initial point consists of states violating the mutual exclusion property on b. Therefore both dashed rectangles are *forbidden regions*, i.e. are not part of the space of (legal) states. This provides us with a particular po-space, as defined below. The idea is that a po-space is a topological space in which points are ordered globally through time.

Definition 3.1. (1) A *partial order* $\leqslant$ on a set U is a reflexive, transitive and antisymmetric relation. We write $x < y$ for ($x \leqslant y$ and $x \neq y$).

(2) A partial order $\leqslant$ on a topological space X is called *closed* if $\leqslant$ is a closed subset of $X \times X$ in the product topology. In that case, $(X, \leqslant)$ is called a *pospace*.

Example 3.2. The partial order on $\mathbb{R}^2$ used in the Swiss flag example above generalizes to $\mathbb{R}^n$: $(x_1, \ldots, x_n) \leqslant (y_1, \ldots, y_n)$ if $x_i \leqslant y_i$ for all $i = 1, \ldots, n$.

Remark 3.3. Let $(X, \leqslant)$ denote a pospace.

(1) For every $x \in X$, the sets $\downarrow x = \{y \in X | y \leqslant x\}$ and $\uparrow x = \{y \in X | y \geqslant x\}$ are closed.

(2) For every pair of points $y_1, y_2 \in X$, the set $[y_1, y_2] = \{x \in X | y_1 \leqslant x \leqslant y_2\} = \downarrow y_2 \cap \uparrow y_1$ is closed.

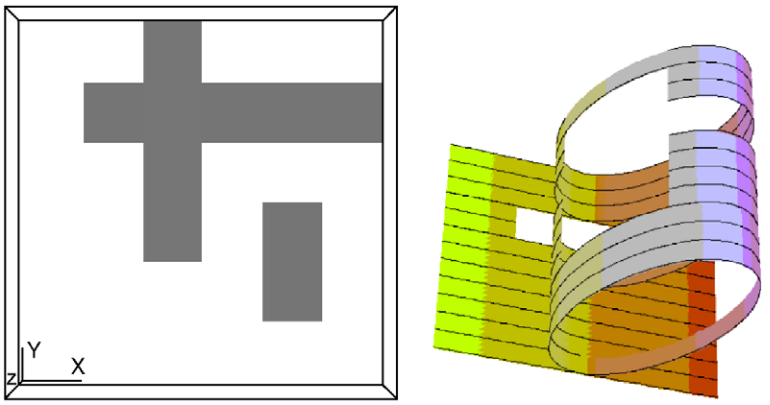


Fig. 3. Folding of a po-space to give semantics to a loop.

(3) A partially ordered topological space is a pospace if and only if whenever $a \not\leqslant b$, there exist open sets U and V with $a \in U$ and $b \in V$ such that for all $x \in U$ and $y \in V$ $x \not\leqslant y$. Hence a pospace is Hausdorff (see [31,63]).

When there are loops involved, having a partial order is too strong an assumption. This fact will lead us to local po-spaces (Definition 3.4). For instance, consider the following two processes in parallel:

```
A=Pa.(Pb.Vb.Pc.Vc)*.Va
B=Pc.Pb.Vc.Pa.Va.Vb
PROG=A|B
```

`A` can loop any number of times after `Pa`. As a topological space of states, this consists in folding the po-space on the left of Fig. 3 so that the start time of the left parenthesis is identified with the final time of execution of the right parenthesis, giving the topological space on the right of the same figure. Notice that this topological space cannot possibly be globally ordered, but is of course locally ordered as in the following definition (expressing the flow of time, locally).

Definition 3.4. Let X be a topological space.

- A collection $\mathcal{U}(X)$ of pairs $(U, \leqslant_U)$ with partially ordered open subsets U covering X is a *local partial order* on X if for every $x \in X$ there is a non-empty open neighbourhood $W(x) \subset X$ with a partial order $\leqslant_{W(x)}$ such that the restrictions of $\leqslant_U$ and $\leqslant_{W(x)}$ to $U \cap W(x)$ coincide for all $U \in \mathcal{U}(X)$ with $x \in U$, i.e.,

$$y \leqslant_U z \iff y \leqslant_{W(x)} z \quad \text{for all } U \in \mathcal{U}(X) \text{ such that } x \in U \text{ and for all } y, z \in W(x) \cap U. \tag{1}$$

- Two local partial orders on X are equivalent if their *union* is a local partial order.
- A topological space X together with an equivalence class of local partial orders is called a *locally partially ordered space* and a neighbourhood $W(x)$ as above is called a *partial order neighbourhood*. If, moreover, X is Hausdorff and there is a covering $\mathcal{U}$ such that for all $(U, \leqslant_U) \in \mathcal{U}$ the order $\leqslant_U$ on U is a closed relation ($(U, \leqslant_U)$ is a pospace), then X together with an equivalence class of coverings by po-spaces is a *local po-space*, or *lpo-space*.

When X is a local po-space, a neighbourhood $W(x)$ as in Definition 3.4, s.t. the partial order on $W(x)$ is closed, is called a *po-neighbourhood* of x.

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 3.5. (1) The partial orders $\leqslant_U$ need not coincide on their intersection, but on a sufficiently small neighbourhood, $W(x)$ of each point, all $\leqslant_U$ where $x \in U$ agree.

(2) This is a sort of *germ* or *sheaf* type definition of a local partial order and in particular of the monotone functions (below).

(3) 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 circle (3.6) with local partial order given by a chosen direction, then the hull of the relation is the trivial relation $x \leqslant y$ for any pair x and y. The same is true for the torus.

(4) By an abuse of notation, we will henceforth denote a locally partially ordered space $(X, \mathcal{U})$ without the $\leqslant_U$.

(5) The equivalence of local partial orders is an equivalence relation. To prove transitivity, suppose $\mathcal{U}$ and $\mathcal{V}$ are equivalent and that $\mathcal{V}$ and $\mathcal{T}$ are equivalent and let $x \in X$. Now let $(W_1(x), \leqslant_{W1})$ be a neighbourhood of x such that whenever $x, y, z \in W_1(x) \cap R$, where $R \in \mathcal{U} \bigcup \mathcal{V}$, $y \leqslant_R z \Leftrightarrow y \leqslant_{W1} z$ and let $W_2(x)$ have the analogous properties for $\mathcal{V} \bigcup \mathcal{T}$. Suppose moreover wlog, that there are $U \in \mathcal{U}$, $V \in \mathcal{V}$ and $T \in \mathcal{T}$ such that $W_1(x) \in U \cap V$ and $W_2(x) \in V \cap T$. Then the partial orders $\leqslant_{W1}$ and $\leqslant_{W2}$ agree on $W_1(x) \cap W_2(x)$, since for $y, z \in W_1(x) \cap W_2(x)$, $y \leqslant_{W1} z \Leftrightarrow y \leqslant_V z \Leftrightarrow y \leqslant_{W2} z$. The neighbourhood $W_1(x) \cap W_2(x)$ with this partial order satisfies the definition above for $\mathcal{U} \cup \mathcal{V} \cup \mathcal{T}$.

(6) There are various constructions of local po-spaces from other local po-spaces—glueing is used in the construction in Section 6. Another example is the product: given two local po-spaces, the product inherits a local po-structure from the product of the coverings—we leave the details to the reader.

(7) In an earlier version of this paper, we defined local partial orders in a slightly different way. A covering $\mathcal{U}$ as above was called a local partial order, if any x has a neighbourhood $W(x)$ (*with no ordering required*) such that whenever $U, V \in \mathcal{U}$ and $y, z \in W(x) \cap U \cap V$, $y \leqslant_U z \Leftrightarrow y \leqslant_V z$.

Since one may assume that $W(x) \subset U$ for some $U \in \mathcal{U}$, and hence the restriction of $\leqslant_U$ gives $\leqslant_{W(x)}$ which agrees with the order on all $V \cap W(x)$. Hence the fact that we require an ordering on $W(x)$ in the new definition does not make any difference. The difference is, that in the new definition, we only ask that elements from the cover which are *neighbourhoods of x* have partial orders which agree on $W(x)$. In the old definition, we required $W(x)$ to be a po-neighbourhood of all its points, not just x.

The cover $U_1 = \{e^{i\theta} \in S^1 | 0 < \theta < 2\pi\}$ and $U_2 = \{e^{i\theta} \in S^1 | \pi < \theta < 3\pi\}$ ordered by increasing θ gives a local partial order on the circle $S^1 = \{e^{i\theta} \in \mathbb{C}\}$ with the new definition, but not with the old one. There is no neighbourhood around $x = (1, 0)$ where the two orders $\leqslant_{U_1}$ and $\leqslant_{U_2}$ agree. But since only U_1 is a neighbourhood of $(1, 0)$; for the new definition, we could let $W((1, 0)) = U_1$.

In Example 3.6, we give another covering of the circle, which is a local partial order with both definitions.

Given a covering satisfying the new definition, one could ask whether there is an equivalent covering, which satisfies the old definition, ("shrinking" the elements in the cover, as in the circle example) but we did not go into this. For the local partial order which we introduce for cubical sets in Section 6 the answer is yes. After a subdivision, the local partial order given by the cover of (new and smaller) stars of vertices does in fact satisfy the old definition.

Example 3.6. (1) The circle $S^1 = \{e^{i\theta} \in \mathbb{C}\}$ has a local partial order: $U_1 = \{e^{i\theta} \in S^1 | \pi/4 < \theta < 7\pi/4\}$ has a (partial) order given by the order of the θ and $U_2 = \{e^{i\theta} \in S^1 | 5\pi/4 < \theta < 11\pi/4\}$ 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 local 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 four copies of I^2 have the partial order induced from $\mathbb{R}^2$ and no further relations.

The geometric counterpart of an execution is a path from $(0, 0)$ which is increasing in both coordinates. An execution of the whole program is an increasing path from $(0, 0)$ to the final point in the upper right hand corner. Just looking

at the picture if Fig. 2, it is clear that:
- The point at $(2/5, 2/5)$ (which corresponds to (P_b, P_a)) is a deadlock—there are no executions from there. Everything stops.
- Once an execution path has entered the area marked "unsafe", there is no way of continuing to the final point.
- The points marked "unreachable" will never be reached by an execution path initiating in $(0, 0)$

We give a formal definition of these concepts here.

Definition 3.7. (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 for any $x \in X$ there are partial order neighbourhoods $W(x)$ and $W(f(x))$ such that

$$x_1 \leqslant_{W(x)} x_2 \Rightarrow f(x_1) \leqslant_{W(f(x))} f(x_2)$$

whenever $x_1, x_2 \in f^{-1}(W(f(x))) \cap W(x)$.

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

The notion of dipath corresponds to the intuitive idea of traces of executions as explained in Figs. 1 and 2 for instance. Of course, there is a major algebraic structure on the set of dipaths, which is concatenation:

Remark 3.8. (1) Let $f_1, f_2 : I \to X$ denote two dipaths with $f_1(1) = f_2(0)$. Their *concatenation* $f_1 * f_2$ given by

$$(f_1 * f_2)(t) = \begin{cases} f_1(t), & t \leqslant 0.5; \\ f_2(2t-1), & t \geqslant 0.5. \end{cases}$$

is again a dipath.

(2) One might look at dimaps from *arbitrary* intervals and allow equivalence classes with respect to strictly increasing homeomorphisms between intervals.

Definition 3.9 (*Compare Penrose [64]*). 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 3.10. *If X has a global partial order, $\leqslant$ the relation $\prec$ is a new partial order.*

Proof. The relation $\prec$ is coarser than the relation $\leqslant$, i.e., $x \prec y \Rightarrow x \leqslant y$. Hence, $\prec$ is antisymmetric. Concatenation of dipaths shows the transitivity of $\prec$. □

This is a geometric "reachability" relation which underlies most proofs in semantics.

Remark 3.11. If X is locally partially ordered, the relation $\prec$ is still transitive, but it is not necessarily antisymmetric as one can see from Example 3.6 with the oriented circle. Moreover, the relation $\prec$ may be closed while $\leqslant$ is not and vice versa. See [18] for examples of this.

Definition 3.12. Let $y \in X$, $S \subset X$.

(1) The set $J^+(y) := \{x \in X | y \prec x\}$ 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 3.13. 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 successful outcomes of the computations.

4. Dihomotopy and diconnected components

The objective of this section is twofold: first of all, we want to formalize the concept of *deformation* of a path used in the Introduction. Technically, we speak of dihomotopies between dipaths in a given lpo-space. Secondly, we describe how the dihomotopy concept can be used to split a po-space into "di-connected components" such that points (states) in the same component have the same properties with respect to dipaths (executions) visiting them.

Throughout this section, $(X, \leqslant)$ is supposed to be a locally partially ordered topological space (lpo-space).

4.1. Dihomotopy

It is important that the dipaths to be considered cannot be shrinked under a deformation. This is taken care of in

Definition 4.1. (1) A dipath $\alpha : \vec{I} \to X$ is called *inextendible*, if there is no dipath $\beta : \vec{J} \to X$ such that $\alpha(\vec{I}) \subset \beta(\vec{J})$ and $\alpha(\vec{I}) \neq \beta(\vec{J})$.

(2) The set of all inextendible dipaths in X is denoted as $\vec{P}_1(X)$.

In particular, an inextendible dipath α starts at an initial point and ends at a final point of the lpo-space.

Let $\vec{I}$ denote the ordered unit interval, and let I denote another unit interval (with equality as partial order, i.e., without using order properties).

Definition 4.2. (1) A continuous map $H : \vec{I} \times I \to X$ is called a *dihomotopy* if every partial map $H_t = H(-, t) : \vec{I} \to X,\ t \in \vec{I}$, is an inextendible dipath.

(2) Two inextendible dipaths $\alpha, \beta : \vec{I} \to X$ are called *dihomotopic* if there is a dihomotopy $H : \vec{I} \times I \to X$ with $H_0 = \alpha$ and $H_1 = \beta$. We write: $\alpha \precsim \beta$.

(3) The set of dihomotopy classes of inextendible dipaths in X is denoted as $\vec{\pi}_1(X)$.

Remark 4.3. (1) A homotopy of paths is just a 1-parameter deformation. What is new in the definition above, is that we insist that *all intermediate* paths H_t preserve the orientation.

