BRICS NS-04-2 Cousot et al. (eds.): GETCO 2004 Preliminary Proceedings

Basic Research in Computer Science

Preliminary Proceedings of the Workshop on

Geometry and Topology in Concurrency and Distributed Computing

GETCO 2004

Amsterdam, The Netherlands, October 4, 2004

Patrick Cousot
Lisbeth Fajstrup
Eric Goubault
Maurice Herlihy
Martin Raußen
Vladimiro Sassone
(editors)

BRICS Notes Series **NS-04-2**

ISSN 0909-3206 **September 2004**

See back inner page for a list of recent BRICS Notes Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `NS/04/2/`

GEometry and Topology in COncurrency and Distributed Computing

Preliminary Proceedings of GETCO 2004
Satellite Workshop of DISC 2004
Amsterdam, The Netherlands, 4. October 2004

Patrick Cousot
Lisbeth Fajstrup
Eric Goubault
Maurice Herlihy
Martin Raußen
Vladimiro Sassone

Preliminary print under the patronage of
BRICS: Basic Research in Computer Science
Centre of the National Danish Research Foundation

Foreword

Eric Goubault[1]
LIST/DTSI/SLA, CEA Saclay
91191 Gif-sur-Yvette, France

The main mathematical disciplines that have been used in theoretical computer science are discrete mathematics (especially, graph theory and ordered structures), logics (mostly proof theory for all kinds of logics, classical, intuitionistic, modal etc.) and category theory (cartesian closed categories, topoi etc.). General Topology has also been used for instance in denotational semantics, with relations to ordered structures in particular.

Recently, ideas and notions from mainstream "geometric" topology and algebraic topology have entered the scene in Concurrency Theory and Distributed Systems Theory (some of them based on older ideas). They have been applied in particular to problems dealing with coordination of multi-processor and distributed systems. Among those are techniques borrowed from algebraic and geometric topology: Simplicial techniques have led to new theoretical bounds for coordination problems. Higher dimensional automata have been modelled as cubical sets with a partial order reflecting the time flows, and their homotopy properties allow to reason about a system's global behaviour.

This workshop aims at bringing together researchers from both the mathematical (geometry, topology, algebraic topology etc.) and computer scientific side (concurrency theorists, semanticians, researchers in distributed systems etc.) with an active interest in these or related developments.

It follows the first workshop on the subject "Geometric and Topological Methods in Concurrency Theory" which has been held in Aalborg, Denmark, in June 1999. Then came GETCO'00 in Pennstate, USA, GETCO'01 in Aalborg, Denmark, all associated with CONCUR. GETCO'02 was associated with DISC'02 in Toulouse, and GETCO'03 was held jointly with CMCIM'03, associated with CONCUR in Marseille. This year's GETCO'04 workshop is again associated with DISC, in Amsterdam.

The Workshop has been financially supported by the Basic Research Institute in Computer Science (Aarhus, Denmark), which I thank very warmly. I also wish to thank the referees, the authors and the programme committee members for their very precise and timely job. Many thanks are also due to Michael Mislove who kindly supported the workshop by letting us submit the papers through the Electronic Notes in Theoretical Computer Science.

In organizing the workshop—setting up the website, keeping track of the submissions, getting the preproceedings ready, etc.—Ulrich Fahrenberg and Emmanuel Haucourt have done an excellent job, and I thank them for that.

Last but not least, I wish to thank the organizers of DISC 2004, Jaap-Henk Hoepman, Paul Vitanyi, and Rachid Guerraoui, for their cooperation regarding this workshop.

[1]Email: `Eric.Goubault@cea.fr`

Table of Contents

On the Asynchronous Computability Theorem
(Early Draft)*

Rachid Guerraoui Petr Kouznetsov Bastian Pochon

Distributed Programming Laboratory
EPFL

Abstract

The impossibility of wait-free set agreement [1, 5, 8], can be seen as a corollary to a more general result: the characterization of asynchronous wait-free solvable tasks. Such a characterization, presented in the form of the *asynchronous computability theorem*, was first formulated and proved by Herlihy and Shavit [7], and then was reformulated and proved in a different manner by Borowsky and Gafni [3]. In fact, the characterization criterion defined in [3] turned out to be more convenient for impossibility results and, in particular, for the impossibility of wait-free set agreement. Interestingly, the criterion of [7] can be derived from the one of [3] through showing a purely geometrical result by means of distributed algorithms. Unfortunately, the characterization criterion is given in [3] just as a side result and the proof of the equivalence of the two criteria above is hard to digest. In this short note, we revisit the proofs of the characterization criteria of [3, 7] trying to bridge our personal "knowledge gap". We believe this note can be of interest for other researchers who try to apply geometrical methods in distributed computing.

1 Introduction

In 1993, Herlihy and Shavit proposed an elegant way to characterize *wait-free computable tasks* [5–7]. Essentially, they reduced the question of wait-free computability of a given decision task to verifying certain purely geometrical properties. They showed that any decision task T can be associated with a pair of geometrical structures, called *simplicial complexes*, an *input* complex $\mathcal{I}$ and *output* complex $\mathcal{O}$, and a relation $\Delta \subseteq \mathcal{I} \times \mathcal{O}$, the *task specification* of T, that carries every input vector to a non-empty set of output vectors. The following theorem from [7] gives necessary and sufficient conditions for a decision task to be wait-free solvable in the read-write memory model.

Theorem 1 *A decision task $(\mathcal{I}, \mathcal{O}, \Delta)$ is wait-free solvable using read-write memory if and only if there exists a chromatic subdivision σ of $\mathcal{I}$ and a color-preserving map $\mu : \sigma(\mathcal{I}) \to \mathcal{O}$ such that for each simplex $S \in \sigma(I)$, $\mu(S) \in carrier(S, \mathcal{I})$.*

Theorem 1 gives a geometrical criterion of wait-free solvability. A natural question comes: can we find an efficient algorithm to verify this criterion? Unfortunately, the answer is "no": the problem of wait-free computability is actually undecidable (for three or more participating processes) [4]. In other words, no wait-free solvability criterion would be efficient. So the question of whether the criteria of Theorem 1 is efficient *in general* is not theoretically interesting. However, it can be interesting to see whether the characterization is convenient with respect to some specific tasks in distributed computing.

In 1997, Borowsky and Gafni proposed an alternative criterion [3], formulated as follows:

*In 2004, the Gödel Prize for outstanding papers in the area of theoretical computer science was awarded to Maurice Herlihy and Nir Shavit for their paper "The Topological Structure of Asynchronous Computability" (J. ACM, 1999), and to Mike Saks and Fotios Zaharoglou for their paper "Wait-Free k-Set Agreement is Impossible: The Topology of Public Knowledge" (SIAM J. Computing, 2000). The very same result was simultaneously obtained by Elizabeth Borowsky and Eli Gafni [1], but it has never been published as a journal paper, which is a necessary condition for the Gödel Prize.

Theorem 2 *A decision task* $(\mathcal{I}, \mathcal{O}, \Delta)$ *is wait-free solvable using read-write memory if and only if there exists an iterated standard chromatic subdivision* χ^K *of* $\mathcal{I}$ *and a color-preserving map* $\mu : \chi^K(\mathcal{I}) \to \mathcal{O}$ *such that for each simplex* $S \in \chi^K(I)$, $\mu(S) \in$ *carrier*$(S, \mathcal{I})$.

Clearly, since no efficient criterion of wait-free solvability exists [4], these two criteria are computationally equivalent. However, the set of conditions in Theorem 2 is more restrictive: any chromatic subdivision of Theorem 1 is substituted by an iterated standard chromatic subdivision. In this sense, Theorem 2 is more convenient for showing impossibility results. In particular, the impossibility of set agreement becomes almost straightforward.

In this note, we first briefly sketch how these two criteria are obtained in, respectively, [7] and [3]. Then we revisit the derivation of Theorem 1 from Theorem 2 conjectured in [3] and we point out the main steps of its proof.

2 Herlihy and Shavit's criterion

Theorem 1 is proved in [7] using the following arguments.

- $\Rightarrow$ Assume a task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ has a wait-free solution using read-write memory. Let $\mathcal{P}(\mathcal{I})$ be the corresponding *protocol complex* constituted by states of the system resulting from running all possible execution of the protocol, and let δ be the corresponding *decision map* that applies to the final local states of $\mathcal{P}(\mathcal{I})$.

  Applying quite involved arguments, we can shown that $\mathcal{P}(\mathcal{I})$ has a *span*, that is, there is a chromatic subdivision $\sigma(\mathcal{I})$ and a color-preserving map ϕ from $\sigma(\mathcal{I})$ to $\mathcal{P}(\mathcal{I})$ such that, for every simplex $S \in \sigma(\mathcal{I})$, $\phi(S) \in \mathcal{P}(carrier(S, \mathcal{I}))$. Clearly, such a span (σ, ϕ) and the composition of ϕ and the decision map δ of $\mathcal{P}$ implies the result. The existence of the span is established by the fact that the protocol complex $\mathcal{P}(\mathcal{I})$ is sufficiently *connected*.