(2) The local partial order need *not* be preserved wrt to the variables in I.

(3) Dihomotopy is obviously an equivalence relation.

(4) Without the restriction to *inextendable* dipaths, every dipath $\alpha : \vec{I}$ would be dihomotopic to a constant dipath via the dihomotopy

$$H(s, t) = \alpha((1 - t)s), \quad s \in \vec{I},\ \ t \in I.$$

Then all dipaths in the same connected component of X would be dihomotopic to each other.

(5) Assume that both the set of initial points and the set of final points in X are *discrete*. Then, 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. Similarly, they have to end in the same final point. Hence, in this case, $\vec{\pi}_1(X)$ splits into subsets corresponding to every particular pair of an initial and a final point.

Why is dihomotopy a relevant concept? It can be argued in fact that schedules (or dipaths) which can be deformed into each other by a continuous family of executions are in fact equivalent (yield same results) *for all* possible actual values that the semaphore (in fact, the variable they protect) can take. Since, in our geometric model, schedules correspond to dipaths, dihomotopic dipaths correspond to equivalent executions.

To see why this holds, consider for instance Fig. 2. Any dipath below the central hole (with shape of a cross), see the second picture of Fig. 4, is such that T_1 gets hold of locks a and b before T_2 does. If we suppose that all assignments on a and b are protected by the corresponding P operations, it is clear that only the order of accesses to the shared variables on an execution path counts for determining what is actually computed on this path, as shown in the second picture of Fig. 4. In fact in this system, there are only two essential behaviours (that do not go into a deadlock see first picture of Fig. 4): one is the type of dipaths just discussed, the other one is the class of dipath going above the central hole (see third picture of Fig. 4). In terms of schedule of executions, the latter corresponds to executions in which T_2

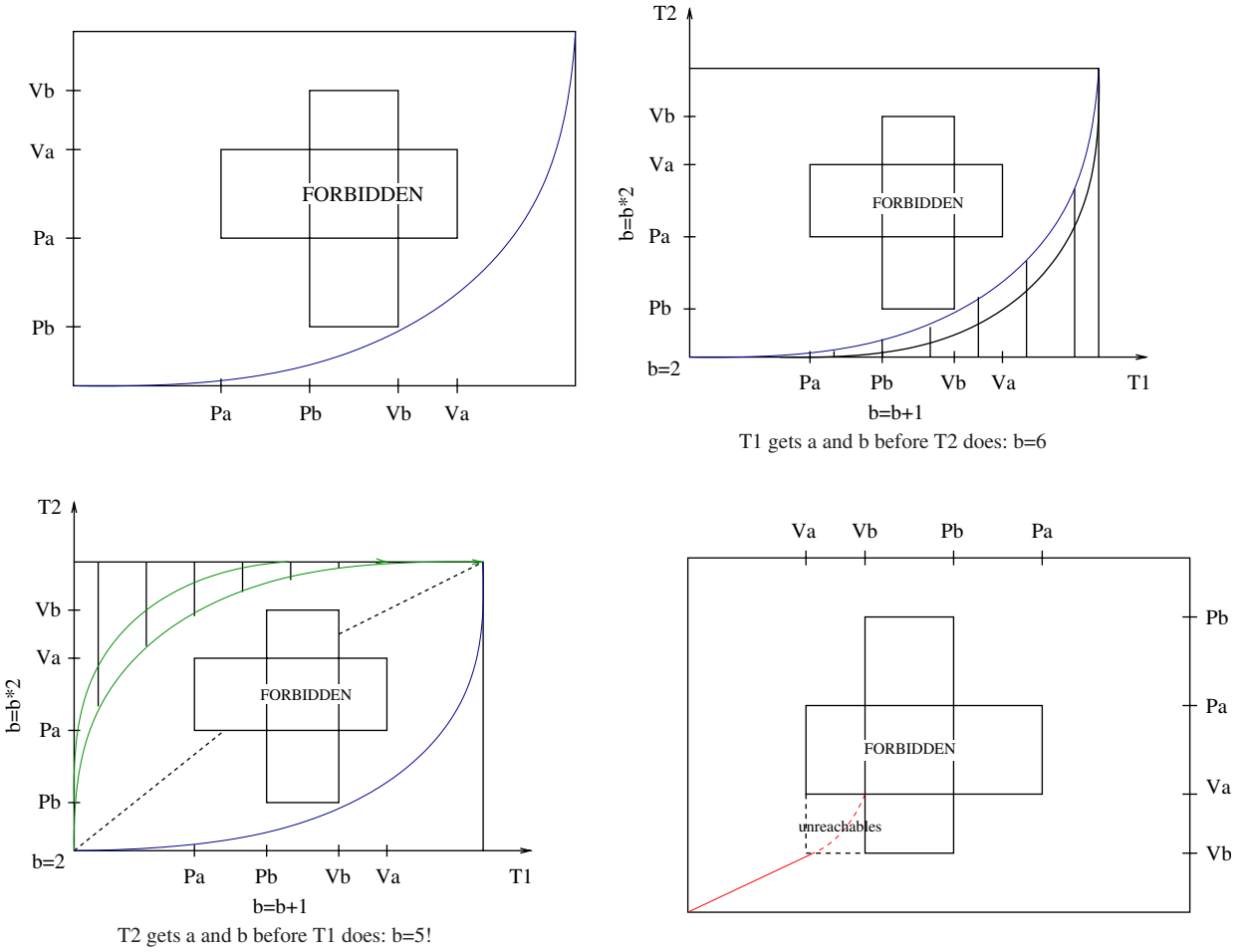


Fig. 4. Essential schedules for the Swiss flag.

is the first to read and write (after having got the corresponding locks) on a and b, before T_1 does. These are of course the only two dipaths from $(0, 0)$ to $(1, 1)$ modulo "continuous deformations".

There is another rather combinatorial point of view on dipaths and dihomotopies, in which local dihomotopies are modelled by rectangles (with the semantical meaning: first a then b is equivalent to first b then a, this is very much similar to Mazurkiewicz trace theory [61], and this provides another reason for dihomotopy to provide us with the "right" equivalence relation on dipaths). In a more abstract setting, see also our Section 7, this version of dihomotopy has been investigated by Grandis [46,45]. For nice enough lpo-spaces (like the cubical sets from Section 6, both dihomotopy concepts are shown to be equivalent by Fajstrup [19]).

Previously, it had been conjectured [65] in a similar context, that homotopy (instead of our dihomotopy) was the proper notion to discriminate essentially different schedules. Our Example 4.4 (4) and its interpretation in Section 4.3 shows that homotopy without directions is not sufficient.

Example 4.4. The following examples show several crucial effects of the directedness (monotonicity) requirements in the definitions:

(1) Fig. 5 represents a path from an initial to a final point that cannot be homotoped to a dipath (with fixed end-points).
(2) Fig. 6 shows two po-spaces (a square with two holes in different positions) that are homeomorphic (even fixing the end points) but that show different behaviour with respect to dihomotopy. For the po-space on the left, there are three classes of dihomotopy classes of dipaths from the bottom to the top; for the one on the right, there are four

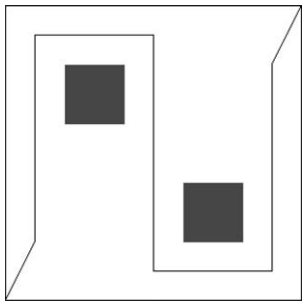


Fig. 5. A path that is not homotopic to a dipath.

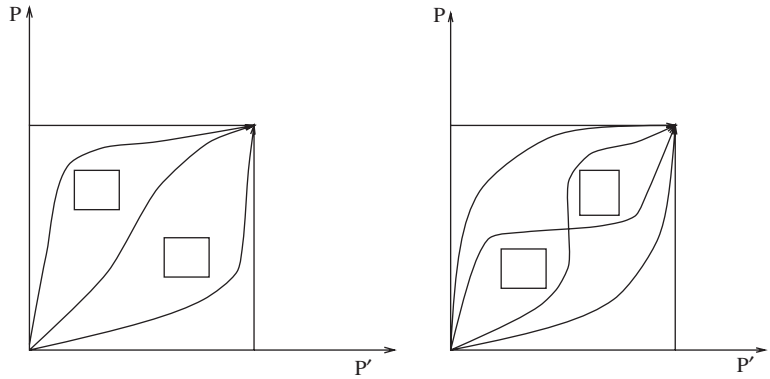


Fig. 6. The two possible relative configurations of holes.

dihomotopy classes. In particular, the two spaces cannot be dihomeomorphic to each other (relative to end points), i.e., there is no invertible *di*map from one to the other. Here is a practical interpretation for this distinction: for the program on the left side, there is no schedule allowing the vertical process to arrive first at the left obstruction as well as the horizontal process to arrive first at the right obstruction (Fig. 6).

(3) For the complement of the Swiss flag in Fig. 2, there are four classes of dihomotopy classes: two from the bottom to the top, one from the bottom to the deadlock point, and one from the lower corner of the unreachable region to the top point.

(4) Fig. 7 shows a "room with three barriers" and two dipaths from the unique initial to the unique final point. The po-space in question is a solid cube with three blocks removed. The two vertical rectangular blocks stretch from the bottom surface to the top surface. Without the third middle block removed, this corresponds to a product of the rightmost space from Fig. 2 with an interval $\vec{I}$—which would again give rise to four dihomotopy classes. But now we remove additionally a third middle box, that does not touch any boundary surface, but such that its extremities overlap with those of the two other blocks when projected to the front face.

Consider the two dipaths in Fig. 7: both of them pass the first block on the right and the last block on the left. But the first dipath passes under and the second over the middle block. When regarded as paths from the initial point to the final point, i.e., forgetting about orientations, these two paths are homotopic, even fixing the end points. One may connect the two dipaths by a homotopy (1-parameter family) of *paths* which pass by either to the left or to the right of the middle block. But no such homotopy can be directed. It has to contain an intermediate path which changes direction—which is prohibited for a dipath. This is not just a geometric artefact. A concurrent program with that geometric model with different results for the schedules modelled by the two dipaths will be given in Section 4.3.

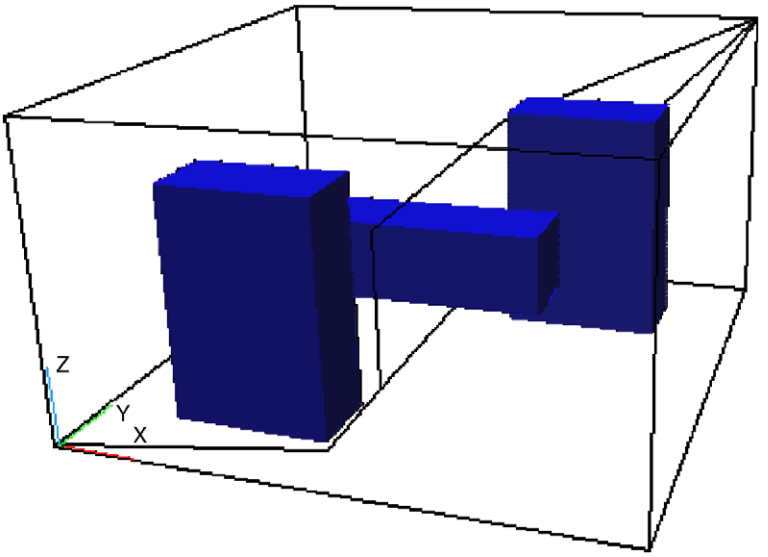


Fig. 7. Room with three barriers and two non-dihomotopic dipaths.

4.2. Diconnected components

Which points (states) can be visited by a dipath dihomotopic to a given one? To answer this question, we need to introduce a few concepts:

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

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

(2) Two points $x, y \in X$ are *homotopy history equivalent* if

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

(3) The *diconnected components* of X consists of the path 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.

(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 vice versa.

(3) Two points $x, y \in X$ may be history equivalent, but there is no path inbetween them passing only through points history equivalent to them. Then, they do not belong to the same diconnected component, cf. Example 4.7.

(4) If there are finitely many dihomotopy classes of dipaths in X, 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

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

(5) In subsequent work [20], the concept of a diconnected component has been refined. Using the *fundamental category* of a po-space, we define and analyse its *component category*.

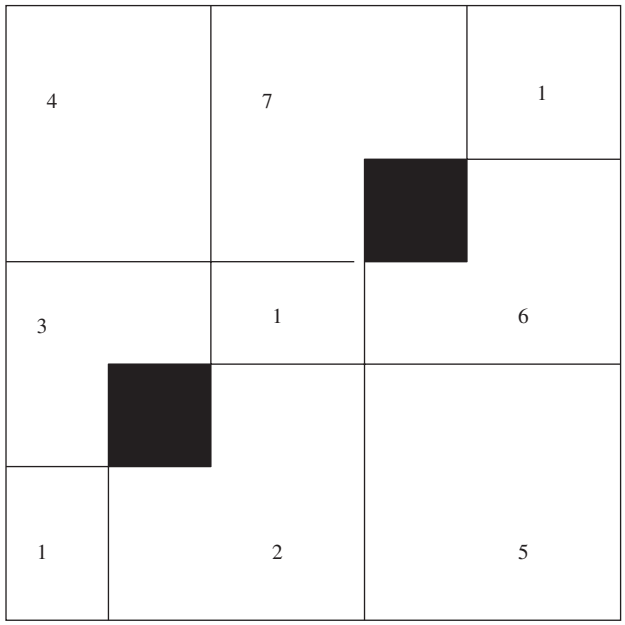


Fig. 8. "Two partially ordered holes".

Point 4 of the remark above is of primary importance for program analysis. Each homotopy history $h\alpha$ corresponds to some property of accesses of shared resources in the PV model. The decomposition of the state space X shows that there are elementary regions which are separated by the properties that executions visiting 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 $\vec{I}^2$ (see Fig. 2) 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). From region 2, we can proceed to 4 or to 6, meaning that we are either going to deadlock (this is the unsafe region of the first picture of Fig. 4) in the future or that T_2 will get a and b before T_1. Region 6 can only be reached from 2 and continue to 9: T_2 has got a and b before T_1. Region 9 can be reached from the unreachable region 7 (that we already identified as the fourth picture of Fig. 4) 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 $\vec{I}^2$ (see Fig. 8) has seven homotopy history components. Region 1 contains both the initial point **0**, the final point **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 $Pa.Va.Pb.Vb|Pa.Va.Pb.Vb$.