- $\Leftarrow$ Assume now that there is a subdivision σ of $\mathcal{I}$ and a map μ satisfying the conditions of Theorem 1. To show that the task is wait-free solvable, it is sufficient to solve the *chromatic simplex agreement* task on the subdivided complex $\sigma(\mathcal{I})$: the task has $\mathcal{I}$ as an input complex and $\sigma(\mathcal{I})$ as an output complex. Let S be an input simplex of the task. Every process starts with a vertex of S of its color and must decide on a vertex of $\sigma(S)$ of its color, such that all decided vertexes constitute a simplex of $\sigma(S)$.

  The simplex agreement protocol of [7] works as follows. It first solves the simplex agreement on the *iterated standard chromatic subdivision* $\chi^K(\mathcal{I})$ (using the iterated participated set protocol [2]). Using a variation of *approximate agreement* or ε-*agreement*, it is shown that, for sufficiently large K, there is a simplicial color-preserving map ϕ: $\chi^K(\mathcal{I}) \to \sigma(\mathcal{I})$.

  As a result, task T is solved as follows. Processes solve chromatic simplex agreement on $\chi^K(\mathcal{I})$ and then apply map ϕ to their resulting views. Every process ends up with a vertex of $\sigma(\mathcal{I})$ so that all vertexes constitute a simplex of $\sigma(\mathcal{I})$. Then processes apply μ and obtain a simplex of $\mathcal{O}$ satisfying the task specification.

3 Borowsky and Gafni's criterion

In [3], a different approach has been taken. A new computation model, called the *iterated immediate snapshot model* is introduced. A nice property of this model is that its k-round full information protocol complex has the structure of a recursive standard chromatic subdivision $\chi^K(\mathcal{I})$ [7]. That is, a task is solvable *in the iterated immediate snapshot model* if and only if, for some sufficiently large K, there exists a color-preserving map μ that maps every simplex S of $\chi^K(\mathcal{I})$ to a simplex of *carrier*$(S, \mathcal{I})$.

The (one-shot) variant of the model was first introduced in [2] where it is shown to be implementable in the read-write memory model. Thus, any task solvable in the iterated immediate snapshot model is also solvable in the read-write memory model. In fact, the *converse* is also true: any read-write memory protocol that employs a bounded number of reads and writes can be simulated in the iterated immediate snapshot model. Since any read-write memory protocol that solves a task can be shown (applying König's lemma) to employ only a bounded number of reads and writes, we have the result.

Now the fact that the read-write memory model and the iterated immediate snapshot model have the same power of task solvability leads to Theorem 2.[1]

4 Derivation of Theorem 1 from Theorem 2

In [3], an alternative way to derive Theorem 1 is proposed. In fact, having the equivalence between the two models presented above, the necessity part of Theorem 1 is now straightforward: we can just take χ^K as σ. The proof of the sufficiency part is based on the following (purely geometrical) result.

Theorem 3 *Let σ be a chromatic subdivision of a simplex S^n. For some $K \in \mathbb{N}$, there exists a color and carrier preserving map ϕ from $\chi^K(S^n)$ to $\sigma(S^n)$.*

Theorem 3 and the fact that any decision task solvable in the iterated immediate snapshot model is solvable in the read-write model immediately imply the sufficiency part of Theorem 1.

Now we briefly sketch the proof of Theorem 3. Since any decision task solvable in the read-write memory model is solvable in the iterated immediate snapshot model, it is sufficient to show that the chromatic simplex agreement on a subdivided simplex S^n is solvable in the read-write memory model.

We first recall the following topological result (a corollary to the simplicial approximation theorem is recalled:

Lemma 4 *Let σ be a chromatic subdivision of a simplex S^n. For some $K \in \mathbb{N}$, there exists a carrier preserving map ϕ from the iterated barycentric subdivision $BSD^K(S^n)$ to $\sigma(S^n)$.*

Note that, since BSD^K is not a chromatic subdivision, μ is just carrier preserving now. We can define a straightforward carrier preserving map from the standard chromatic subdivision $\chi(S^n)$ to $BSD(S^n)$. The previous result can hence be refined as follows:

Lemma 5 *Let σ be a chromatic subdivision of a simplex S^n. There exists K and a carrier preserving map ϕ from the iterated standard chromatic subdivision $\chi^K(S^n)$ to $\sigma(S^n)$.*

The lemma above implies that the *non-chromatic simplex agreement over a subdivided simplex S^n* (called n-NCSA in [3]) is wait-free solvable in the read-write memory model. For sufficiently large K, there exists a carrier preserving map ϕ from $\chi^K(S^n)$ to $\sigma(S^n)$. Now the aim is to show that a *color-preserving* map indeed exists.

First, we make the following observation. Let C be an m-connected complex. Consider any set U of $k \leq m+1$ vertexes of C. Since C is m-connected, we can associate U with a subcomplex of C that is homeomorphic to some subdivided simplex S^k. We can thus solve k-NCSA on the subdivided simplex, say in K steps. Let μ be the corresponding carrier preserving map from $\chi^K(S^k)$ to the subdivision of S^k. In fact, we can assure that this map μ agrees with the corresponding map defined for any subset of the vertexes in U. Since there is only a finite number of such sets of vertexes in C, there is an upper bound on all such K's. We conclude that there exists $\hat{K} \in \mathbb{N}$ and a map $\hat{\mu}$, such that a non-chromatic simplex agreement can be solved for any subset of at most $m+1$ vertexes C in $\hat{K}$ rounds applying a decision map $\hat{\mu}$.

Now we are ready to present an algorithm that solves the *chromatic* simplex agreement over a subdivided simplex $\sigma(S^n)$.

The algorithm proceeds in rounds. We first give a description of the first round. Consider a barycentric subdivision of $\sigma(S^n)$, $BSD(\sigma(S^n))$. Every process runs a NCSA protocol on $BSD(\sigma(S^n))$ and decides on a vertex of $BSD(\sigma(S^n))$. Every such vertex is a barycenter of a simplex s_i of $\sigma(S^n)$. Moreover:

(1) all such simplexes s_i have a non-empty intersection;

(2) the union of these simplexes is a simplex of $\sigma(S^n)$.

[1]The central result of [3] is namely the emulation of any read-write memory protocol in the iterated immediate snapshot model.

Then every process p_i writes s_i in its designated register and takes a memory snapshot. Let $\{s_j\}_{j\in S}$ be the result of the snapshot. If a vertex of p_i's color is $\cap_{j\in S} s_j$, then p_i decides on this vertex and terminates. Otherwise p_i computes a "core", denoted by $core_i$, as a $\cup_{j\in S} s_j$ minus a vertex of p_i's color (if any) found in $\cup_{j\in S} s_j$. The core represents the vertexes on which processes other than p_i *could have decided* in this round. Since $\cap_{j\in S} s_j$ is non-empty, at least one process decides in the round.

Now consider round k of the algorithm. From round $k-1$, every (not yet decided) process carries its $core_i$. First, p_i chooses a vertex of its color in the *link* of the core in $\sigma(S^n)$, writes the its choice into the memory together with $core_i$ and takes a memory snapshot. Since $\sigma(S^n)$ is n-connected, the link of any intersection of the cores is "sufficiently" connected. The process then runs the NCSA algorithm on the barycentric subdivision of the link, writes the decided simplex in its register, takes a snapshot, etc. Note that since we restrict ourselves to the link of the intersection of the cores, no two processes can decide on vertexes which do not belong to the same simplex. Further, if a process decides on a vertex in round k, this vertex will stay in all cores of all subsequent rounds. Finally, since at least one process decides every round, the algorithm terminates in at most $n+1$ rounds.

References

[1] E. Borowsky and E. Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, pages 91–100, May 1993.

[2] E. Borowsky and E. Gafni. Immediate snapshots and fast renaming. In *Proceedings of the 12th Annual ACM Symposium on Principles of Distributed Computing (PODC'93)*, pages 41–52, August 1993.

[3] E. Borowsky and E. Gafni. A simple algorithmically reasoned characterization of wait-free computation. In *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing (PODC97)*, August 1997.

[4] E. Gafni and E. Koutsoupias. Three-processor tasks are undecidable. *SIAM Journal on Computing*, 28(3):970–983, 1999.

[5] M. Herlihy and N. Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, May 1993.

[6] M. Herlihy and N. Shavit. A simple constructive computability theorem for wait-free computation. In *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC)*, July 1994.

[7] M. Herlihy and N. Shavit. The topological structure of asynchronous computability. *Journal of the ACM*, 46(6):858–923, November 1999.

[8] M. Saks and F. Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, pages 101–110, May 1993.

An Axiomatic Approach to Computing the Connectivity of Synchronous and Asynchronous Systems

Maurice Herlihy[*] Sergio Rajsbaum[†] Mark Tuttle[‡]

Abstract

We present a unified, axiomatic approach to proving lower bounds for the k-set agreement problem in both synchronous and asynchronous message-passing models. The proof involves constructing the set of reachable states, proving that these states are highly connected, and then appealing to a well-known topological result that high connectivity implies that set agreement is impossible. We construct the set of reachable states in an iterative fashion using a round operator that we define, and our proof of connectivity is an inductive proof based on this iterative construction and simple properties of the round operator.