(3) The "room with three barriers" in I^3 from Example 4.4 (4) has nine homotopy history components. This can be seen by comparing with Fig. 8 by a projection. In fact, the region over the middle component (labelled 1) splits into three history components according to which of the two "critical" dihomotopy classes can visit (the first, the second or both).

4.3. A weak synchronization example

The rest of this section is devoted to an example elaborating on Example 4.4 (4)—the "room with three barriers" and shows that the two homotopic but not dihomotopic dipaths may correspond to essentially different executions in a concrete situation:

For this and further examples, we have to enhance our toy programming languages slightly 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 $k \geqslant 1$. Examples of such objects can be redundant functional units (for instance, in microprocessors, or in workshop modelizations), 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 s in the local memory, in the local array y. Then any array operation like access at the `i`th element, "`s[i]`", can be performed locally by doing $y[i]$.
- `push(x,s)` corresponds to asking to take one of the free places of the buffer s (in FIFO order here) to put value x: 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.

To explain the different meanings of the two dipaths in Example 4.4 (4) in computer-scientific terms, consider the following three programs:.

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

Basically,

- T_1: sets value 1 to a (remember, this is just a shared variable), then pushes the current value of $b[1]$ by the current value of a on the 2-cell buffer b, and finally sets c (a shared variable) to the sum of the elements in b.
- T_2: just pushes the value of a plus one in b.
- T_3: first sets a to be equal to 3, then pushes 0 on b, then the value of a times the value of the first cell of b, then puts the sum of the entries of b in c.

Notice that the functions that are computed by each of the three processes are sensitive to the order in which values are written in `b`, hence, as we will see, are able to distinguish the two schedules pictured in Fig. 7. The first dipath (the one below the central hole in Fig. 7) 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	
–	–	Pb	$a = 3$
–	–	Va	$b = (\square_3 0, 0)$
Pa	–	–	
Va	–	–	$a = 1$
Pb	–	–	
Vb	–	–	$b = (\square_1 1, \square 0)$
Pc	–	–	
Vc	–	–	$c = 1$
–	–	Pc	
–	–	Vb	$b = (\square_3 3, 1)$
–	–	Vc	$\underline{c = 4}$
–	Pb	–	
–	Vb	–	$b = (\square_1 2, 3)$

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	
–	–	Pb	$a = 3$
–	–	Va	$b = (\square_3 0, 0)$
Pa	–	–	
Va	–	–	$a = 1$
–	Pb	–	
–	Vb	–	$b = (\square_2 2, \square_3 0)$
Pb	–	–	
Vb	–	–	$b = (\square_1 3, \square_2 2)$
Pc	–	–	
Vc	–	–	$c = 5$
–	–	Pc	
–	–	Vb	$b = (\square_3 9, 3)$
–	–	Vc	$\underline{c = 12}$

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.

5. Parametrized and Euclidean partial orders

In this section, we look at a particular subcategory of lpo-spaces, where locally, the time ordering is the componentwise ordering in $\mathbb{R}^n$. These spaces are a special case of parametrized spaces in which it is possible to "cut transversally to time". This technique allows sometimes, but not always, to find out that two dipaths are *not* dihomotopic: this is the case, when those pass through *distinct connected components* of the cuts.

5.1. Definitions

Definition 5.1. Let U be a set with a partial order $\leqslant$. A subset $V \subset U$ is called *achronal* if for all $x, y \in V : x \leqslant y \Rightarrow x = y$; compare [64].

Example 5.2. In the po-space $(\mathbb{R}^2, \leqslant)$ with the natural partial order, the *anti-diagonal* $\bar{\Delta} = \{(t, -t) |\ t \in \mathbb{R}\}$ is achronal.

Achronal cuts are much like instantaneous snapshots that are at the base of some recent geometric proofs of nonexistence of some fault-tolerant protocols for distributed systems (see for instance [50]). The idea is that, under some circumstances (for instance, studying a distributed system with a global clock), one can observe a system at some given "still moments", and that one can find interesting *classical* algebraic topological invariants that help understand the possible states that a system can reach. We refer to this work more in detail in Section 10.

Definition 5.3. Let $(X, \leqslant)$ denote a po-space.

(1) We call $(X, \leqslant)$ *parametrized* if there is a (parameter) dimap $F : X \to \mathbb{R}$ such that every *cut* $X_t := F^{-1}(t)$ is achronal for every $t \in \mathbb{R}$.
(2) We call $(X, \leqslant)$ *Euclidean* if there are finitely many dimaps $f_i : X \to \mathbb{R}$ such that

$$\forall x, y \in X : \quad x \leqslant y \ \Leftrightarrow\ \forall i : \quad f_i(x) \leqslant f_i(y); \forall i : f_i(x) = f_i(y) \ \Leftrightarrow\ x = y.$$

(3) A *local* partial order on a topological space X is called *parametrized*, resp. *Euclidean* if it (or a refinement of it) consists of parametrized, resp. Euclidean partial orders.

Remark 5.4. For a Euclidean partial order, the relation $\leqslant$ is given by comparison (via the map $f = (f_1, \ldots, f_n) : X \to \mathbb{R}^n$) with the natural partial order on Euclidean space $\mathbb{R}^n$ from Example 3.2.

Lemma 5.5. (1) *A parametrized po-space gives rise to a new partial order* $\leqslant'$ *defined by*: $x \leqslant' y \Leftrightarrow x = y$ or $F(x) < F(y) \in \mathbb{R}$.
It satisfies: $x \leqslant y \Rightarrow x \leqslant' y$.
(2) *A Euclidean partial order is parametrized.*

Proof. (1) Obvious.
(2) Let X and $f_i : X \to \mathbb{R}$ be given as in Definition 5.3.2. The function $F = \sum f_i : X \to \mathbb{R}$ is a dimap. For every $t \in \mathbb{R}$, the preimage $X_t := F^{-1}(t) \subset X$ is achronal. □

5.2. Well-parametrized dipaths and dihomotopies

For the rest of this section, let $(X, \leqslant, F)$ denote a parametrized po-space *foliating* X into the cuts X_t, $t \in \mathbb{R}$. We arrive at a better understanding when the parameter map F is used to reparametrize dipaths and dihomotopies to yield new parametrizations matching with that foliation:

Definition 5.6. Let $\vec{J}$ denote a closed real interval.
(1) A dipath $\alpha : \vec{J} \to X$ is called *well-parametrized* if $F(\alpha(t)) = t$ for every $t \in \vec{J}$.
(2) A dihomotopy $H : \vec{J} \times I \to X$ is called well-parametrized if $F(H_t(0))$ is constant and if every dipath $H_t : \vec{J} \to X$, $t \in I$ is well-parametrized.
(3) A dipath $\beta : \vec{J} \to X$ is called a *reparametrization* of another dipath α if there is a *monotonic* map $h : \vec{I} \to \vec{J}$ such that $\alpha = \beta \circ h$.

Remark, that also $F(H_t(r))$ is constant along a well-parametrized dihomotopy H.
Almost as in any course on elementary differential geometry, we get:

Proposition 5.7. (1) *To any dipath* $\alpha : \vec{I} \to X$, *there is a unique well-parametrized reparametrization* $\beta : \vec{J} \to X$.
(2) *To any dihomotopy* $H : \vec{I} \times I \to X$ *from one cut to another* (*i.e.*, $F(H_t(1))$ *and* $F(H_t(0))$ *are both constants*), *there is a unique reparametrization* $\bar{H} : \vec{J} \times I \to X$, *i.e.*, *such that every dipath* $\bar{H}_t$ *is a well-parametrized reparametrization of* H_t *for every* $t \in I$.

Proof. The following proofs are relatively complicated, since we do not assume, that the dipaths involved are strictly monotonic. For strictly monotonic maps, one arrives at more elementary proofs via the *dihomeomorphism* $h := F \circ \alpha$.
(1) The map h given by $h(s) := (F \circ \alpha)(s)$ is monotonic and continuous; its image is the interval $\vec{J} = [F(\alpha(0)), F(\alpha(1))]$. For $s \in \vec{J}$, choose $0 \leqslant t_s \in \vec{I}$ such that $h(t_s) = h(0) + s$ using the continuity of α. The assignment $\beta(s) = \alpha(t_s)$ yields a well-defined monotonic map β. From the definition, we get immediately, that $\alpha = \beta \circ h$. The map β is continuous, as well: let $U \subset X$ be an open non-empty set. Then $\alpha^{-1}(U)$ is either empty or it contains an open interval, on which α is not constant. Hence, $\beta^{-1}(U)$ contains an open interval, as well.
(2) For every $t \in I$, let $h_t : \vec{I} \to \vec{J}$ denote the monotonic map given by $h_t(s) := (F \circ H_t)(s)$. Proceeding as in (1) for every $t \in I$, we arrive at a unique map $\bar{H} : \vec{J} \times I \to X$ such that $H_t = \bar{H}_t \circ h_t$. Each map $\bar{H}_t$ is a well-parametrized dipath; the trouble is to ensure that the map $\bar{H}$ is continuous:
Let again $U \subset X$ denote an open set and let $\bar{H}(\bar{s}_0, t_0) \in U$. Choose $s_0 \in \vec{I}$ such that $H(s_0, t_0) = \bar{H}(\bar{s}_0, t_0)$. Since H_t is a dipath, one can choose $0 < \varepsilon, \delta_1, \delta_2$ such that $H([s_0 - \delta_1, s_0 + \delta_2]) \subset U$ and
$F(H(s_0 - \delta_1, t_0)) = F(H(s_0, t_0)) - \varepsilon$, $F(H(s_0 + \delta_2, t_0)) = F(H(s_0, t_0)) + \varepsilon$.

The continuity of H and a compactness argument allows to choose $0 < \delta_i' < \delta_i,\ 1 \leqslant i \leqslant 2,\ 0 < \rho$, such that $H([s_0 - \delta_1', s_0 + \delta_2'] \times]t_0 - \rho, t_0 + \rho[) \subset U$ and

$$F(H(s_0 - \delta_1', t_0 + \tau)) < F(H(s_0, t_0)) - \frac{\varepsilon}{2}, F(H(s_0 + \delta_2', t_0 + \tau)) > F(H(s_0, t_0)) + \frac{\varepsilon}{2}$$

for $|\tau| < \rho$. As a consequence,
$\bar{H}(]\bar{s}_0 - \frac{\varepsilon}{2}, \bar{s}_0 + \frac{\varepsilon}{2}[\times]t_0 - \rho, t_0 + \rho[) \subseteq H(]s_0 - \delta_1', s_0 + \delta_2'[\times]t_0 - \rho, t_0 + \rho[) \subseteq U$. □

5.3. Components of cuts distinguish dihomotopy classes

It is in general not easy to find out whether two given dipaths are dihomotopic or not. This is the main motivation for investigating cuts $X_t := F^{-1}(t)$ with respect to a parametrization $F : X \to \mathbb{R}$. We show here that dipaths proceeding through *different* (classical) connected components of a particular cut can never be dihomotopic:

Proposition 5.8. *Let $\alpha_1, \alpha_2 : (\vec{I}, 0, 1) \to (X; X_a, X_b)$ denote dimaps that are dihomotopic by a dihomotopy $H : (\vec{I}, 0, 1) \times I \to (X; X_a, X_b)$, which preserves the "end cuts" X_a and X_b. Then, the uniquely determined elements $x_t^i \in \alpha_i(\vec{I}) \cap X_t,\ 1 \leqslant i \leqslant 2$, are contained in the same path component of X_t for $t \in [a, b]$.*
In other words: if α_1 and α_2 pass through different path components of X_t for some parameter $t \in \mathbb{R}$, then α_1 and α_2 cannot be dihomotopic.

This proposition is an immediate consequence of the existence of a reparametrization from Proposition 5.7 and of the following particular case:

Proposition 5.9. *Let $H : \vec{J} \times I \to X$ denote a well-parametrized dihomotopy between two well-parametrized dipaths $\alpha_1, \alpha_2 : I \to X$. Then, $\alpha_1(t)$ and $\alpha_2(t)$ are contained in the same path component of X_t for every $t \in I$.*

Proof. The map $H(-, t) : I \to X_t$ is a path from $\alpha_1(t)$ to $\alpha_2(t)$. □

Remark 5.10. The criterion in Proposition 5.8 is exploited to give a complete classification of dihomotopy classes for a 2D mutual exclusion model in [67]. Two different parameter maps, i.e., the two projections to the axes, have to be employed for this task.

Unfortunately, the criterion from Proposition 5.8 is in general *not* sufficient to distinguish different dihomotopy classes. An easy example showing that a family of *diagonal* cuts alone is not enough is:

Example 5.11. 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, i.e., a cube with an interior box removed, as pictured in Fig. 9. This po-space models the PV program:

```
A=Pa.Va
B=Pa.Va
C=Pb.Vb
```

where `a`, `b` and `c` are all binary semaphores.

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; in fact, they are polygons with or without a hole.

Hence to get full information about dihomotopy classes, it does not suffice to study *just one* family of cuts. One might still ask whether the non-existence of a dihomotopy—within nice po-spaces, e.g., those arising from cubical sets (treated in next section)—can always be detected if one knows *all* families of cuts and their connected components.