1 Introduction

The consensus problem [18] has received a great deal of attention. In this problem, $n+1$ processors begin with input values, and all must agree on one of these values as their output value. Fischer, Lynch, and Paterson [7] surprised the world by showing that solving consensus is impossible in an asynchronous system if one processor is allowed to fail. This leads one to wonder if there is any way to weaken consensus to obtain a problem that can be solved in the presence of $k-1$ failures but not in the presence of k failures. Chaudhuri [5] defined the k-set agreement problem and conjectured that this was one such problem, and a trio of papers [4, 13, 19] proved that she was right. The k-set agreement problem is a generalization of consensus, where we relax the requirement that processors agree on a single value: the set of output values chosen by the processors may contain as many as k distinct values, and not just 1.

Set agreement (and in particular consensus) has been studied in both synchronous and asynchronous models of computation, but mostly independently.

[*]Brown University, Computer Science Department, Providence, RI 02912; mph@cs.brown.edu.

[†]Instituto de Matemáticas, Universidad Nacional Autónoma de México, Ciudad Universitaria, D.F. 04510, Mexico; rajsbaum@math.unam.mx

[‡]HP Labs, One Cambridge Center, Cambridge, MA 02142; mark.tuttle@hp.com.

Indeed, prior proofs for these models appeared to have little in common, as reflected by the organization of a main textbook in the area [14], where the first part is devoted to synchronous systems and the second part of the book to asynchronous systems. Recent work has been uncovering more and more features and structure in common to both models e.g. [8, 12, 15, 16]. However, these results are in the form of transformations between models, or on proofs that have a similar structure in both models. Only [15] describes an abstract model that encompasses both models, with clearly identified properties that are needed to carry out consensus impossibility results. To go from consensus to set agreement a big step in complexity is encountered, since one must deal with higher dimensional topology instead of just graphs, as discovered by the trio of papers [4, 13, 19] mentioned above. The contribution of this paper is to present a new axiomatic approach were set consensus impossibility proofs can be derived in a uniform manner for both synchronous and asynchronous models.

All known proofs for the set agreement lower bound depend — either explicitly or implicitly — on a deep connection between computation and topology. These proofs essentially consider the simplicial complex representing all possible reachable states of a set agreement protocol, and then argue about the connectivity of this complex. These lower bounds for set agreement follow from the observation that set agreement cannot be solved if the complex of reachable states is sufficiently highly-connected. This connection between connectivity and set agreement has been established both in a generic way [11] and in ways specialized to particular models of computation [1, 4, 6, 10, 11, 12, 19]. Once the connection has been established, however, the problem reduces to reasoning about the connectivity of a protocol's reachable complex.

The primary contribution of this work is a new, substantially simpler proof of how the connectivity of the synchronous and asynchronous complexes evolve over time. Our proof depends on two key insights:

1. The notion of a *round operator* that maps a global state to the set of global states reachable from this state by one round of computation, an operator satisfying a few simple algebraic properties.

2. The notion of an *absorbing poset* organizing the set of global states into a partial order, from which the connectivity proof follows easily using the round operator's algebraic properties.

We believe this new approach has several novel and elegant features. First, we are able to isolate a small set of elementary combinatorial properties of the round operator that suffice to establish the connection with classical topology in a model-independent way. Second, these properties require only local reasoning about how the computation evolves from one round to the next. Finally, most connectivity arguments can be difficult to follow because they mix semantic, combinatorial, and topological arguments. Those arguments are cleanly separated here. The round operator definition relies on semantics: it is a combinatorial restatement of the properties of the synchronous model. Once the round operator is defined, however, we need no further appeals to properties of the original model. We reason in a purely combinatorial way about intersections of global states, and how they can be placed in a partial order. Once these combinatorial arguments are in place, we appeal directly to well-known theorems of topology to establish connectivity. These topology theorems are treated as "black boxes," in the sense that we apply them directly without any need to make additional topological arguments. Furthermore, our absorbing posets are very similar to shellable complexes e.g. [3] so we have uncovered yet one more link between the work of topologists and distributed computing.

For lack of space most of the proofs have been omitted, but appear in the full paper.

2 Preliminaries

2.1 Models

We consider two (standard) message-passing models, the *synchronous* and *asynchronous models*. In both models, we restrict our attention to computations with a round structure: the initial state of each processor is its input value, and computation proceeds in a sequence of rounds. In each round, each processor sends messages to other processors, receives messages sent to it by the other processors in that round, performs some internal computation, and changes state. We assume that processors are following a full-information protocol, which means that each processor sends its entire local state to every processor in every round. This is a standard assumption to make when proving lower bounds. A processor can fail by crashing in the middle of a round, in which case it sends its state only to a subset of the processors in that round. Once a processor crashes, it never sends another message after that.

In the synchronous model [2, 14], all processors execute round r at the same time, and processor P fails to receive a message from processor Q, then Q must have crashed, either in that round or in the previous round.

In the asynchronous model, there is no bound on processor step time nor on message delivery time, so a crashed processor cannot be distinguished from a slow processor. Our results, however, depend only on the unbounded message delivery time. Since our goal is to prove impossibility results, we are free to restrict our attention to executions in which processors take steps at a regular pace, and only message delivery times are delayed. In the behaviors we consider, messages from one processor to another are delivered in FIFO order, but when one message from P to Q is delivered, all outstanding messages from P to Q are delivered at the same time.

It is convenient to recast the asynchronous model in the following *omissions-failure* form. There are at most f potentially faulty processors. At each round, the nonfaulty processors broadcast their states to all processors (including the faulty processors). Each faulty processor broadcasts its state to some subset of the processors, and may omit to send to the others. Processors never crash. It can be shown that k-set agreement lower bounds in this omissions failure model

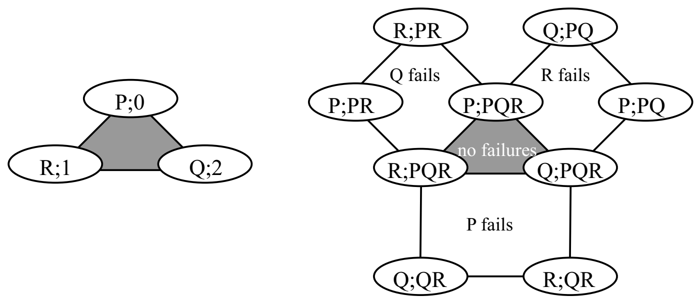


Figure 1: A global state S and the set $\mathcal{S}_1(S)$ of global states after one round from S.

carry over to the standard asynchronous crash-failure model; see [9] for a similar argument.

2.2 Combinatorial Topology

We represent the local state of a processor with a vertex labeled with that processor's id and its local state. We represent a global state as a set of labeled vertexes, labeled with distinct processors, representing the local state of each processor in that global state. In topology, a *simplex* is a set of vertexes, and a *complex* is a set of simplexes that is closed under containment. The *dimension* of a simplex is equal to its number of vertexes minus one. Applications of topology to distributed computing often assume that these vertexes are points in space and that the simplex is the convex hull of these points in order to be able to use standard topology results. As you read this paper, you might find it helpful to think of simplexes in this way, but in the purely combinatorial work done in this paper, a simplex is just a set of vertexes.

As an example, consider the simplex and complex illustrated in Figure 1. On the left side, we see a simplex representing an initial global state in which processors P, Q, and R start with input values 0, 2, and 1. Each vertex is labeled with a processor's id and its local state (which is just its input value in this case). On the right we see a complex representing the set of states that arise after one round of computation from this initial state if one processor is allowed to crash. The labeling of the vertexes is represented schematically by a processor id such as P and a string of processor ids such as PQ. The string PQ is intended to represent the fact that P heard from processors P and Q during the round but not from R, since R failed that round. (We are omitting input values on the right for notational simplicity.) The simplexes that represent states after one round are the 2-dimensional triangle in the center and the 1-dimensional edges that radiate from the triangle (including the edges of the triangle itself). The central triangle represents the state after a round in which no processor fails. Each edge represents a state after one processor failed. For example, the edge with vertexes labeled $P; PQR$ and $Q; PQ$ represent the global state after a round in which R fails by sending a message to P and not sending to Q: P heard from all three processors, but Q did not hear from R.

What we do in this paper is define round operators like the round operator $\mathcal{S}_1$ that maps the simplex S on the left of Figure 1 to the complex $\mathcal{S}_1(S)$ on the right, and then reason about the connectivity of $\mathcal{S}_1(S)$. Informally, connectivity in dimension 0 is just ordinary graph connectivity, and connectivity in higher dimensions means that there are no "holes" of that dimension in the complex. When we reason about connectivity, we often talk about the connectivity of a simplex S when we really mean the connectivity of the induced complex consisting of S and all of its faces. For example, both of the complexes in Figure 1 are 0-connected since they are connected in the graph theoretic sense. In fact, the complex on the left is also 1-connected, but the complex on the right is not since there are "holes" formed by the three cycles of 1-dimensional edges.

Given a simplex S, a *labeling* of S from a set V is a new simplex constructed by replacing each vertex s of S with a pair (s, v), where $v \in V$.

Given a simplex S and a set V, we define the *pseudosphere* $\mathcal{P}(S, V)$ to be this set of labelings of S with elements of V. (We call $\mathcal{P}(S, V)$ a pseudosphere because is has some of the topological properties of a sphere.) The face S is called the *base simplex* of the pseudosphere, and given a simplex T of a pseudosphere $\mathcal{P}(S, V)$, we define *base*(T) to be the base simplex S of the pseudosphere.

The input complex for k-set agreement is $\mathcal{P}(S, V)$, the pseudosphere in which each vertex is labeled with an input from a set V, where $|V| > k$. The set of all reachable states of a protocol P with initial states $\mathcal{P}(S, V)$ is the protocol complex $\mathcal{C} = \mathcal{C}(\mathcal{P}(S, V))$. The fundamental connection between k-set agreement and connectivity is expressed in the following theorem (e.g. [11]):

Theorem 1: Let P be a protocol, and let $\mathcal{C}$ be its protocol complex. If $\mathcal{C}$ is $(k-1)$-connected, then P cannot solve k-set agreement.

Thus, our main task will be to prove that $\mathcal{C}$ is $(k-1)$-connected. Proving that a union of complexes is connected is made easier by the following theorem[1]. Notice that if A and B are complexes then both $A \cup B$ and $A \cap B$ are complexes.

Theorem 2 (Mayer-Vietoris): Let A, B be two complexes. Then $A \cup B$ is c-connected if A and B are c-connected and $A \cap B$ is $(c-1)$-connected.

Think about the special one-dimensional case of this statement: a graph that is the union of subgraphs A, B is 0-connected (connected in the graph theoretic sense) if A and B are 0-connected and $A \cap B$ is -1-connected (nonempty).

To prove that a complex $\mathcal{C}$ is c-connected, we split $\mathcal{C}$ into subcomplexes with less and less simplexes, and apply repeatedly the Mayer-Vietoris theorem. At the bottom of this recursion, we get complexes with just one simplex, and use the following fact.

[1] Actually this theorem is a well-known corollary of the Mayer-Vietoris sequence, which is described in most algebraic topology textbooks; see for example [20] Chapter 4, Section 6.

Theorem 3: A simplex of dimension at least ℓ is $(\ell - 1)$-connected.

In this paper all we need to assume from topology is the previous two theorems. Both are very basic algebraic topology facts that appear in standard textbooks such as [17, 20].

3 Absorbing Posets and Round Operators

The *codimension* of two simplexes S_0 and S_1 is a measure of how much they have in common defined by

$$codim(S_0, S_1) = \max_i \{\dim(S_i) - \dim(\cap_j S_j)\}$$

where $\dim(\emptyset) = -1$ is the dimension of the empty simplex. Two useful properties of this definition are that if $S \subseteq T$ then

$$codim(S, T) = \dim(T) - \dim(S),$$

and if $S \subseteq X \subseteq T$ then

$$codim(S, T) = codim(S, X) + codim(X, T).$$

Let $\mathcal{S}$ be a nonempty set of simplexes, and $\preceq$ a partial order on $\mathcal{S}$.

Definition 4: We say that $(\mathcal{S}, \preceq)$ is an *absorbing poset* if for every two simplexes S and T in $\mathcal{S}$ with $T \not\preceq S$ there is a T_S in $\mathcal{S}$, $T_S \preceq T$ such that

$$S \cap T \subseteq T_S \cap T \tag{1}$$

$$codim(T_S, T) = 1 \tag{2}$$

$$codim(S \cap T, T_S) \leq codim(S, T). \tag{3}$$

The first two properties say that when considering pairwise intersections of simplexes — as we will frequently do in our Mayer-Vietoris arguments — pairs of high codimension are "absorbed" by pairs of low codimension, and we can restrict our attention to pairs of simplexes of codimension one. The third property just says that T_S satisfies the same property that S and T do, namely, $codim(S \cap T, X) \leq codim(S, T)$ for $X = S, T$. An absorbing poset is almost equivalent to a *shellable complex* [3]. In a shellable complex, Equations 2 and 3 apply only to principal faces ("facets") of the complex, while our construction allows one complex in $\mathcal{S}$ to be a proper face of another. It follows that every absorbing poset induces a shellable complex, but not vice-versa.

Lemma 5: If $\mathcal{A}$ is a set of simplexes such that every pair of simplexes has codimension 1, then $(\mathcal{A}, <)$ is an absorbing poset, where $<$ is any total order on $\mathcal{A}$.

Proof: For any simplexes S and T in $\mathcal{A}$ such that $S < T$, pick $T_S = S$. Substituting S for T_S, it is easy to check that the three conditions of Definition 4 are satisfied:

$$S \cap T \subseteq S \cap T$$
$$codim(S, T) = 1$$
$$codim(S \cap T, S) \leq codim(S, T).$$

3.1 Axioms

A *simplicial operator* $\mathcal{Q}$ is a family of maps. Each map $\mathcal{Q}_\ell$ carries a simplex of dimension $m \geq \ell$ to a nonempty set of simplexes, where each simplex has dimension at most m. The subscript ℓ is the operator's *degree*. For $\ell < 0$, it is convenient to define $\mathcal{Q}_\ell(S)$ to be the empty set. Note that $\mathcal{Q}_\ell(\emptyset) = \emptyset$ for all ℓ, and $\mathcal{Q}_0(S) \neq \emptyset$ for any nonempty simplex S.

Simplicial operators extend naturally to sets of simplexes. If $\mathcal{A}$ is a set of simplexes,

$$\mathcal{Q}_\ell(\mathcal{A}) = \bigcup_{A \in \mathcal{A}} \mathcal{Q}_\ell(A). \tag{4}$$

The exact meaning of the operator will vary from model to model. In the synchronous message-passing model, ℓ is the number of processors that can crash in each round. In the asynchronous model, ℓ is the number of processors that remain partially silent in each round.

We use $\mathcal{Q}_k\mathcal{Q}_\ell(S)$ to denote the composition of $\mathcal{Q}_k$ and $\mathcal{Q}_\ell$ applied to S, $\mathcal{Q}_\ell^r(S)$ to denote the r-fold composition of $\mathcal{Q}_\ell$ applied to S, and $\|\mathcal{Q}_\ell^r(S)\|$ to denote the simplicial complex induced by the set $\mathcal{Q}_\ell^r(S)$ (i.e., closed under containment).

The first axiom says that the states reachable after the failure of ℓ processors are reachable after the failure of even more processors.

Axiom 1:

$$\mathcal{Q}_\ell(S) \subseteq \mathcal{Q}_m(S)$$

when $\ell \leq m$.

The next axiom describes multi-round executions. We introduce a model-specific, integer-valued linear function ϕ. Informally, $\phi(f)$ is the number of failures needed in a round to hide the existence of f faulty processors. We will see that in the synchronous model, faulty processors crash, so $\phi(f) = 0$. In the asynchronous model, faulty processors fail to send messages, so $\phi(f) = f$.

Axiom 2: Let $k \geq \ell$. For all $r > 0$, if $c = codim(S_0, S_1)$,

$$\|\mathcal{Q}_k^r\mathcal{Q}_\ell(S_0)\| \cap \|\mathcal{Q}_k^r\mathcal{Q}_\ell(S_1)\| = \|\mathcal{Q}_{k-\phi(c)}^r\mathcal{Q}_{\ell-c}(S_0 \cap S_1)\|.$$

The right-hand-side of this equation is the set of states for processors that cannot tell whether the initial state was S_0 or S_1. The processors that can tell the difference must be silenced in the first round, requiring an extra c failures, and must be kept silent for the remaining rounds, requiring $\phi(c)$ extra failures in each subsequent round.

Axiom 3: For every simplex S, $\mathcal{Q}_\ell(S)$ is an absorbing poset.

4 Theorems and Lemmas

Lemma 6: Let $i \geq j$. For all $r > 0$, if $S \subseteq T$,

$$\|\mathcal{Q}_i^r \mathcal{Q}_j(S)\| \subseteq \|\mathcal{Q}_{i+\phi(c)}^r \mathcal{Q}_{j+c}(T)\|$$

where $c = \mathit{codim}(S, T)$.

Proof: By Axiom 2,

$$\|\mathcal{Q}_k^r \mathcal{Q}_\ell(S_0)\| \cap \|\mathcal{Q}_k^r \mathcal{Q}_\ell(S_1)\| = \|\mathcal{Q}_{k-\phi(c)}^r \mathcal{Q}_{\ell-c}(S_0 \cap S_1)\|$$

where $c = \mathit{codim}(S_0, S_1)$, implying that

$$\|\mathcal{Q}_{k-\phi(c)}^r \mathcal{Q}_{\ell-c}(S_0 \cap S_1)\| \subseteq \|\mathcal{Q}_k^r \mathcal{Q}_\ell(S_0)\|$$

The claim follows by setting $S = S_0 \cap S_1$, $T = S_0$, $i = k - \phi(c)$, and $j = \ell - c$.