6. Cubical sets as local po-spaces

Cubical sets were defined by Serre [72] and a theoretical framework is developed by Brown and Higgins [7], and by Jardine [55], see also [34] for their use as models for HDA. We show here that they are the natural combinatorial

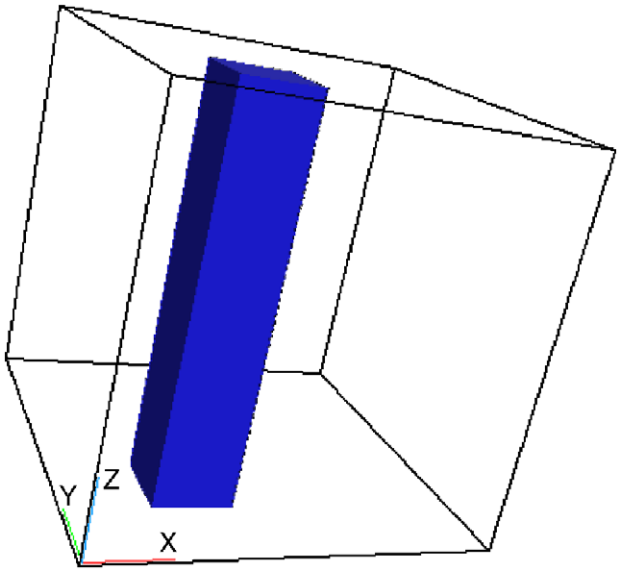
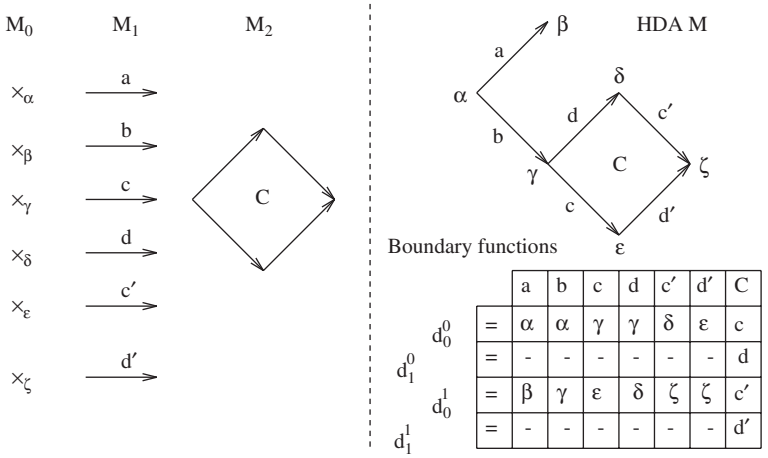


Fig. 9. A simple example where instant snapshots do not distinguish dihomotopy classes.



		a	b	c	d	c′	d′	C
d^0_0	=	α	α	γ	γ	δ	ε	c
d^0_1	=	-	-	-	-	-	-	d
d^1_0	=	β	γ	ε	δ	ζ	ζ	c′
d^1_1	=	-	-	-	-	-	-	d′

Fig. 10. The glueing of elementary cubes.

counterpart of local po-spaces (the centre of this is Theorem 6.23 and Proposition 6.38). This makes the link with more standard combinatorial techniques for reasoning about concurrent systems (interleaving ones or truly concurrent ones like HDA). For the correspondence with transition systems, see Goubault [40].

6.1. Cubical sets

In Fig. 10, the connection between the combinatorial description and the geometric description of an HDA is pictured. In the following, the reader should keep in mind, that this is what we are defining: a combinatorial way of describing the relations between a set of cubes of varying dimensions, glued along common boundaries or faces.

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

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

The (iterated) face maps determine glueings of cubes, e.g, if $\partial^l_j z = \partial^k_i y = x$, then the cubes z and y share a face, x. If $k = 1$, then x is an upper face of y and if $k = 0$, it is a lower face. The commutator rules follow from this point of view.

Remark 6.2. In Brown et al. [8,7], a cubical set is defined—our precubical sets differ from cubical sets in that they do not have degeneracies.

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

$$f_n \circ \partial^0_i = \partial^0_i \circ f_{n+1},$$

$$f_n \circ \partial^1_i = \partial^1_i \circ f_{n+1}$$

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

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

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

6.2. The geometric realization of a pre-cubical set

Given a pre-cubical set M, we want to construct the corresponding geometric object $|M|$ by glueing cubes. For this purpose, let $\square_n$ be the standard cube in $\mathbb{R}^n$ $(n \geqslant 0)$, $\square_n = \{(t_1, \ldots, t_n) | \forall i, 0 \leqslant t_i \leqslant 1\}$, $\square_0 = \{0\}$.

Notice that for cubes $y = \{(t_1, \ldots, t_n) \in \square_n\}$ and $x = \{(u_1, \ldots, u_{n-1}) \in \square_{n-1}\}$, the statement x is the ith upper face of y could be expressed $\partial^1_i y = x$ or by defining an inclusion $\delta^1_i : x \to y$,

$$\delta^1_i(u_1, \ldots, u_{n-1}) = (u_1, \ldots, u_{i-1}, 1, u_i, \ldots, u_{n-1}).$$

This dual point of view is given here:

Let $\delta^k_i : \square_{n-1} \to \square_n$, $1 \leqslant i \leqslant n$, $k = 1, 2$, be the continuous functions $(n \geqslant 1)$

$$\begin{array}{ccc} \square_n & \xleftarrow{\delta^0_i} & \square_{n-1} \\ {\scriptstyle \delta^1_i}\uparrow & & \\ \square_{n-1} & & \end{array}$$

defined by

$$\delta^k_i(t_1, \ldots, t_{n-1}) = (t_1, \ldots, t_{i-1}, k, t_i, \ldots, t_{n-1}).$$

Then,

Lemma 6.4.

$$\delta^k_i \delta^l_j = \delta^l_{j+1} \delta^k_i \quad (i \leqslant j).$$

Proof. Left to the reader.

We notice that, as indicated above, the commutator relations satisfied by the δ^k are dual to the commutator relations satisfied by ∂^k in a pre-cubical set.

Definition 6.5. Let M be a precubical set and let $\mathbf{R}(M) = \coprod_n M_n \times \square_n$ with disjoint sum topology induced by the discrete topology in M_n and the topology from $\mathbb{R}^n$ on $\square_n$.

Let $\equiv$ be the equivalence relation induced by the identities:

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

Let $|M| = \mathbf{R}(M)/\equiv$ have the quotient topology. The topological space $|M|$ is called the *geometric realization* of M.

Another point of view is that M is a labelling of a subdivision of $|M|$ into cubes. An element $y_m \in M_m$ is the label of an m-cube in $|M|$. Let $p \in |M|$, then there is a minimal cube in the subdivision of $|M|$ containing p, namely the unique cube $x \times \square_k$ which has p in the interior. We call x the *carrier* of p.

6.3. An open covering of $|M|$ and a subclass of precubical sets

We want to define a local partial order on $|M|$ using the partial order on each cube and taking a "local" transitive closure. For this we need an open covering.

Definition 6.6. Let M be a pre-cubical set and y an element of M. Then $x \in M$ is a *face* of y if there exists a (possibly empty) collection of indices $k_1, \ldots, k_1$ being 0 or 1 and $l_1, \ldots, l_i$ integers such that $x = \partial_{l_1}^{k_1} \cdots \partial_{l_i}^{k_i} y$.

Definition 6.7. The open star of a point $p \in |M|$ with respect to the subdivision M is

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

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

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

Remark 6.8. (1) If $x \in M_n$ is the carrier of $p \in |M|$, then $St(p, M)$ is the union of the interiors of cubes in $|M|$ which are labelled by an element of $St(x, M)$.

(2) By an abuse of notation, we will omit M and write $St(p)$ and $St(x)$ if there is no risk of confusion. And moreover, when x is a cube, we will write $St(x)$ for the geometric object $St(p)$ where $p \in |M|$ and x is the carrier of p.

(3) If p and q in $|M|$ have the same carrier, i.e., if they are in the interior of the same cube, then $St(p) = St(q)$

(4) For $p \in |M|$, and $x \in M_n$ its carrier, there is a unique $t \in \overset{\circ}{\square}_n$, s.t. $p = [(x, t)] \in R(M)/\equiv$. This means that there is a canonical representative of the equivalence classes in $R(M)/\equiv$.

The stars of vertices $v_i \in M_0$ define an open covering of $|M|$ and the partial order on each of these should be the transitive hull of the partial order in the (open) cubes.

There are some obvious problems with this, such as if the upper and lower vertex of an interval $\square_1$ are identified, in which case the star of this vertex is a loop and the transitive hull of the relation relates all points to all points. To avoid this and also some less obvious problems, we restrict the class of precubical sets:

Definition 6.9. Let M be a pre-cubical set. M is a non-self-linked cubical set if for all faces x of an n-cube y, x can be written uniquely 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*").

Remark 6.10. One may still have loops in $|M|$, 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.

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 ensure that this works for iterated boundaries:

Lemma 6.11. *Let M be a precubical set. 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 $l'_1 < l'_2 < \cdots < l'_m$, 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 get a canonical form. □

When the complex is non-self-linked, the canonical form is unique. And there is another unique form:

Lemma 6.12. *Let M be a non-self-linked precubical complex, and x, y be elements of M. Suppose x is a face of y. Then x can be written uniquely as*
- $x = \partial^0_{l'_1} \cdots \partial^0_{l'_j} \partial^1_{l'_{j+1}} \cdots \partial^1_{l'_i} y$ *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, $x = \partial^{u_1}_{v_1} \cdots \partial^{u_i}_{v_i} y$ verify the following: let $K_0(x, y)$ (respectively, $K_1(x, y)$) be the cardinal of the set $\{j/1 \leqslant j \leqslant i, k_j = 0\}$ (respectively, $\{j/1 \leqslant j \leqslant i, k_j = 1\}$), then $K_0(x, y)$ (respectively, $K_1(x, y)$) is also the cardinal of $\{j/1 \leqslant j \leqslant i, u_j = 0\}$ (respectively, $\{j/1 \leqslant j \leqslant i, u_j = 1\}$). Hence $K_k(x, y)$ depends only on x and y.

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. □

6.4. The face ordering

Lemma 6.13. *Let M be a non-self-linked cubical set, 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 $x = \partial^{k_1}_{l_1} \cdots \partial^{k_i}_{l_i} y$ and $y = \partial^{k'_1}_{l'_1} \cdots \partial^{k'_j}_{l'_j} x$ by definition. So $x = \partial^{k_1}_{l_1} \cdots \partial^{k_i}_{l_i} \partial^{k'_1}_{l'_1} \cdots \partial^{k'_j}_{l'_j} x$ 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 6.12. But $K_0(x, x) = K_1(x, x) = 0$ again by Lemma 6.12 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 $x = \partial^{k_1}_{l_1} \cdots \partial^{k_i}_{l_i} y$ and $y = \partial^{k'_1}_{l'_1} \cdots \partial^{k'_j}_{l'_j} z$. So $x = \partial^{k_1}_{l_1} \cdots \partial^{k_i}_{l_i} \partial^{k'_1}_{l'_1} \cdots \partial^{k'_j}_{l'_j} z$ and $x <_F z$.

Lemma 6.14. *Let M be a non-self-linked cubical set, 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 $x = \partial^{k_1}_{l_1} \cdots \partial^{k_i}_{l_i} y$ and $y = \partial^{k'_1}_{l'_1} \cdots \partial^{k'_j}_{l'_j} z$. So $x = \partial^{k_1}_{l_1} \cdots \partial^{k_i}_{l_i} \partial^{k'_1}_{l'_1} \cdots \partial^{k'_j}_{l'_j} z$ 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. □

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

Lemma 6.15. *Let M be a non-self-linked cubical set 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 $|y|$ in $|M|$ whose carrier is y on the cube $|x|$ 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 $|M|$, 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 $l_1, \ldots, l_j$. 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. □

Remark 6.16. We actually only need M to be non-singular, i.e., that $\partial^k_l x = \partial^{k'}_{l'} x \Rightarrow k \neq k'$, for the above lemma to hold.

Lemma 6.17. *Let M be a non-self-linked cubical set. 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-linked 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 $b = \partial_{l_1}^{k_1} \cdots \partial_{l_j}^{k_j} y$ and $b' = \partial_{l'_1}^{k'_1} \cdots \partial_{l'_{j'}}^{k'_{j'}} y$ on canonical form. Then $b = \{(t_1, \ldots, t_n) \in \square_n | t_{l_i} = k_i \text{ for } i = 1 \ldots j\}$ and $b = \{(t_1, \ldots, t_n) \in \square_n | t_{l'_i} = k'_i \text{ for } i = 1 \ldots j'\}$ and the intersection $b \cap b' = \{(t_1, \ldots, t_n) \in \square_n | t_{l_i} = k_i \text{ for } i = 1 \ldots j \text{ and } t_{l'_i} = k'_i \text{ for } i = 1 \ldots j'\}$.

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. □

Another useful fact, which is geometrically quite clear is the following:

Lemma 6.18. *Suppose w is an upper face of an n-cube y and that x is a face in y which has a non-empty intersection with w. Then the upper vertex of x is in w.*

Proof. Let $w = \partial_{l(y,w)}^{\mathbf{1}} y$ and $x = \partial_{l(y,x)}^{k(y,x)} y$ be the canonical representations of w and x as faces in y. Since x and w have non-trivial intersection, whenever i is a coordinate in the vector/multi-index $l(y, x)$ (denoted $i \in l(y, x)$) and $k(y, x)_i = 0$, then $i \notin l(y, w)$. Now the upper vertex v of x considered in y is $v = \partial_{l(x,v)}^{\mathbf{1}} \partial_{l(y,x)}^{k(y,x)} y$. Hence v is the point $(t_1, \ldots, t_n) \in \square_n$ where $t_i = 0$ if $i \in l(y, x)$ and $k(y, x)_i = 0$ and $t_i = 1$ else. This is in w, since $t_i = 0$ only if $i \notin l(y, w)$. □

Remark 6.19. By symmetric arguments we have: if w is a lower face of z which intersects another face x non-trivially, then the lower vertex of x is in w.

6.5. A local partial order on the geometric realization of a non-self-linked precubical complex

In this section M is a non-self-linked precubical complex with geometric realization $|M|$. As remarked above, any p in $|M|$ has a unique representative (x, t) where $x =$ carrier (p) and $t \in \overset{\circ}{\square}_n$ (for some n). When we choose a representative of a point in $|M|$, it will always be this one.

We give a partial order $\leqslant_x$ on the open neighbourhood $U^x = St(p, M)$ of p, for all p, and we prove that the open cover of $|M|$ $\mathcal{U} = \{St(v, M) | v \in M_0\}$ by stars of vertices with the partial order $\leqslant_v$ defines a local partial order. A local po-neighbourhood of p is given by $(St(x), \leqslant_x)$, where x is the carrier.