Lemma 7: If $(\mathcal{S}, \preceq)$ is an absorbing poset, and S, T, and T_S are defined as in Definition 4, then

$$\mathit{codim}(S \cap T, T_S \cap T) < \mathit{codim}(S, T).$$

Proof: Because $S \cap T \subseteq T_S \cap T$, $\mathit{codim}(S \cap T, T_S \cap T)$ is just the number of vertexes in $T_S \cap T$ but not in $S \cap T$.

There are two cases to consider. First, suppose there is a vertex in T but not in T_S. It follows that

$$\mathit{codim}(S \cap T, T_S \cap T) < \mathit{codim}(S \cap T, T) \leq \mathit{codim}(S, T).$$

Second, suppose instead that $T \subset T_S$. Because T and T_S are distinct, there is vertex in T_S but not in T. It follows that

$$\mathit{codim}(S \cap T, T) < \mathit{codim}(S \cap T, T_S).$$

By Equation 3,

$$\mathit{codim}(S \cap T, T_S) \leq \mathit{codim}(S, T).$$

Combining these inequalities yields the bound.

The next lemma states that every state reachable with a certain number of failures is also reachable with more failures.

Lemma 8:
$$\mathcal{Q}_j^r\mathcal{Q}_k(S) \subseteq \mathcal{Q}_\ell^r\mathcal{Q}_m(S).$$
when $j \leq \ell$ and $k \leq m$.

Proof: We argue by induction on $r \geq 0$. When $r = 0$, the claim follows from Axiom 1.

Suppose $r > 0$. Since $\mathcal{Q}_j^{r-1}\mathcal{Q}_k(S) \subseteq \mathcal{Q}_\ell^{r-1}\mathcal{Q}_m(S)$ by the induction hypothesis, we have

$$\mathcal{Q}_j^r\mathcal{Q}_k(S) = \mathcal{Q}_j\mathcal{Q}_j^{r-1}\mathcal{Q}_k(S) \subseteq \mathcal{Q}_j\mathcal{Q}_\ell^{r-1}\mathcal{Q}_m(S) \subseteq \mathcal{Q}_\ell\mathcal{Q}_\ell^{r-1}\mathcal{Q}_m(S) = \mathcal{Q}_\ell^r\mathcal{Q}_m(S).$$

□

Lemma 9: Let $(\mathcal{S}, \preceq)$ be an absorbing poset, and let $T \in \mathcal{S}$ be a maximal simplex with respect to $\preceq$. We claim that the following sets are both absorbing posets: $(\mathcal{L}, \preceq)$, where $\mathcal{L} = \{L | L \in \mathcal{S} - \{T\}\}$, and $(\mathcal{M}, \preceq)$, where $\mathcal{M} = \{T\}$.

Lemma 10: Let $(\mathcal{A}, \preceq)$ be an absorbing poset containing more than one simplex, and let $A \in \mathcal{A}$ be a maximal simplex with regards to $\preceq$. For each $B \neq A$ in $\mathcal{A}$, there exists a $A_B \in \mathcal{A}$ satisfying the three conditions of Definition 4. We claim that the set
$$\mathcal{B} = \{A_B \cap A | B \in \mathcal{A} - \{A\}\}$$
is an absorbing poset for any total order $<$ on the elements of $\mathcal{A} - \{A\}$

Lemma 11: If every simplex in $\mathcal{Q}_k^r\mathcal{Q}_\ell(\mathcal{A})$ has dimension at least d, then so does every simplex in $\mathcal{A}$.

Lemma 12: Let $\mathcal{Q}_k^r\mathcal{Q}_\ell$ be a composition of simplicial operators where $k \geq \ell$. If $(\mathcal{S}, \preceq)$ is an absorbing poset then for every two simplexes S and T in $\mathcal{S}$ with $T \not\preceq S$ there is a T_S in $\mathcal{S}$ with $T_S \preceq T$, such that

$$\|\mathcal{Q}_k^r\mathcal{Q}_\ell(S)\| \cap \|\mathcal{Q}_k^r\mathcal{Q}_\ell(T)\| \subseteq \|\mathcal{Q}_k^r\mathcal{Q}_\ell(T_S)\| \cap \|\mathcal{Q}_k^r\mathcal{Q}_\ell(T)\|$$
$$\mathit{codim}(T_S, T) = 1$$
$$\mathit{codim}(S \cap T, T_S) \leq \mathit{codim}(S, T).$$

Lemma 13: If $(\mathcal{A}, \preceq)$ is an absorbing poset where ℓ is the minimum dimension of any simplex in $\mathcal{A}$, then $\|\mathcal{A}\|$ is $(\ell - 1)$-connected.

Theorem 14: Let $\mathcal{Q}_k^r\mathcal{Q}_\ell$ be a composition of simplicial operators where $k \geq \ell$, and $(\mathcal{A}, \preceq)$ an absorbing poset. If every simplex in $\mathcal{Q}_k^r\mathcal{Q}_\ell(\mathcal{A})$ has dimension at least ℓ, then $\|\mathcal{Q}_k^r\mathcal{Q}_\ell(\mathcal{A})\|$ is $(\ell - 1)$-connected.

5 The Synchronous Model

We assume a standard synchronous message-passing model with crash failures [2, 14]. The system has $n+1$ processors, and at most f of them can crash in any given execution. Each processor begins in an initial state consisting of its input value, and computation proceeds in a sequence of rounds. In each round, each processor sends messages to other processors, receives messages sent to it by the other processors in that round, performs some internal computation, and changes state. We assume that processors are following a full-information protocol, which means that each processor sends its entire local state to every processor in every round. This is a standard assumption to make when proving lower bounds. A processor can fail by crashing in the middle of a round, in which case it sends its state only to a subset of the processors in that round. Once a processor crashes, it never sends another message after that

A simplex X is *between* two simplexes T and R if $T \subseteq X \subseteq R$. We use $[T : R]$ to denote the set of simplexes between T and R.

Definition 15: Given simplexes S, T, and R, the *pseudosphere* $\mathcal{P}(S, [T : R])$ is the set of all possible labelings of S with simplexes between T and R.

We call this set a pseudosphere because the induced complex has some of the topological properties of a sphere. The simplex S is called the *base simplex* of the pseudosphere, and given a simplex X of a pseudosphere $\mathcal{P}(S, [T : R])$, we define *base*(X) to be S.

Given a simplex S and a set D of processors, let $F = S/D$ be the face of S obtained from S by deleting the vertexes labeled with processors in D. The set of states reachable from S by one round of synchronous computation in which the processors in D fail can be represented by the pseudosphere $\mathcal{P}(F, [F : S])$, the set of all possible labelings of F with simplexes between F and S.

Next, we define the failure operator. Given a simplex S and an integer $\ell \geq 0$, the *ℓ-failure operator* $\mathcal{F}_\ell(S)$ maps S to the set of all faces F of S with *codim*$(F, S) \leq \ell$, which is the set of all faces obtained by deleting at most ℓ vertexes from S. This models the sets of at most ℓ processors that can fail in one round of computation from S.

Definition 16: For every integer $\ell \geq 0$, the *synchronous round operator* $\mathcal{S}_\ell(S)$ is defined by

$$\mathcal{S}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(F, [F : S]).$$

We now check that the synchronous round operator satisfies our axioms.

Lemma 17: $\mathcal{S}_\ell$ satisfies Axiom 1:

$$\mathcal{S}_\ell(S) \subseteq \mathcal{S}_m(S)$$

when $\ell \leq m$.

Proof: Since $\ell \leq m$ implies $\mathcal{F}_\ell(S) \subseteq \mathcal{F}_m(S)$, it follows that

$$\mathcal{S}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(F, [F:S]) \subseteq \bigcup_{F \in \mathcal{F}_m(S)} \mathcal{P}(F, [F:S]) = \mathcal{S}_m(S).$$

In this model, the integer-valued linear function ϕ is simply $\phi(f) = 0$.

Lemma 18: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{S}_k^r \mathcal{S}_\ell(S_0)\| \cap \|\mathcal{S}_k^r \mathcal{S}_\ell(S_1)\| \subseteq \|\mathcal{S}_k^r \mathcal{S}_{\ell-c}(S_0 \cap S_1)\|.$$

Lemma 19: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{S}_k^r \mathcal{S}_{\ell-c}(S_0 \cap S_1)\| \subseteq \|\mathcal{S}_k^r \mathcal{S}_\ell(S_0)\| \cap \|\mathcal{S}_k^r \mathcal{S}_\ell(S_1)\|$$

Corollary 20: $\mathcal{S}_\ell$ satisfies Axiom 2: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{S}_k^r \mathcal{S}_\ell(S_0)\| \cap \|\mathcal{S}_k^r \mathcal{S}_\ell(S_1)\| = \|\mathcal{S}_k^r \mathcal{S}_{\ell-c}(S_0 \cap S_1)\|.$$

To show that $\mathcal{S}_\ell$ satisfies Axiom 3, we impose a partial order on simplexes of $\mathcal{S}_\ell(S)$. Recall that

$$\mathcal{S}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(F, [F:S]).$$

This expression suggests a lexicographic order. We will combine a total order on simplexes F in $\mathcal{F}_\ell(S)$ with a partial order on simplexes of each $\mathcal{P}(F, [F:S])$.