Let $(y, u) \in U^x$, then $x = \partial_{l_1}^{k_1} \ldots \partial_{l_i}^{k_i}(y)$ because $y \in St(x, M)$, so (x, t) is identified with $(y, \delta_{l_i}^{k_i} \ldots \delta_{l_1}^{k_1}(t))$. This identification is unique, since M is non-self-linked, so we can define

Definition 6.20. Let $U^x = St(p, M)$, where x is the carrier of p. We define a relation on U^x:

$$(x, t) \leqslant_{U^x} (y, u) \text{ if } \delta_{l_i}^{k_i} \ldots \delta_{l_1}^{k_1}(t) \leqslant u \quad \text{in } \square_{n+i},$$

$$(y, u) \leqslant_{U^x} (x, t) \text{ if } \delta_{l_i}^{k_i} \ldots \delta_{l_1}^{k_1}(t) \geqslant u \quad \text{in } \square_{n+i}.$$

And we give a partial order on U^x: let $x \in M$ and let (z, v) be a point in U^x with carrier z. We say

$$(z, v) \leqslant_x (y, u)$$

if there exists b in the star of x and t such that

$$(z, v) \leqslant_{U^b} (b, t) \leqslant_{U^b} (y, u).$$

To see that this is in fact a partial order, we need:

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

• *Suppose* $(y, u) \leqslant_{U^x} (x, t)$ *and* $x \neq y$, *then necessarily*, $x = \partial^1_{l_1} \dots \partial^1_{l_j} y$ (*where* $j \geqslant 0$). *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. $y \in St(x, M)$ so there is a collection of indices such that $x = \partial^{k_0}_{l_0} \dots \partial^{k_j}_{l_j} y$ ($j \geqslant -1$). j cannot be equal to -1 since $x \neq y$. Suppose now that there is an index k_i ($0 \leqslant i \leqslant j$) which is equal to one. We have $u \in \overset{\circ}{\square}_n$ and $n \geqslant 1$ (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 $\leqslant_{U^x}$. □

And this gives the following characterization of the partial order:

Lemma 6.22. *Suppose* $(z, v) \leqslant_x (y, u)$, *that is*, $\exists (b, t)$, $b \in St(x, M)$, $(z, v) \leqslant_{U^b} (b, t) \leqslant_{U^b} (y, u)$. *Then we have the following cases*,

(a) $b = \partial^1_{l_1} \dots \partial^1_{l_j} z$, *for some collection of indices and* $K_1(b, z) = j \geqslant 1$ ($K_0(b, z) = 0$); $b = \partial^0_{l'_1} \dots \partial^0_{l'_{j'}} y$, *for some collection of indices and* $K_0(b, y) = j' \geqslant 1$ ($K_1(b, y') = 0$).

(b) $b = z$, $b = \partial^0_{l'_1} \dots \partial^0_{l'_j} y$, *for some collection of indices and* $K_0(b, y) = j \geqslant 1$ ($K_1(b, y') = 0$), *and the relation above shrinks down to* $(z, v) \leqslant_{U^z} (y, u)$.

(c) $b = y$, $b = \partial^1_{l_1} \dots \partial^1_{l_j} z$, *for some collection of indices and* $K_1(b, z) = j \geqslant 1$ ($K_0(b, z) = 0$), *and the relation above shrinks down to* $(z, v) \leqslant_{U^y} (y, u)$.

(d) $y = z$ *and the relation above shrinks down to* $v \leqslant u$.

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 multi-index, $\mathbf{1} = (1, 1, \dots, 1)$, $\mathbf{0} = (0, 0, \dots, 0)$ and they may all be empty.

Now we can state,

Theorem 6.23. *The geometric realization of a non-self-linked cubical set M defines a locally po-space with covering being* $\{St(x, M)/x \in M_0\}$ *and local partial order* $\leqslant_x$ *on* $St(x, M)$.

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

Antisymmetry. Suppose $(z, v) \leqslant_x (y, u)$ and $(y, u) \leqslant_x (z, v)$. This means there are (b, t) and (b', t') with $b \in St(x, M)$ and $b' \in St(x, M)$ such that

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

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

and moreover

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

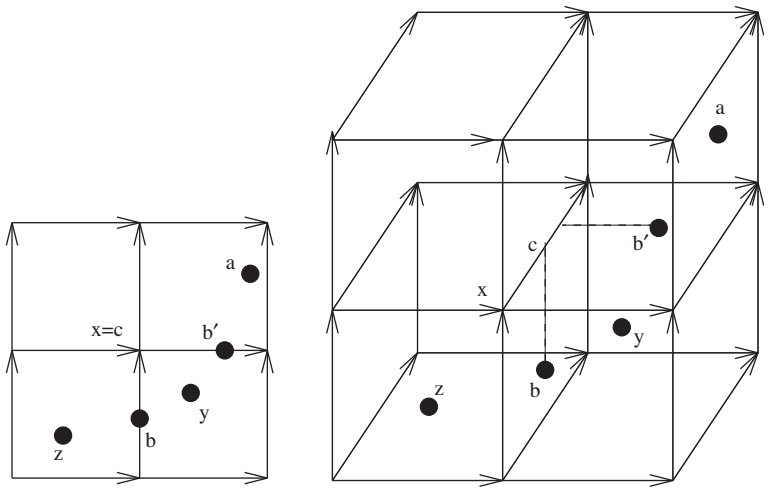


Fig. 11. Illustration of the proof.

where $*$ means a multi-index, $\mathbf{1} = (1, 1, \ldots, 1)$, $\mathbf{0} = (0, 0, \ldots, 0)$ 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 $K_0(b, y) + K_0(b', z) = 0$ 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 $b = z$.

Transitivity. Suppose $(z, v) \leqslant_x (y, u) \leqslant_x (a, w)$. This means there are (b, t) and (b', t') with $b \in St(x, M)$ and $b' \in St(x, M)$ such that,

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

with

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

and moreover

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

where $*$ means a multi-index, $\mathbf{1} = (1, 1, \ldots, 1)$, $\mathbf{0} = (0, 0, \ldots, 0)$ and they may all be empty. By Lemma 6.17, the intersection of b and b' is a face c of y containing x (see Fig. 11). 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 6.15. Define $d = p(t)$. Necessarily (looking at the coordinates in y, as p is a dimap), $p(t') \geqslant d$. Now it is enough to see that

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

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

Po-neighbourhoods. To see that this defines a local partial order, we define po-neighbourhoods: let $p = (x, t) \in |M|$. Then we claim that $St(p, M)$ with partial order defined above, defines a po-neighbourhood of p:

Since $p \in St(v)$ implies $St(x) \subset St(v)$, we have to see, that when $(y, u), (z, v) \in St(x) \subseteq St(v_1) \cap St(v_2)$, then $(y, u) \leqslant_{v_1} (z, v) \Leftrightarrow (y, u) \leqslant_{v_2} (z, v)$. If x is a vertex, then $v_1 = v_2 = x$ and there is nothing to check. Suppose now, that

x is not a vertex and suppose $(y,u)\leqslant_{v_1}(z,v)$. Since $(y,u)\leqslant_{v_1}(z,v)$ if and only if $(y,(\frac{1}{2},\ldots,\frac{1}{2}))\leqslant_{v_1}(z,(\frac{1}{2},\ldots,\frac{1}{2}))$, we will suppose all coordinates are $\frac{1}{2}$ and write $y\leqslant_{v_1}z$.

(1) If $y\leqslant_{v_1}z$ is of type (b), (c) or (d), the inequality is induced by the Euclidean partial order on the cube in $|M|$ labelled y (in case b and d) or the cube labelled z (in case c and d), and the relation will clearly hold in $St(v_2)$ also.
(2) Suppose now $y\leqslant_{v_1}z$ is of type (a). Then there is a $w\in St(v_1)$ s.t. $y\leqslant_{U^w}w\leqslant_{U^w}z$. If $w\in St(v_2)$, we are done.
Now w is an upper face of y and a lower face of z, and v_1 is in the intersection of x, y, z and w. Hence, by 6.18 when we study x and w as faces in y, the upper vertex of x is in w. Similarly, by 6.19 from the inclusion in z, the lower vertex of x is in w, which implies that x is a face of w, so v_2 is a vertex in w. □

To give an idea of some of the locally po-spaces we can construct considering this:

Example 6.24. Let M be the cubical set

$$M_2=A,B,C,D,\quad M_1=a,b,c,d,e,f,g,h,\quad M_0=p,q,r,s,$$
$$d^0_1A=d^0_1C=a,\quad d^0_2A=d^0_2B=b,\quad d^0_1B=d^0_1D=c,\quad d^0_2D=d^0_2C=d,$$
$$d^1_1A=d^1_1D=e,\quad d^1_2A=d^1_2D=f,\quad d^1_1B=d^1_1C=g,\quad d^1_2B=d^1_2C=h,$$
$$d^0_1a=d^0_1b=d^0_1c=d^0_1d=p,\quad d^1_1a=d^1_1c=d^0_1f=d^0_1h=q,$$
$$d^1_1b=d^1_1d=d^0_1e=d^0_1g=r,\quad d^1_1e=d^1_1f=d^1_1g=d^1_1h=s.$$

Then $|M|$ is the projective plane, and one can give cubical models for projective spaces of all dimensions in the same way. Hence a local partial order does not induce an orientation in the manifold sense.

6.6. Subdivision and non-singular precubical sets

Even if the precubical set is self-linked, the above construction may give rise to a local partial order on the geometric realization: the circle constructed by identifying the vertices $\square_1$ could be subdivided into two intervals, which are then non-self-linked. The generalization of this is via barycentric subdivision which we explore in this subsection.

Definition 6.25. Let M be a pre-cubical set. M is a non-singular cubical set if for all its n-cubes x, $\hat{\partial}^k_l(x)=\hat{\partial}^{k'}_{l'}(x)$ implies $k\neq k'$.

So an upper face may be identified with a lower face.

Definition 6.26. Let K be a cubical set and let K' be another cubical set. Then K' is a subdivision of K if there is a dihomeomorphism $f:|K'|\to|K|$ (meaning that f and f^{-1} are dimaps) such that,
- $\forall x\in K'_n,\exists y\in K_n,\ f(x,\square_n)\subseteq(y,\square_m)$,
- $\forall y\in K,\exists x_1,\ldots,x_k\in K',\ (y,\square_m)=\bigcup_{i=1,\ldots,k}f(x_i,\square_{n_i})$.

Definition 6.27. The *standard n-dicube* is the topological space $\square_n$ with the covering $\mathbf{U}=\{\square_n\}$ and local partial order $\leqslant_{\square_n}$ induced from the pointwise ordering in $\mathbb{R}^n$. The n-dicube is then a local po-space.

Definition 6.28. Let M be a local po-space. A singular n-dicube is any dimap from the standard n-dicube to M.

Lemma 6.29. *Let K be a cubical set. The* "*barycentric subdivision*" of *K is defined as follows. Consider the singular n-dicubes of* $|K|$, $\sigma_x:\square_n\to|K|$, $\sigma_x(t)=(x,t)$, *where* $x\in K$, *and the* 2^n *functions*,

$$s_{b_1,\ldots,b_n}:\square_n\to\square_n$$

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

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

Then the subset SdK of $|K|$ with,

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

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

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

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

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

Lemma 6.30. *Let M be a non-singular cubical set. Then $Sd(M)$ is non-self-linked.*

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

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: $\partial_i^k(\sigma_x \circ s_{b_1,\ldots,b_{k-1},k,b_{k+1},\ldots,b_n}) = \partial_j^l(\sigma_x \circ s_{b_1,\ldots,b_{k-1},l,b_{k+1}\ldots,b_n})$. □

Now using both Lemma 6.30 and Theorem 6.23 we can give a local po-space structure to any non-singular cubical set.

6.7. The singular cube functor and locally po-spaces

We study the categorical properties of geometric realization: geometric realization is a functor—from non-self-linked precubical sets to local po-spaces.

We are going to construct a right-adjoint to it in this section.

Lemma 6.31. *Let $f : X \longrightarrow Y$ be a morphism between the two pre-cubical sets X and Y. Then f induces a continuous map $|f|$ from $|X|$ to $|Y|$.*

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 $|X|$ to $|Y|$. It holds trivially for $u = 1$. To prove the induction step 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. □

When the precubical set in non-self-linked, we get a functor to local po-spaces:

Proposition 6.32. *Let $f : M \to N$ be a morphism of pre-cubical sets. Then $|f| : |M| \to |N|$ is a dimap.*

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

The combinatorial counterpart of the standard n-dicube is clearly the faces of such a cube with the boundary maps induced from the geometry.

Definition 6.33. Let $D_{[n]}$ be the free pre-cubical set generated by a unique n-cube $I_{[n]}$, i.e. the pre-cubical set of faces of $I_{[n]}$ which is formally,

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

Then,

Lemma 6.34. • *$|D_{[n]}|$ is homeomorphic to the space $\square_n$ with the usual topology in $\mathbb{R}^n$.*

• *Any morphism of pre-cubical sets $\sigma : D_{[n]} \to M$ such that $f \circ \delta^k_l = f \circ \delta^{k'}_{l'}$ implies $k = k'$ and $l = l'$ (we call this a singular cube) induces a continuous map $|\sigma| : \square_n \to |M|$.*

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

Proposition 6.35. • *$|D_{[n]}|$ is in fact (a refinement of) the po-space $\square_n$ with the componentwise partial order in $\mathbb{R}^n$.*

• *Any singular cube $\sigma_x : D_{[n]} \to M$ induces a dimap $|\sigma_x| : \square_n \to |M|$ (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 6.22. Let us consider again the map

$$\begin{array}{rcl} f_{I_{[n]}} : |D_{[n]}| & \to & \square_n \\ (\partial^{k_1}_{l_1} \cdots \partial^{k_i}_{l_i}(I_{[n]}), v) & \to & \delta^{k_i}_{l_i} \cdots \delta^{k_1}_{l_1}(v). \end{array}$$

We know by Lemma 6.34 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, $f_{I_{[n]}}(x,t) \leqslant f_{[n]}(y,u)$ (respectively, $f_{I_{[n]}}(y,u) \leqslant f_{I_{[n]}}(x,t)$) is equivalent to $(x,t) \leqslant_{U^x} (y,u)$ (respectively, $(y,u) \leqslant_{U^x} (x,t)$). To see this, let $y = \partial^{u_1}_{v_1} \cdots \partial^{u_j}_{v_j}(I_{[n]})$ and $x = \partial^{k_1}_{l_1} \ldots \partial^{k_i}_{l_i}(y)$. $(x,t) \leqslant_{U^x} (y,u)$ is equivalent to $\delta^{k_i}_{l_i} \ldots \delta^{k_1}_{l_1}(t) \leqslant u$ then to $f_{I_{[n]}}(y, \delta^{k_i}_{l_i} \cdots \delta^{k_1}_{l_1}(t)) \leqslant f_{I_{[n]}}(y,u)$ by monotony of f. But $f_{I_{[n]}}(x,t) = f_{I_{[n]}}(y, \delta^{k_i}_{l_i} \cdots \delta^{k_1}_{l_1}(t))$ so this is equivalent to $f_{I_{[n]}}(x,t) \leqslant f_{I_{[n]}}(y,u)$.

Let us consider now (z,v) and (y,u) be any element of $|D_{[n]}|$. Suppose $(z,v) \leqslant_x (y,u)$ for some vertex x of $D_{[n]}$. Then there exists $(b,t) \in |D_{[n]}|$ such that $(z,v) \leqslant_{U^b} (b,t) \leqslant_{U^b} (y,u)$, hence $f_{I_{[n]}}(z,v) \leqslant f_{I_{[n]}}(b,t) \leqslant f_{I_{[n]}}(y,u)$. Inversely, suppose that we have (z,v) and (y,u) such that $f_{I_{[n]}}(z,v) \leqslant f_{I_{[n]}}(y,u)$ and z and y belong to $St(x, D_{[n]})$ for some vertex x of $D_{[n]}$. Then by Lemma 6.17 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 $b = \partial^1_{f_1} \cdots \partial^1_{f_k}(y)$ and $b = \partial^0_{f'_1} \cdots \partial^0_{f'_{k'}}(z)$. So we have $u \leqslant \delta^1_{f_k} \cdots \delta^1_{f_1} p^y_b(u)$, i.e. $(y,u) \leqslant_{U^b} (b, p^y_b(u))$ (using the projection defined in Lemma 6.15).

We also have $\delta^0_{f'_{k'}} \cdots \delta^0_{f'_1} p^z_b v \leqslant v$ thus $(b, p^z_b(v)) \leqslant_{U^b} (z,v)$. Now, $p^y_b(u) \leqslant p^z_b(v)$ because otherwise, looking at the coordinates of $f(y,u)$ and $f(z,v)$ we cannot have $f(y,u) \leqslant f(z,v)$. Therefore $(y,u) \leqslant_{U^b} (b, p^y_b u) \leqslant_{U^b} (z,v)$ hence $(y,u) \leqslant_x (z,v)$.

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

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

Lemma 6.36. *Let $(M, \leqslant)$ 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 ∂^k_l such that $\partial^k_l(f) = f \circ \delta^k_l$. This gives $S(M)$ the structure of a pre-cubical set. Moreover, this is non-self-linked.*

Proof. This is mostly a standard proof [34]. We just have to check that faces of singular n-dicubes are still singular n-dicubes, i.e. that they verify the extra-condition that they are non-self-linked. This is straightforward. □

Similarly,

Lemma 6.37. *Let $(M, \leqslant)$ and $(N, \leqslant)$ be two locally partially ordered topological spaces and let $f : M \to N$ be a dimap. Then $S(f) : S(M) \to S(N)$ defined by, for all $x : \square_n \to M \in S$, $f(x) = f \circ x : \square_n \to N$, is a map of pre-cubical sets. S defines a functor from the category of locally partially ordered topological spaces to the (full sub-) category of (non-self-linked) pre-cubical sets.*

Proof. This is obvious (the composition of dimaps is a dimap). □

Proposition 6.38. $|.|$ *is left-adjoint to S.*

Proof. We prove that there exist two natural transformations

$$\eta : Id \to S(|.|),$$

$$\varepsilon : |S| \to Id$$

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

$$S \xrightarrow{\eta S} S(|S|) \xrightarrow{S\varepsilon} S\,,$$

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

are the identities.

We can first show that

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

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

in a natural manner for all M pre-cubical set 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, $Id : \square_n \to (x, \square_n)$. These in turn induce injective morphisms $f_x : \square_n \to |M|$, 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 subpre-cubical set 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 $f : \square_n \to X$. Now, $|S(X)|$ is an amalgamated sum of $(x, \square_n)$, $x \in S(X)_n$. The x induce on $\coprod_x (x, \square_n)$ and then on $|S(X)|$ an injective morphism in the category of local po-spaces. It is an easy exercise 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. □

6.8. Combinatorial dihomotopy

We introduce combinatorial dihomotopy and prove that a combinatorial dihomotopy of dipaths gives rise to an ordinary dihomotopy in the geometric realization. The other implication that dihomotopic geometric realizations of combinatorial dipaths are in fact combinatorially dihomotopic is also true, but is not proved in this paper.

Definition 6.39. Let N be a cubical set. A dipath in N is any sequence $p = (p_1, \ldots, p_k)$ of elements of N_1 such that for all i, $1 \leqslant i < k$, $\partial_1^1(p_i) = \partial_1^0(p_{i+1})$. $\partial_1^0(p_1)$ is the initial point of p. $\partial_1^1(p_k)$ is the final point of p.

Definition 6.40. Let N be a cubical set 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) $p = (p_1, \ldots, p_k)$ and $q = (q_1, \ldots, q_k)$,

(2) for all i, $1 \leqslant i < j$, $p_i = q_i$, and for all i, $j+1 < i \leqslant k$, $p_i = q_i$,
(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 6.41. • *Any combinatorial dipath p in M induces a (topological) dipath $|p|$ in $|M|$.*

• *Any combinatorial dihomotopy between two paths p and q in M induces a (topological) dihomotopy between $|p|$ and $|q|$.*

Proof. Let $p = (p_1, \ldots, p_k)$ be a dipath in N. Let $x : I \to |N|$ defined by, $\forall i$, $0 \leqslant i \leqslant k-1$, $\forall t \in I$, $\frac{i}{k} \leqslant t \leqslant \frac{i+1}{k}$, $x(t) = (p_{i+1}, k(t - \frac{i}{k})) \in |N|$.

Then x is a dimap (the local partial order in $|N|$ being defined as in Theorem 6.23),

- around each point in $(p_i, \overset{\circ}{\square}_1)$ (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 $\frac{i}{k} < t < \frac{i+2}{k}$, $x(t) \in St(\partial_1^0(p_{i+1}), \square_0) = St(\partial_1^1(p_i), \square_0)$.

We set $|p| = x$.

It is enough to prove now that if p and q are elementary dihomotopic, then $|p|$ and $|q|$ are dihomotopic. We suppose that we have $A \in N_2, k, l \in \mathbb{N}$ such that $p = (p_1, \ldots, p_k), q = (p_1, \ldots, p_{l-1}, q_l, q_{l+1}, p_{l+2}, \ldots, p_k)$ 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 $0 \leqslant t \leqslant \frac{l-1}{k}$ and $\frac{l+1}{k} \leqslant t \leqslant 1$, $H(\alpha, t) = |p|(t) = |q|(t)$,
- for $\frac{l-1}{k} \leqslant t \leqslant \frac{l}{k}$, $H(\alpha, t) = (A, (\alpha(kt - l + 1), (1-\alpha)(kt - l + 1))) \in |N|$,
- for $\frac{l}{k} \leqslant t \leqslant \frac{l+1}{k}$, $H(\alpha, t) = (A, ((1-\alpha)(kt - l) + \alpha, \alpha(kt - l) + 1 - \alpha)) \in |N|$.

Then H is the desired dihomotopy between $|p|$ and $|q|$. □

For the converse to this proposition, see [19].

7. Two-phase locking is safe: a dihomotopy proof

7.1. Introduction

Let us take a simple example, first given in [48]. Consider a distributed database in which transactions T_1 to T_n access shared variables $a, b, \ldots$ using locks: *Pa* to lock the exclusive access to a and *Va* to unlock a so that other transactions can use a. This is the same language that we used up to now.

Let us now have a look at the so-called "serialization" problem. Consider the following two transactions `R` and `S` put in parallel, where we have put actual assignments of the shared variables, when properly locked:

```
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-
```

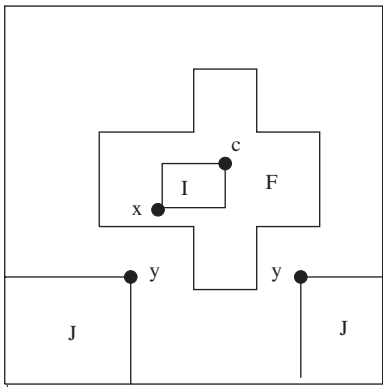


Fig. 12. Sets F, $I(\mathbf{x}, \mathbf{c})$, $K(\mathbf{y}, \mathbf{c})$.

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*.