We assume a total order $\leq_{\text{id}}$ on processor ids, which induces a total order on the vertexes of a simplex. We begin by imposing a lexicographic total order on the faces F of S. First we order the faces by decreasing dimension, so that large faces occur before small faces. Then we order faces of the same dimension with a rather arbitrary rule based on our total order on processor ids: we order F before G if the smallest processor id labeling vertexes in F and not G comes before the smallest processor id labeling G and not F. Formally:

Definition 21: Define the total order $<_f$ on the faces of a simplex S by $F <_f G$ if

1. $\dim(F) > \dim(G)$ or
2. $\dim(F) = \dim(G)$ and $p_F <_{\text{id}} p_G$ where

$$p_F = \min\{\mathit{ids}(F) - \mathit{ids}(G)\} \quad \text{and} \quad p_G = \min\{\mathit{ids}(G) - \mathit{ids}(F)\}.$$

Define $F \leq_f G$ if $F <_f G$ or $F = G$.

Next we order the simplexes in a pseudosphere $\mathcal{P}(F, [F:S])$ using the following face ordering: we order A before B if, for each vertex v of the base simplex F, the face of S labeling v in A comes before the face of S labeling v in B. Formally:

Definition 22: Define the partial order $\preceq_p$ on the simplexes of a pseudosphere $\mathcal{P}(F, [F : S])$ by $A \preceq_p B$ if and only if $A_v \leq_f B_v$ for each vertex v in F, where A_v and B_v are the simplexes labeling the vertex v in A and B.

Now we order $\mathcal{S}_\ell(S)$ lexicographically using the face and pseudosphere orders: we order the simplexes in a pseudosphere $\mathcal{P}(F, [F : S])$ before the simplexes in a pseudosphere $\mathcal{P}(G, [G : S])$ if F is ordered before G in the face ordering, and we order the simplexes within a single pseudosphere using the pseudosphere ordering. Formally:

Definition 23: Define the partial order $\preceq_r$ on the simplexes in $\mathcal{S}_\ell(S)$ by $A \preceq_r B$ if and only if

1. *different pseudospheres: base*$(A) <_f$ *base*(B) or
2. *same pseudosphere: base*$(A) =$ *base*(B) and $A \preceq_p B$

Theorem 24: $\mathcal{S}_\ell$ satisfies Axiom 3: For every simplex S, $(\mathcal{S}_\ell(S), \preceq_r)$ is an absorbing poset.

Theorem 25: Assume $n + 1 \geq f + k + 1$. No synchronous protocol for k-set agreement halts in fewer than $\lfloor f/k \rfloor + 1$ rounds in the presence of f crash failures.

Proof: Suppose there is a protocol that halts in fewer than $\lfloor f/k \rfloor + 1$ rounds, and assume without loss of generality that it halts in exactly $r = \lfloor f/k \rfloor$ rounds in every execution. Consider the subset of executions in which at most k processors halt in every round. For the input complex $\mathcal{P}(S, V)$, the set of final states of such executions is $\mathcal{S}_k^r(\mathcal{P}(S, V))$. Every simplex in this complex has dimension at least k. By Theorem 14, this complex is $(k-1)$-connected, and by Theorem 1, the protocol cannot solve k-set agreement.

6 Asynchronous Model

Informally, the asynchronous round operator $\mathcal{A}_\ell(S)$ is defined as follows. There are at most ℓ faulty processors in each round, although the set of faulty processors can change from round to round. Faulty processors never crash, but they can omit sending messages. In each round, all nonfaulty processors send their states to all the processors (including faulty ones), while the faulty processors send messages to an arbitrary subset of processors (perhaps none).

Definition 26: For every integer $\ell \geq 0$, the *asynchronous round operator* $\mathcal{A}_\ell(S)$ is defined by

$$\mathcal{A}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(S, [F : S]).$$

At each asynchronous round every processor is labeled with states that include all nonfaulty processors and some subset of faulty processors. Compare with Definition 16. Notice that every simplex in $\mathcal{A}_\ell(S)$ has the same dimension as S.

We now check that the asynchronous round operator satisfies our axioms.

Lemma 27: $\mathcal{A}_\ell$ satisfies Axiom 1:

$$\mathcal{A}_\ell(S) \subseteq \mathcal{A}_m(S)$$

when $\ell \leq m$.

Proof: Since $\ell \leq m$ implies $\mathcal{F}_\ell(S) \subseteq \mathcal{F}_m(S)$, it follows that

$$\mathcal{A}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(S, [F : S]) \subseteq \bigcup_{F \in \mathcal{F}_m(S)} \mathcal{P}(S, [F : S]) = \mathcal{A}_m(S).$$

In this model, the integer-valued linear function ϕ is simply $\phi(f) = f$.

Lemma 28: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{A}_k^r \mathcal{A}_\ell(S_0)\| \cap \|\mathcal{A}_k^r \mathcal{A}_\ell(S_1)\| \subseteq \|\mathcal{A}_{k-c}^r \mathcal{A}_{\ell-c}(S_0 \cap S_1)\|.$$

Lemma 29: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{A}_{k-c}^r \mathcal{A}_{\ell-c}(S_0 \cap S_1)\| \subseteq \|\mathcal{A}_k^r \mathcal{A}_\ell(S_0)\| \cap \|\mathcal{A}_k^r \mathcal{A}_\ell(S_1)\|.$$

Corollary 30: $\mathcal{A}_\ell$ satisfies Axiom 2: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{A}_k^r \mathcal{A}_\ell(S_0)\| \cap \|\mathcal{A}_k^r \mathcal{A}_\ell(S_1)\| = \|\mathcal{A}_{k-c}^r \mathcal{A}_{\ell-c}(S_0 \cap S_1)\|.$$

To show that $\mathcal{A}_\ell$ satisfies Axiom 3, we impose a partial order on simplexes of $\mathcal{A}_\ell(S)$. Recall that

$$\mathcal{A}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(S, [F : S]).$$

If $F' \subseteq F$, then $\mathcal{P}(S, [F' : S]) \subseteq \mathcal{P}(S, [F : S])$, so we can restrict our attention to faces of codimension ℓ. Unlike in the synchronous model, where simplexes have varying dimensions, all simplexes in this set are labelings of S, and all have dimension n.

We use the same total order $\leq_{\mathrm{id}}$ on processor ids, the same total order $<_f$ on the faces of a simplex S. Next we order the simplexes in $\mathcal{A}_\ell(S)$ using this face ordering: we order A before B if, for each vertex v of the base simplex S the face of F labeling v in A comes before the face of F labeling v in B. Formally:

Definition 31: Define the partial order $\preceq_p$ on the simplexes of $\mathcal{A}_\ell(S)$ by $A \preceq_p B$ if and only if $A_v \leq_f B_v$ for each vertex v in S, where A_v and B_v are the simplexes labeling the vertex v in A and B.

Theorem 32: $\mathcal{A}_\ell$ satisfies Axiom 3: For every simplex S, $(\mathcal{A}_\ell(S), \preceq_p)$ is an absorbing poset.

Theorem 33: No asynchronous protocol for k-set agreement exists in the presence of k crash failures.

References

[1] Hagit Attiya and Sergio Rajsbaum. The combinatorial structure of wait-free solvable tasks. In *Proceedings of the 10th International Workshop on Distributed Algorithms*, volume 1151 of *Lecture Notes in Computer Science*, pages 322–343. Springer-Verlag, October 1996.

[2] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. McGraw–Hill, 1998.

[3] Anders Björner and Michelle L. Wachs. Shellable nonpure complexes and posets, i. *Transactions of the American Mathematical Society*, 348(4):1299–1327, 1996.

[4] E. Borowsky and E. Gafni. Generalized flp impossibility result for t-resilient asynchronous computations. In *Proceedings of the 1993 ACM Symposium on Theory of Computing*, May 1993.

[5] S. Chaudhuri. Agreement is harder than consensus: Set consensus problems in totally asynchronous systems. In *Proceedings Of The Ninth Annual ACM Symosium On Principles of Distributed Computing*, pages 311–234, August 1990.

[6] Soma Chaudhuri, Maurice Herlihy, Nancy A. Lynch, and Mark R. Tuttle. Tight bounds for k-set agreement. *Journal of the ACM (JACM)*, 47(5):912–943, 2000.

[7] M. Fischer, N.A. Lynch, and M.S. Paterson. Impossibility of distributed commit with one faulty process. *Journal of the ACM*, 32(2), April 1985.

[8] Eli Gafni. Round-by-round fault detectors (extended abstract): unifying synchrony and asynchrony. In *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 143–152. ACM Press, 1998.