A solution is given by the two-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;
```

It is well-known, that such a "two-phase locking" protocol is a scheduling strategy that ensures that a concurrent program has the same effect as a serial execution of the individual programs. We will present a geometric proof of the serializability of the two-phase protocol in the spirit of Gunawardena [48], but directly based on the directed homotopy theory concepts we have developed.

The "two-phase locking" protocol is a scheduling strategy that ensures that a concurrent program has the same effect as a serial execution of the individual programs as explained above.

The aim of this section is twofold. First of all, we want to give a modification of Gunawardena's reasoning in the framework of the present paper. Our proof is certainly more technical, but it seems to have several advantages: first of all, we avoid Gunawardena's "wobbling" problems (cf. [48, 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 $a \geqslant 1$ of transactions can acquire a lock to the same shared object at the same time.

After this paper had been written, Grandis proved in [46] an oriented version of the Seifert–van Kampen theorem (see also a weaker version, proven by the third author in [41]). This theorem allows, at least in principle, to calculate the dihomotopy sets $\pi_1(X; x_0, x_1)$ between certain start and end points from the dihomotopy sets of pieces of X and to their intersections. Our proof relies on a clever choice of such pieces to make these calculations work in practice.

7.2. Blockwise starshaped sets

First, we modify the concept of a "star-shaped" set in a vector space (used in [48]) in the presence of a partial order:

Definition 7.1. (1) For $x, c \in \mathbf{R}$ let $I(x, c)$ denote the interval $[x, c] \cup [c, x]$ between x and c.

(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. 12.

(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 7.2. (1) An n-cube R (n-rectangle in [21]) is blockwise starshaped with respect to every point in R. A Euclidean ball $\{\mathbf{x} \in \mathbf{R}^n || \mathbf{x} - \mathbf{c}| \leqslant r\}, r > 0$, is blockwise starshaped only with respect to its centre $\mathbf{c}$. The triangle $T = \{(x_1, x_2) \in \mathbf{R}^2 | x_i \geqslant 0, x_1 + x_2 \leqslant 1\}$ is starshaped with respect to every of its points. It is blockwise starshaped only with respect $(0, 0)$.

(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 non-empty "central" intersection if it is a model of a two-phase locked transaction system.

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

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

(2) For $\mathbf{x} = [x_1, \ldots, x_n], \mathbf{c} = [c_1, \ldots, c_n] \in \mathbf{R}^n$ let $K(\mathbf{x}, \mathbf{c}) = \prod K(x_i, c_i)$, cf. Fig. 12.

Lemma 7.4. *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 $K(\mathbf{y}, \mathbf{c}) \subset X$ for every $\mathbf{y} \in X$.*

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

7.3. Partitions and contractions

Suppose $\mathbf{c} \in \overset{\circ}{F} \subset I^n$ and F is 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 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 K-sets above as follows:

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

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

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-cubes $I_\Delta = I_{\delta_1 \ldots \delta_n} = \prod I^i_{\delta_i}$. There is an obvious partial order between those subcubes giving rise to $n!$ directed paths from $I_{-1\ldots-1}$ to $I_{1\ldots1}$. We need the following subsets of I_Δ:

Definition 7.5. Given $\Delta \in \{-1, 1\}^n$ with $\delta_i = 1$ and $\delta_j = -1$. Then

(1) $I^{i,j}_\Delta = \{\mathbf{x} \in I_\Delta | (x_i, x_j) \neq (0, 0)\}$.

(2) $I^{i,j}_{\Delta 0} = \{\mathbf{x} \in I^{i,j}_\Delta | x_i = 0\}$, $I^{i,j}_{\Delta 1} = \{\mathbf{x} \in I^{i,j}_\Delta | x_j = 0\}$.

(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 $\mathbf{p}^{i,j}_{\Delta 0}$ and $\mathbf{p}^{i,j}_{\Delta 1}$ with
$(\mathbf{p}^{i,j}_{\Delta *})_k = \delta_k,\ k \neq i, j,\ (\mathbf{p}^{i,j}_{\Delta 0})_i = 0,\ (\mathbf{p}^{i,j}_{\Delta 0})_j = -1, (\mathbf{p}^{i,j}_{\Delta 1})_i = 1,\ (\mathbf{p}^{i,j}_{\Delta 1})_j = 0.$

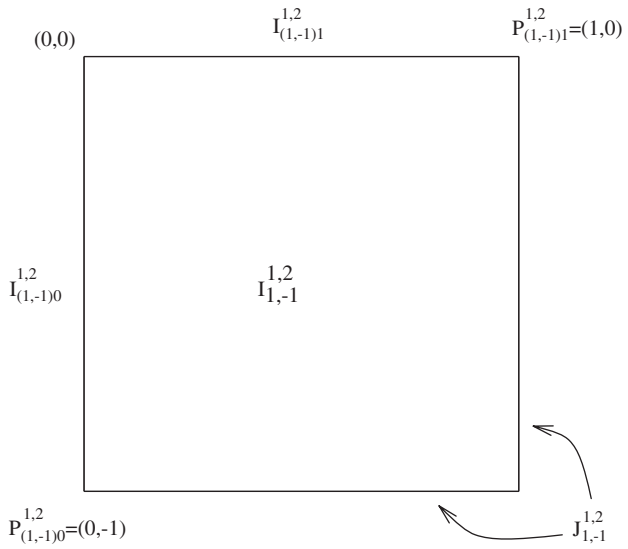


Fig. 13. Subrectangle with faces.

In fact, $I^{i,j}_{\Delta}$ is one of the 2^n sub-n-cubes mentioned above with a 2-codimensional face removed; $I^{i,j}_{\Delta 0}$ and $I^{i,j}_{\Delta 1}$ represent two of its faces; $J^{i,j}_{\Delta}$ represents a (totally ordered) 1-complex in its boundary from $\mathbf{p}^{i,j}_{\Delta 0}$ to $\mathbf{p}^{i,j}_{\Delta 1}$ (Fig. 13).

Next comes a definition of a *dimap*, cf. Definition 3.7,

$$\Phi^{\Delta,i,j} = (\Phi^{\Delta,i,j}_1, \ldots \Phi^{\Delta,i,j}_n) : (I^{i,j}_{\Delta}; I^{i,j}_{\Delta 0}, I^{i,j}_{\Delta 1}) \to (J^{i,j}_{\Delta}, \mathbf{p}^{i,j}_{\Delta 0}, \mathbf{p}^{i,j}_{\Delta 1}).$$

Definition 7.6. (1) $\Phi^{\Delta,i,j}_k(\mathbf{x}) = \delta_k$ for $k \neq i, j$.

(2) $\Phi^{\Delta,i,j}_i(\mathbf{x}) = \frac{x_i}{-x_j}$ and $\Phi^{\Delta,i,j}_j(\mathbf{x}) = -1$ for $x_i \leqslant -x_j$.

(3) $\Phi^{\Delta,i,j}_i(\mathbf{x}) = 1$ and $\Phi^{\Delta,i,j}_j(\mathbf{x}) = \frac{x_j}{x_i}$ for $x_i \geqslant -x_j$.

There is a directed version of deformation retracts (cf. [16]) in the po-space environment:

Definition 7.7. Let $A \subset X$ denote an inclusion of two po-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: $H_t|A = id,\ t \in I$.

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

Proposition 7.8. (1) *The* 1*-dimensional subcomplex* $J^{i,j}_{\Delta}$ *is a strong deformation di-retract of* $I^{i,j}_{\Delta}$.

(2) *Moreover,* $J^{i,j}_{\Delta} \cap X = J^{i,j}_{\Delta} \setminus F$ *is a strong deformation di-retract of* $I^{i,j}_{\Delta} \cap X = I^{i,j}_{\Delta} \setminus F$.

(3) *Let* $\Delta = (\delta_1, \ldots, \delta_n)$ *and* $\Delta' = (\delta'_1, \ldots, \delta'_n)$ *with* $\delta'_k = \delta_k$ *for* $k \neq j,\ \delta_j = -1,\ \delta'_j = 1,\ \delta_l = \delta'_l = -1$ *denote two successive subcubes. Then, the maps* $\Phi^{\Delta,i,j}$ *and* $\Phi^{\Delta',j,l}$ *agree on the intersection of their domains of definition. Likewise for the homotopies in the deformation retractions from* (1) *above.*

Proof. (1) The dimap property depends only on the (x_i, x_j)-coordinates. In the projection on the (x_i, x_j)-plane, the map $\Phi^{\Delta,i,j}$ "stretches" every oriented wedge (like the stipled one in Fig. 14) 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 the restrictions of $\Phi^{\Delta,i,j}$ to points "under", resp. "over" that antidiagonal are dimaps and that $\Phi^{\Delta,i,j}$ fixes $J^{\Delta,i,j}$.

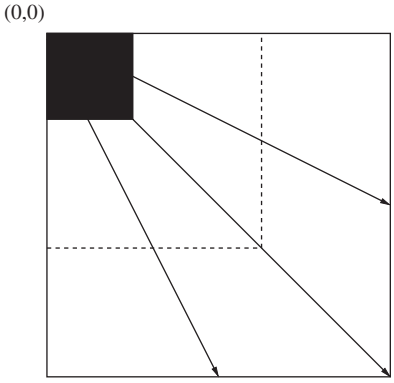


Fig. 14. The dimap $\Phi^{\Delta,i,j}$.

The (linear) 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 $\Phi^{\Delta,i,j}(x)$ along a line; in particular, all of the maps H_t are di-maps.

(2) We have to show that the $\Phi^{\Delta,i,j}$ preserves $I_{\Delta}^{i,j} \setminus F$. By definition, $\Phi_k^{\Delta,i,j}(\mathbf{x}) \geqslant x_k$ for $\delta_k = 1$ and $\Phi_k^{\Delta,i,j}(\mathbf{x}) \leqslant x_k$ for $\delta_k = -1$. Use Lemma 7.4. Similarly for the dihomotopy $H^{i,j}$.

(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 $\Delta_- = (-1,\ldots,-1)$ and $\Delta_+ = (1,\ldots,1)$. Let $I_{\Delta_\pm}^i = \{\mathbf{x} \in I_{\Delta_\pm} | x_k = \pm 1,\ k \neq i;\ x_i = 0\}$, $J_{\Delta_\pm}^i = \{\mathbf{x} \in I_{\Delta_\pm} | x_k = \pm 1,\ k \neq i\}$, and $(\mathbf{p}_{\Delta_\pm}^i)_k = \pm 1,\ k \neq i,\ (\mathbf{p}_{\Delta_\pm}^i)_i = 0$. Then, we define

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

Proposition 7.9. *The analogue of Proposition* 7.8 *holds in these two cases as well.*

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

$$I_\sigma = I_{\Delta-} \cup \bigcup_{k=1}^{n-1} I_{\Delta_\sigma(k)}^{\sigma(k),\sigma(k+1)} \cup I_{\Delta+}, \quad J_\sigma = J_{\Delta-}^{\sigma(1)} \cup \bigcup_{k=1}^{n-1} J_{\Delta_\sigma(k)}^{\sigma(k),\sigma(k+1)} \cup J_{\Delta+}^{\sigma}(n) \tag{2}$$

denote a union of $(n+1)$ subcubes (without certain 2-dimensional faces) in an ascending chain, resp. a *totally ordered* 1-dimensional subcomplex in the boundary. Glueing the dimaps and dihomotopies from Propositions 7.8 and 7.9 together, we obtain

Corollary 7.10. (1) *The* 1*-complex J_σ is a strong deformation di-retract of the complex I_σ.*

(2) *If $\mathbf{0} \in \overset{\circ}{F}$ 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 7.11. A po-space with a totally ordered 1-subcomplex (dihomeomorphic to an interval) as a strong deformation di-retract is the analogue to a *contractible* space in ordinary topology. Hence, one might call such a po-space *dicontractible*. The strategy of the proof was thus to subdivide the underlying po-space into dicontractible pieces with

control on the intersection. Applying Grandis oriented version [46] of the Seifert–van Kampen theorem (with the chosen subdivision) makes it in fact possible to calculate the *fundamental category* [46,43] of the space $X = I^n \setminus F$.

7.4. Application to dipaths and serializability

As an application, we obtain the following result about dipaths generalizing Gunawardena's result [48]. Remark that the forbidden region in a process graph is blockwise starshaped with respect to a central point according to Example 7.2. Remark also, that a dipath on the 1-skeleton of the boundary of a hypercube I^n corresponds exactly to a serial execution.

Theorem 7.12. *Let $F \subset I^n = [-1, 1]^n$ such that $\mathbf{0} \in \overset{\circ}{F}$ and F is blockwise starshaped with respect to $\mathbf{0}$. Every dipath in $X = I^n \setminus F$ from $-\mathbf{1} = (-1, \ldots, -1)$ to $\mathbf{1} = (1, \ldots, 1)$ is dihomotopic to a dipath on the 1-skeleton $(\partial I^n)_1$ of the boundary $\partial I^n = \{(x_1, \ldots, x_n) | \ \exists k : x_i = \pm 1, \ i \neq k\}$ of I^n.*

Proof. Let $\Sigma = \{(x_1, \ldots, x_n) \in I^n | \ x_i = x_j = 0 \text{ for some } 1 \leqslant i < j \leqslant n\} \subset I^n$ 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 $J_\sigma \subset (\partial I^n)_1$ from $-\mathbf{1}$ to $\mathbf{1}$ by Corollary 7.10.

It remains to handle dipaths intersecting Σ. Since a dipath has dimension 1 and the singular set Σ has codimension 2, 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, cf. Section 6. If X is the "forbidden region" corresponding to "mutual exclusion" in a process graph (cf. [48,21]), the subdivision will have that non-intersection property by construction. □

8. Open mathematical problems

This paper has the purpose of stating some of the most basic definitions and properties in directed homotopy theory. Since a preprint version of this paper appeared in 1999, a lot of progress has been made, but there are still a lot of challenging problems. Let us mention some of them, with pointers to the literature that has appeared meanwhile:

(M1) (Local) po-spaces do not right away have nice categorical properties. Cartesian closedness is important. Several alternative ways have been proposed, notably the category of flows by Gaucher [28] and the category of d-spaces by Grandis [46]. Gaucher has also defined and investigated relations between flows that capture the analogue of homotopy equivalence in the classical setting.

(M2) The dicomponents of this paper have obtained a more solid foundation starting from the fundamental *category* (dipaths modulo dihomotopy) of a po-space in which a certain system of morphisms gets inverted. In particular, the dicomponents are interpreted as the objects of a category, which can be viewed as a graph with commutation relations. For definitions and properties, cf. [43,68,20]. The investigation and exploitation of naturality properties for these categories presents still a major challenge.

(M3) Compositionality is also a major mathematical issue: suppose X is a po-space and that the fundamental categories (cf. (M2)) are well-understood for the pieces A, B and their intersection $A \cap B$. What can one infer about the fundamental category of X? What about the categories of dicomponents? The first question has been solved by Grandis [46] in a Seifert–van Kampen theorem for d-spaces, cf. (M1) above, and by Goubault, under some restricted circumstances, for local po-spaces in [41]. Consequences for the dicomponents (M2) still have to be sorted out.

(M4) How can one define directed *homology* and exploit it for calculations? What are the relations between dihomotopy and dihomology? Here also, a categorical framework has to be established. This is work in progress; for a very first attempt cf. [34]. More refined theories have been mostly defined by Gaucher, see for instance [27] or [30,29].

(M5) *Higher* dihomotopy (with a suitable structure) has to be investigated more closely, having a potential of giving more subtle invariants of lpo-spaces. For some first attempts, cf. [76,43,68].

(M6) Many po-spaces arising from concurrency set-ups are equipped with a certain degree of *symmetry*. Hence, a group equivariant dihomotopy theory ought to be established.

9. 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 new results might be derived from this theory, e.g.:

(CS1) How can we exploit (M2) so that we can derive the "essential schedules" of a concurrent system? 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 [21]. A generalization would be important since these schedules describe fine (safety) properties of concurrent systems (about the possible orderings on accesses to shared resources for instance) that for instance encompass serializability issues. (M3) would make it possible to attack 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. Since we wrote this paper, hints about how to do this have now been published, see in particular [44,20,68].

(CS2) (M4) and (M5) would make it possible to consider more refined properties of fault-tolerant systems and make a direct link with the work of Herlihy, Rajsbaum and Shavit. Basically the aim is to give the semantic foundations to a computability and complexity theory for fault-tolerant distributed systems. (M1) would help to describe the basic fundamental synchronization models that one can imagine for concurrent systems.

(CS3) (M4) would make it possible to have good algorithms (from linear algebra) giving compatible information about schedules of concurrent programs. This was already hinted in [35].

(CS4) It is an old question [77] as to which monoids can be represented by finite canonical term rewriting systems, which is quite related to our notions and techniques. More generally, we could ask what can be computed in more general structures [62]. This is very much linked to (CS2).

(CS5) We think that ideas from this framework might help to design better algorithms for distributed databases schedulers (like better "path-pushing" algorithms) and for micro-instructions schedulers.

10. Related work

A typical problem in concurrency theory and in distributed systems theory is to distinguish non-determinism from "true" concurrency. In some cases, we would like to be able to specify the actual use of shared resources of a parallel program, like, how many processors are busy or idle, or should a process wait for a shared variable? As can be seen in Fig. 15, 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, in ordinary transition systems semantics. 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. Our solution is to explicitly fill in the boundary of a square if we want to specify true concurrency instead of mutual exclusion, or to fill in the boundary of a cube when we want to specify that three processes can be active at the same time and not just two etc.