[9] Maurice Herlihy and Lucia D. Penso. Tight bounds for k-set agreement with limited scope failure detectors. In Faith Fich, editor, *Distributed Computing, 17th International Conference, DISC 2003, Sorrento, Italy, October 1-3, 2003 Proceedings*, volume 2848 of *Lecture Notes in Computer Science*, pages 279–291. Springer, 2003.

[10] Maurice Herlihy and Sergio Rajsbaum. Set consensus using arbitrary objects. In *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing*, pages 324–333, August 1994.

[11] Maurice Herlihy and Sergio Rajsbaum. Algebraic spans. *Mathematical Structures in Computer Science*, 10(4):549–573, August 2000. Special Issue: Geometry and Concurrency.

[12] Maurice Herlihy, Sergio Rajsbaum, and Mark R. Tuttle. Unifying synchronous and asynchronous message-passing models. In *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 133–142. ACM Press, 1998.

[13] Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *Journal of the ACM (JACM)*, 46(6):858–923, 1999.

[14] Nancy Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[15] Yoram Moses and Sergio Rajsbaum. A layered analysis of consensus. *SIAM Journal of Computing (SICOMP)*, 31(4):989–1021, 2002.

[16] Achour Mostéfaoui, Sergio Rajsbaum, and Michel Raynal. The synchronous condition-based consensus hierarchy. In Rachid Guerraoui, editor, *Distributed Computing, 18th International Conference, DISC 2004, Amsterdam, Netherlands, October 4-8, 2004 Proceedings*, Lecture Notes in Computer Science. Springer, 2004.

[17] J.R. Munkres. *Elements Of Algebraic Topology*. Addison Wesley, Reading MA, 1984. ISBN 0-201-04586-9.

[18] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.

[19] Michael Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. *SIAM Journal on Computing*, 29(5):1449–1483, 2000.

[20] E.H. Spanier. *Algebraic Topology*. Springer-Verlag, New York, 1966.

Consensus Continue?
Stability of Multi-Valued Continuous Consensus!

(Extended Abstract)

Lior Davidovitch* Shlomi Dolev† Sergio Rajsbaum‡

June 23, 2004

Abstract

Multi-valued consensus functions deŢned from a vector of inputs (and possibly the previous output) to a single output are investigated. The consensus functions are designed to tolerate t faulty inputs. Two classes of multi-valued consensus functions are deŢned, the *exact value* and the *range value*, which require the output to be one of the non-faulty inputs or in the range of the non-faulty inputs, respectively. The *instability* of consensus functions is examined, counting the maximal number of output changes along a geodesic path of input changes, a path in which each input is changed at most once. Lower and upper bounds for the instability of multi-valued consensus functions are presented. A new technique for obtaining such lower bounds, using edgewise simplex subdivision is presented. Keywords: consensus, stability.

*Department of Computer Science, Ben-Gurion University, Beer-Sheva, 84105, Israel. Email: liord@cs.bgu.ac.il.

†Contact person, phone: +972-8-6472715, fax: +972-8-6477650, Department of Computer Science, Ben-Gurion University, Beer-Sheva, 84105, Israel. Partially supported by IBM faculty award, NSF grant 0098305, the Israeli ministry of defense, the Israeli ministry of Trade and Industry, and the Rita Altura trust chair in computer sciences.

Email: dolev@cs.bgu.ac.il.

‡Instituto de Matemáticas, UNAM, Ciudad Universitaria, D.F. 04510, México. Email: rajsbaum@math.unam.mx.

1 Introduction

The interest in sensor networks and the way they may control the behavior of a system, being a vehicle, airplane, satellite, or other devices is rapidly growing. The *agreement* functions used to ensure smooth and stable control while reţecting the changes in the environment are of great interest.

An abstraction of many agreement functions is the *consensus* problem, where a set of n processors get input values from some set V and must agree on a value. There is always a non-triviality *validity* requirement that speciŢes restrictions on the decided value as a function of the input values and the failure pattern of the execution. This is a fundamental problem in distributed computing that has been widely studied for more than two decades due to its theoretical and practical interest (e.g., [11, 1, 5]). Research on consensus concentrated on the above, *one-shot* setting where processors start with their input values, and have to solve consensus once. Real distributed systems often need to solve consensus repeatedly, on inputs received one after the other. Thus, researchers have also investigated *continuous* versions of consensus where processors have to adapt their consensus decisions continuously (e.g. [6, 10]).

A typical situation where continuous consensus problems arise is systems that read values from replicated sensors [13]. A fault-tolerant consensus algorithm is needed to decide on a single reading because sensors usually do not give the exact same reading of a physical parameter, or because some sensors can fail. Although in the simplest (and most often considered in theory) version of consensus the validity requirement is that a decided value must have been the reading of at least one sensor, in many real settings it is desired that the decided value is a value that has been produced by a majority of the sensors. These and other non-trivial validity requirements are possible, but they all imply that as the readings of the sensors change because the physical parameters that are sampled change, the consensus value will have to change: in the extreme case, all sensors can change their readings from one single value to another, forcing the consensus decision to change accordingly.

Although processors sometimes have to change their outputs during the repeated executions of a consensus algorithm, we prefer continuous consensus algorithms that are *stable*, i.e., in which the number of times the decision value is changed is as small as possible. Usually averaging functions are used in an independent way from sample to sample, sometimes combined with agreement protocols (e.g. [12]), and hence, there is no attempt to maximize stability. There are several reasons for preferring a stable consensus system (more are described in [6]). Some sensors are discrete and are used to control actuators, which may also be discrete. There is the possible operational ampliȚcation of decision changes, say turning an engine on and off. The energy or other resources consumed are sometimes proportional to the number of transitions; e.g. turning an engine on and off takes energy, time, and reduces its lifetime; some related work in VLSI is [3]. Our results may be useful to study problems (e.g. [9]) about the number of ințuencing variables in boolean functions.

In [6] we initiated a study of the stability of continuous consensus systems for the binary case of $|V| = 2$. We deȚned an abstract formalization of a continuous consensus system and the stability measures. The formalization is not tied to any speciȚc model of computation, in order to understand the basic stability issues. We considered *memoryless* systems where consecutive one-shot consensus executions are independent, versus the stability of systems that can keep memory of previous executions. We also studied the stability of *symmetric* systems where decisions are taken solely on the basis of the distribution of the different input values, but not on what speciȚc sensor or processor produced a particular input value. We characterized the stability of systems according to their memory and symmetry properties, proving tight upper and lower bounds for the various cases.

Results: In this paper we extend the results of [6] to the case of multivalued inputs and outputs, $|V| \geq 2$. It turns out that this generalization provides a rich set of problems, some much more interesting than those of the binary case, where we used topological techniques and higher dimensional complexes.

Let t be the number of sensors that may crash-fail. The validity requirement of [6] is that the decision value is an input of some processor, and that if less than $t+1$ inputs are equal to a value b, then the consensus value must be $1-b$ (to make sure the decision is the value of a correct sensor). For the case of a multi-valued consensus system, we consider two extensions of this requirement:

*Exact value system (*EV*):* requires that the output will be the value of a correct input.

*Range value system (*RV*):* requires that the output will be in the range of the correct inputs.

First we show that EV implies that $n \geq |V|t + 1$. For RV we prove that it is sufȚcient to have $n \geq 2t + 1$. The instability of a consensus system with memory is analyzed proving that it is n in the cases where n is the smallest possible value. The investigation of the rest of the cases for EV systems results in range of instability values as a function of $n, |V|$ and t.

Lower bounds for the case of memoryless system are obtained for symmetric functions. The lower bounds are achieved using a technique to subdivide a simplex from [7] and Sperner's Lemma. This can be seen as a generalization of Lemma 2 in [8]: from having a change in the decision values in one dimension (ordering the input values in a line from one extreme to the other where two consecutive input vectors differ in exactly one input) to the case of several dimensions where the border between the different extreme values is a simplex.

We also present an upper bound for a memoryless symmetric system, which is about a factor of 2 away from our lower bound. An interesting open question is to close this gap. Details can be found in [4].

2 Symmetric Memoryless Systems

In this section we detail our results for the case in which memory is not used, in other words the last decision values are not a part of the input to the consensus function. Moreother, the systems that we will refer to are symmetric, meaning that the decision function is oblivious to the order of the input values, i.e., for every $\vec{x}$ and $\vec{y}$, where $\vec{y}$ is a permutation of $\vec{x}$, $f(\vec{x}) = f(\vec{y})$.

The idea is to use Sperner's Lemma, and in a sense, to generalize the case of [8]. We denote the set of possible input vectors by A, namely each such vector has v (non negative integer) components and the sum of the components is exactly n. We put the input vectors in a space of dimension v. The input vectors in which one component is n and all the rest are zero form a simplex, in fact $(v-1)-$simplex. The rest of the vertices are all convex combination of the above vertices, and therefore reside within the simplex. A *subdivision* of a simplex is a partition of the simplex into simplices, such that the sum of the volumes of the dividing simplices equals the volume of the original simplex, and any two dividing simplices do not intersect. We can use Sperner's Lemma to conclude that there is a dividing simplex such that its v vertices have v distinct decision values.