A first solution has been proposed in slightly different forms, asynchronous transition systems [4], concurrent automata [78], transition systems with independence [81], 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 about more complex mutual exclusion properties, like shared resources that one can access in parallel n times but not $n+1$ ($n \geqslant 2$), 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. This is dealt with in the notion of a counting semaphore, initialized to 2 (or 2-semaphore, i.e. that can be shared by 2 but not by 3 processes at a time). The geometric semantics of such an object is pictured in Fig. 16.

These points are actually crucial in a 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

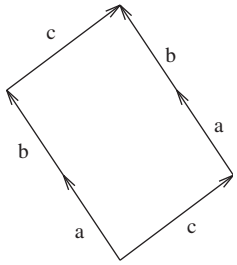


Fig. 15. Interleaving of A and c.

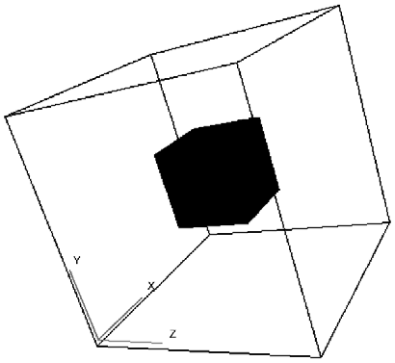


Fig. 16. A 2-semaphore.

busy processors is of primary importance (see Sections 9 and 10), or for optimization of the use of shared resources.

This was first proposed in [65,32], and further treated in [34]. 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).

Most of these aspects are dealt with traditionally 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 for transition systems [2].

Slightly different geometric models have been used in the work of Herlihy, Shavit and Rajsbaum on fault-tolerant protocols for distributed systems. This has given numerous results in the field of distributed systems. It is proved for instance in [53] that in a shared-memory model with single reader/single 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 [54]). We refer the reader to other articles in this area, in particular, the book on distributed algorithms [60], other articles by Herlihy et al. [50–52], some slightly different methods, still geometric in nature, in [3,5,6,11,24] (which originated this field of research, starting with graph theoretical arguments), [69]. Several ideas about classifying data structures according to what protocols they manage to solve are described in [56,57,71]. This should be related to problem (CS2). Some links with directed homotopy have been hinted by one of the co-authors, see [37–39].

In concurrent databases, we refer the reader to the very nice introductory paper of Gunawardena [48]. An application of this theory to the problem of deadlock detection has been made by the authors in [21].

There are other scheduling problems of interest for the theory we developed in this paper, like the ones appearing due to the architecture of recent microprocessors. Let us take an example from [49,66]. 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

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

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

Fig. 17. MIPS R4000 floating point unit.

be executed (otherwise "structural hazards" occur) within a certain number of cycles after some others (see Fig. 17 [1]). 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 [66] 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 formalized in [1], see below). 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. [2]

We see from Fig. 17 that if we suppose that we want to execute two instructions add.s one after the other on the MIPS R4000 floating point unit, 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.

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

On the semantic side, there is a strong link between po-spaces and progress graphs [10,15]. The model presented in [65,32] originated part of this work, and most of the work of one of the co-authors, [42,33,34] and also [59]. There are also links with the homological considerations of [25,26]. Some potentially related semantic models are the n-categorical formulations of [9] and some other combinatorial or categorical formulations in [23,70,73–75].

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 [35] (where some other ideas for program analysis are exemplified) and also fully treated for CCS in [79,80]. A prototype Parallel Pascal Analyser has been implemented by Cridlig (http://www.dmi.ens.fr/∼cridlig); its principles are 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 [22,47] (with the specific use of cubical sets), [58,77]. An extension to homology of categories has been proposed in [62]. Also cubical sets as such have been used in [72] and studied combinatorially and categorically in [7,8].

References

[1] H. Attiya, R. Friedman, A correctness condition for high-performance multiprocessors, in: Proc. 24th STOC, ACM Press, New York, 1992.
[2] A. Arnold, Systèmes de transitions finis et sémantique des processus communicants, Masson, 1992.
[3] H. Attiya, A. Bar-Noy, D. Dolev, D. Peleg, R. Reischuk, Renaming in an asynchronous environment, J. ACM 37 (3) (1990) 524–548.
[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, Technical Report, University of California in Los Angeles, 1995.
[6] E. Borowsky, E. Gafni, Generalized FLP impossibility result for t-resilient asynchronous computations, in: Proc. 25th STOC, ACM Press, New York, 1993.
[7] R. Brown, P.J. Higgins, On the algebra of cubes, J. Pure Appl. Algebra 21 (1981) 233–260.
[8] R. Brown, P.J. Higgins, Colimit theorems for relative homotopy groups, J. Pure Appl. Algebra 22 (1981) 11–41.
[9] R. Buckland, M. Johnson, ECHIDNA: a system for manipulating explicit choice higher dimensional automata, in: AMAST'96: Fifth Internat. Conf. on Algebraic Methodology and Software Technology, Munich, 1996.
[10] S.D. Carson, P.F. Reynolds Jr., The geometry of semaphore programs, ACM Trans. Programming Languages Systems 9 (1) (1987) 25–53.

[1] Taken from [66].
[2] It has two integer arithmetic units.

[11] S. Chaudhuri, Agreement is harder than consensus: set consensus problems in totally asynchronous systems, in: Proc. Ninth Annu. ACM Symp. on Principles of Distributed Computing, ACM Press, New York, 1990, pp. 311–334.
[12] R. Cridlig, Semantic analysis of shared-memory concurrent languages using abstract model-checking, in: Proc. PEPM'95, ACM Press, La Jolla, 1995.
[13] R. Cridlig, Semantic analysis of concurrent ML by abstract model-checking, in: Proc. LOMAPS Workshop, 1996.
[14] R. Cridlig, E. Goubault, Semantics and analyses of Linda-based languages, in: Proc. WSA'93, Lecture Notes in Computer Science, Vol. 724, Springer, Berlin, 1993.
[15] E. Dijkstra, Cooperating Sequential Processes, Academic Press, New York, 1968.
[16] A. Dold, Lectures on Algebraic Topology, second ed., Grundlehren der Matematischen Wissenschaften, Vol. 200, Springer, Berlin, 1980.
[17] U. Fahrenberg, The geometry of timed PV-programs, Electronic Notes in Computer Science, Vol. 81, 2003, pp. 1–14.
[18] L. Fajstrup, Dicovering spaces, Homology Homotopy Appl. 5 (2) (2003) 1–17.
[19] L. Fajstrup, Dipaths and dihomotopies in a cubical complex, Adv. Appl. Math. 35 (2) (2005) R–2003–22.
[20] L. Fajstrup, E. Goubault, E. Haucourt, M. Raussen, Components of the fundamental category, Appl. Categ. Structures 12 (1) (2004) 81–108.
[21] L. Fajstrup, E. Goubault, M. Raussen, Detecting deadlocks in concurrent systems, in: D. Sangiorgi, R. de Simone (Eds.), CONCUR '98; Concurrency Theory, Lecture Notes in Computer Science, Vol. 1466, Springer, Nice, France, Ninth Internat. Conf., Proceedings, 1998, pp. 332–347.
[22] D.R. Farkas, The Anick resolution, J. Pure Appl. Algebra 79 (1992) 159–168.
[23] M. Fiore, G. Plotkin, J. Power, Complete cuboidal sets in axiomatic domain theory (extended abstract), in: Proc. 12th Annu. IEEE Symp. on Logic in Computer Science, IEEE Computer Society Press, Warsaw, Poland, 1997, pp. 268–279.
[24] M. Fisher, N.A. Lynch, M.S. Paterson, Impossibility of distributed commit with one faulty process, J. ACM 32 (2) (1985) 374–382.
[25] P. Gaucher, Connexion de flux d'information en algèbre homologique, Technical Report, IRMA, Strasbourg, available at ⟨http://irmasrv1.u-strasbg.fr/∼gaucher/activite.html⟩, 1997.
[26] P. Gaucher, Etude homologique des chemins de dimension 1 d'un automate, Technical Report, IRMA, Strasbourg, available at ⟨http://irmasrv1.u-strasbg.fr/∼gaucher/activite.html⟩, 1997.
[27] P. Gaucher, Homotopy invariants of higher dimensional categories and concurrency in computer science, Math. Structures Comput. Sci. 10 (4) (2000) 481–524.
[28] P. Gaucher, A convenient category for the homotopy theory of concurrency, Technical Report, math.AT/0201252, 2002, 138pp.
[29] P. Gaucher, From concurrency to algebraic topology, Electronic Notes in Computer Science, Vol. 39 (2), 2001, pp. 1–19.
[30] P. Gaucher, E. Goubault, Topological deformation of higher dimensional automata, Technical Report, arXiv:math.AT/010760, 2001.
[31] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, D.S. Scott, A Compendium of Continuous Lattices, Springer, Berlin, Heidelberg, New York, 1980.
[32] 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.
[33] E. Goubault, Domains of higher-dimensional automata, in: Proc. CONCUR'93, Springer, Hildesheim, 1993.
[34] E. Goubault, The geometry of concurrency, Ph.D. Thesis, Ecole Normale Supérieure Paris, available at ⟨http://www.dmi.ens.fr/∼goubault⟩, 1995.
[35] E. Goubault, Schedulers as abstract interpretations of HDA, in: Proc. PEPM'95, ACM Press, New York, also available at ⟨http://www.dmi.ens.fr/∼goubault⟩, La Jolla, 1995.
[36] E. Goubault, Durations for truly-concurrent actions, in: Proc. ESOP'96, no. 1058, Springer, Berlin, 1996, pp. 173–187.
[37] E. Goubault, The dynamics of wait-free distributed computations, Technical Report, Research Report LIENS-96-26, December 1996.
[38] E. Goubault, A semantic view on distributed computability and complexity, in: Proc. Third Theory and Formal Methods Section Workshop, Imperial College Press, also available at ⟨http://www.dmi.ens.fr/∼goubault⟩, 1996.
[39] E. Goubault, Optimal implementation of wait-free binary relations, in: Proc. 22nd CAAP, Springer, New York, 1997.
[40] E. Goubault, Cubical sets are generalized transition systems, Technical Report, Pre-proceedings of CMCIM'02, also available at ⟨http://www.di.ens.fr/∼goubault⟩, 2001.
[41] E. Goubault, Some geometric perspectives in concurrency theory, Homology Homotopy Appl. 5 (2) (2003) 95–136.
[42] E. Goubault, T.P. Jensen, Homology of higher-dimensional automata, in: Proc. CONCUR'92, Springer, Stonybrook, New York, 1992.
[43] E. Goubault, M. Raussen, Dihomotopy as a tool in state space analysis, in: S. Rajsbaum (Ed.), LATIN 2002: Theoretical informatics, Lecture Notes in Computer Science, Vol. 2286, Springer, Cancun, Mexico, 2002, pp. 16–37.
[44] E. Goubault, M. Raussen, Dihomotopy as a tool in state space analysis, in: Proc. LATIN'02, 2002.
[45] M. Grandis, Directed homotopy theory II. Homotopy constructs, Theory Appl. Categ. 10 (14) (2002) 369–391.
[46] M. Grandis, Directed homotopy theory I. The fundamental category, Cahiers Topologie Géom. Différentielle Catégoriques (2003) 281–316.
[47] J.R.J. Groves, Rewriting systems and homology of groups, in: L.G. Kovacs (Ed.), Groups—Canberra 1989, Lecture Notes in Mathematics, Vol. 1456, Springer, Berlin, 1991, pp. 114–141.
[48] J. Gunawardena, Homotopy and concurrency, Bull. EATCS 54 (1994) 184–193.
[49] E. Harcourt, J. Mauney, T. Cook, From processor timing specifications to static instruction scheduling, in: Proc. Static Analysis Symp.'94, Lecture Notes in Computer Science, Springer, Berlin, 1994.
[50] M. Herlihy, A tutorial on algebraic topology and distributed computation, Technical Report, Presented at UCLA, 1994.
[51] M. Herlihy, S. Rajsbaum, Set consensus using arbitrary objects, in: Proc. 13th Annu. ACM Symp. on Principles of Distributed Computing, ACM Press, New York, 1994.
[52] M. Herlihy, S. Rajsbaum, Algebraic topology and distributed computing, a primer, Technical Report, Brown University, 1995.
[53] M. Herlihy, N. Shavit, The asynchronous computability theorem for t-resilient tasks, in: Proc. 25th STOC, ACM Press, New York, 1993.

[54] M. Herlihy, N. Shavit, A simple constructive computability theorem for wait-free computation, in: Proc. STOC'94, ACM Press, New York, 1994.
[55] J. Jardine, Cubical homotopy theory: a beginning, Technical Report NI02030-NST, Isaac Newton Institute, Cambridge, UK, 2002.
[56] P. Jayanti, On the robustness of Herlihy's hierarchy, in: Proc. 12th Annu. ACM Symp. on Principles of Distributed Computing, Ithaca, New York, USA, 1993, pp. 145–157.
[57] P. Jayanti, Robust wait-free hierarchies, J. ACM 44 (4) (1997) 592–614.
[58] Y. Kobayashi, Complete rewriting systems and homology of monoid algebras, J. Pure Appl. Algebra 65 (1990) 263–275.
[59] E. Lanzmann, Automates d'ordre supérieur, Master's Thesis, Université d'Orsay, 1993.
[60] N. Lynch, Distributed Algorithms, Morgan Kaufmann, Los Altos, CA, 1996.
[61] A. Mazurkiewicz, Basic notions of trace theory, in: Lecture Notes for the REX Summer School in Temporal Logic, Springer, Berlin, 1988.
[62] F. Morace, Finitely presented categories and homology, Technical Report, Université Joseph Fourier, 1995.
[63] L. Nachbin, Topology and Order, Van Nostrand, Princeton, 1965.
[64] R. Penrose, Techniques of Differential Topology in Relativity, Conference Board of the Mathematical Sciences, Vol. 7, Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, USA, 1972.
[65] V. Pratt, Modeling concurrency with geometry, in: Proc. 18th ACM Symp. on Principles of Programming Languages, ACM Press, New York, 1991.
[66] T.A. Proebsting, C.W. Fraser, Detecting pipeline structural hazards quickly, in: Proc. Symp. on Principles of Programming Languages, ACM Press, New York, 1994.
[67] M. Raussen, On the classification of dipaths in geometric models for concurrency, Math. Structures Comput. Sci. 10 (4) (2000) 427–457.
[68] M. Raussen, State spaces and dipaths up to dihomotopy, Homology Homotopy Appl. 5 (2) (2003) 257–280.
[69] M. Saks, F. Zaharoglou, Wait-free k-set agreement is impossible: the topology of public knowledge, in: Proc. 25th STOC, ACM Press, New York, 1993.
[70] V. Sassone, G.L. Cattani, Higher-dimensional transition systems, in: Proc. LICS'96, 1996.
[71] E. Schenk, The consensus hierarchy is not robust, in: Proc. 16th Annu. ACM Symp. on Principles of Distributed Computing, Santa Barbara, California, 1997, p. 279.
[72] J. Serre, Homologie singulière des espaces fibrés applications, Ph.D. Thesis, Ecole Normale Supérieure, 1951.
[73] S. Sokolowski, Homotopy in concurrent processes, Technical Report, Institute of Computer Science, Gdansk Division, 1998.
[74] S. Sokolowski, Investigation of concurrent processes by means of homotopy functors, Technical Report, Institute of Computer Science, Gdansk Division, 1998.
[75] S. Sokolowski, Point glueing in cpo-s, Technical Report, Institute of Computer Science, Gdansk Division, 1998.
[76] S. Sokołowski, Classifying holes of arbitrary dimension in partially ordered cubes, Kansas State University, February 2000, Manuscript.
[77] C.C. Squier, F. Otto, Y. Kobayashi, A finiteness condition for rewriting systems, Theoret. Comput. Sci. 131 (1994) 271–294.
[78] A. Stark, Concurrent transition systems, Theoret. Comput. Sci. 64 (1989) 221–269.
[79] Y. Takayama, Towards Cycle Filling as Parallelization, RIMS Workshop on Concurrency Theory and Applications 96, RIMS Kyoto University, 1998.
[80] Y. Takayama, Extraction of concurrent processes from higher-dimensional automata, in: Proc. CAAP'96, 1996, pp. 72–85.
[81] G. Winskel, M. Nielsen, Models for concurrency, Handbook of Logic in Computer Science, Vol. 3, Oxford University Press, Oxford, 1994, pp. 100–200.