There are several ways to deȚne such a subdivision, fortunately we found (in [7]) a partition that serves us well, in Țnding a geodesic path of length $vt+1$ in every dividing simplex. In particular, we have such a path in the dividing simplex that has different function value for each vertex in its convex.

We will now prove that for any EV system D with a symmetric function f, $instability(\mathrm{D}) \geq vt+1$. To do so, we will use the *edgewise subdivision of a simplex*, deȚned in [7].

Let $\mathcal{S}$ be a $d-$simplex, spanned by $\vec{V}_0, \vec{V}_1, \ldots, \vec{V}_d$. An edgewise subdivision is a function that, given an integer k, transforms every point $\vec{X} \in \mathcal{S}$ into a color scheme M, which is deȚned by a matrix as follows:

$$M = \begin{pmatrix} \chi_{1,0} & \chi_{1,1} & \cdots & \chi_{1,j} \\ \chi_{2,0} & \chi_{2,1} & \cdots & \chi_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{k,0} & \chi_{k,1} & \cdots & \chi_{k,j} \end{pmatrix}$$

where $j \leq d$. Each entry of the matrix is an integer from 0 through d, the columns are pairwise different, and the entries appear in non-decreasing order when read like English text:

$$\chi_{1,0} \leq \chi_{1,1} \leq \ldots \leq \chi_{2,0} \leq \ldots \leq \chi_{k,j}$$

The color scheme deȚnes $j+1$ independent vector $\vec{V}_0^*, \vec{V}_1^*, \ldots, \vec{V}_j^*$, where $\vec{V}_l^* = \frac{1}{k}\sum_{i=1}^{k} \vec{V}_{\chi_{i,l}}$ which span a j-simplex. By applying the function to every point $\vec{X} \in \mathcal{S}$ we obtain a subdivision of $\mathcal{S}$ into subsimplices, some of them are $d-$simplices.

Lemma 2.1 *Let $\mathcal{S}$ be a $(v-1)-$simplex, spanned by $\vec{V}_0, \vec{V}_1, \ldots, \vec{V}_{v-1}$, where $\vec{V}_i = n \cdot \vec{\delta}_{i+1}$, and:*

$$\vec{\delta}_m[k] = \begin{cases} 1 & , m = k \\ 0 & , m \neq k \end{cases}$$

Let $\vec{a}$ be a vertex of any $(v-1)-$simplex in the edgewise subdivision of $\mathcal{S}$ using $k=n$. Then for every $1 \leq i \leq v$, $\vec{a}[i]$ is an integer.

Proof: Let $\vec{a}$ correspond to a column j in some color scheme M. Then,

$$\vec{a}[i] = \frac{1}{n}\sum_{l=1}^{n} \vec{V}_{\chi_{l,j}}[i] = \frac{1}{n}\sum_{l=1}^{n} n \cdot \vec{\delta}_{\chi_{l,j}}[i] = \sum_{l=1}^{n} \vec{\delta}_{\chi_{l,j}}[i]$$

Since $\vec{\delta}_l[m]$ is always integer, then $\vec{a}[i]$ is also integer. ■

We will deȚne $A = \{(\alpha_0, \alpha_1, \ldots, \alpha_{v-1}) \mid \alpha_i \in \mathbb{N} \cup \{0\}, \alpha_0 + \alpha_1 + \ldots + \alpha_{v-1} = n\}$ as before. Lemma 2.1 implies that for every vertex $\vec{a}$ of a

$(v-1)-$simplex of the edgewise subdivision of $\mathcal{S}$, $\vec{a} \in A$. We will deŢne $\tilde{f} : A \to V$ as follows:

$$\tilde{f}(\vec{a}) = f(0^{\vec{a}[1]}1^{\vec{a}[2]}\dots(v-1)^{\vec{a}[v]})$$

and we will color every vertex $\vec{a}$ of the subdivision with $\tilde{f}(\vec{a})$. Since it holds that for every $\vec{a} \in A$ such that $\tilde{f}(\vec{a}) = b$, $\#b(\vec{a}) \geq t+1 > 0$, then the coloring is a Sperner coloring, and according to Sperner's lemma there must exist in the subdivision a subsimplex $\mathcal{S}^*$ such that all its vertices' colors are pairwise different.

Lemma 2.2 *Let $\mathcal{S}$ be a $d-$simplex spanned by $\vec{V}_0, \vec{V}_1, \dots, \vec{V}_d$, such that for every $0 \leq i \leq d$, $\vec{V}_i = n \cdot \vec{\delta}_i$. Let $\mathcal{S}^*$ be a $d-$simplex of the edgewise subdivision of $\mathcal{S}$ using the integer n, when $\mathcal{S}^*$ is spanned by $\vec{V}_0^*, \vec{V}_1^*, \dots, \vec{V}_d^*$. Then $\vec{V}_0^* \to \vec{V}_1^* \dots \to \vec{V}_d^* \to \vec{V}_0^*$ is a geodesic path with minimal changes.*

Now we can prove the main theorem:

Theorem 2.3 *For every* EV *system* D *with a symmetric function f, $instability(\mathrm{D}) \geq vt+1$.*

Proof: $\tilde{f}$ is deŢned over A, which subdivides $\mathcal{S}$ using the edgewise subdivision (Lemma 2.1). We will use $\tilde{f}$ to color the vertices in A. According to Sperner's Lemma, there is a $(v-1)-$simplex in the subdivision so that the colors of its vertices are pairwise different. Let $\vec{a}_0, \vec{a}_1, \dots, \vec{a}_v \in A$ be the vertices spanning $\mathcal{S}$.

As shown above, for every i, b, it holds that $\#b(\vec{a}_i) \geq t$, and for every b there exists i_b such that $\#b(\vec{a}_{i_b}) \geq t+1$, where $i_{b_1} \neq i_{b_2}$ for every $b_1 \neq b_2$. Without loss of generality, we will assume $\#0(\vec{a}_0) \geq t+1$. Then $\vec{a}_0$ corresponds to the input vector $0^t1^t\dots(v-1)^t0\vec{z}$.

According to Lemma 2.2, $\vec{a}_0 \to \vec{a}_1 \to \dots \to \vec{a}_v \to \vec{a}_0$ is a geodesic path with minimal changes. Let c_j the color that change between $\vec{a}_j$ and $\vec{a}_{j+1}$ for every $0 \leq j \leq v-1$, and let $c_v = v$. Now, for every step i we start with an input vector corresponding to $\vec{a}_{i \bmod v}$, and will switch an input value $c_{i \bmod v}$ to $(c_{i \bmod v}+1) \bmod v$, thus arriving to an input vector corresponding to $\vec{a}_{(i+1) \bmod v}$. We can repeat these steps $vt+1$ times, for every input value $b \neq c_0$, the original input vector holds $\#b(\vec{a}_0) \geq t$, and also $\#0(\vec{a}_0) \geq t+1$. And since the colors of the vertices are pairwise different, then the values of $\tilde{f}$ over the vertices are pairwise different, and therefore the values of f over the input vectors corresponding to the vertices are pairwise different (between pairs corresponding to different vertices), then this path yields $vt+1$ changes to the consensus value. ■

Acknowledgment: It is a pleasure to thank Ronen Peretz and Nir Shavit for their comments.

References

[1] H. Attiya and J. Welch, *Distributed Computing: Fundamentals, Simulations and Advanced Topics,* McGraw–Hill, 1998.

[2] S. Chaudhuri, "More *choices* allow more *faults*: Set consensus problems in totally asynchronous systems," *Inf. and Comp.,* 105(1):132-158, 1993.

[3] A.P. Chandrakasan and R.W. Brodersen, *Low power digital CMOS design,* Kluwer Academic Publishers, 1995.

[4] L. Davidovitch, S. Dolev and S. Rajsbaum. "Consensus Continue? Stability of Multi-Valued Continuous Consensus!", Technical Report #2004-03, Department of Computer Science, Ben-Gurion University, May 2004.

[5] S. Dolev, *Self-Stabilization*, MIT Press, 2000.

[6] S. Dolev, and S. Rajsbaum, "Stability of Long-lived Consensus", *Journal of Computer and System Sciences*, Vol. 67, Issue 1, pp. 26-45, August 2003.

[7] H. Edelsbrunner and D. R. Grayson, "Edgewise subdivision of a simplex", *Discrete Comput. Geom. 24*, 2000, 707-719

[8] M.J. Fischer, N.A. Lynch and M.S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process," *Journal of the ACM* 32, 1985, pp. 374-382.

[9] J. Kahn, G. Kalai and N. Linial, "The Inţuence of Variables on Boolean Functions," *Proc. of the IEEE FOCS*, 1988.

[10] L. Lamport, "The part-time parliament," *ACM Trans. on Comp. Systems,* vol. 16 (2):133-169, May 1998.

[11] N.A. Lynch, *Distributed Algorithms,* Morgan Kaufmann Publishers, Inc. 1996.

[12] K. Marzullo, "Tolerating failures of continuous-valued sensors," *ACM Trans. on Comp. Systems,* 8(4):284-304, Nov. 1990.

[13] H. Kopetz and P. Veríssimo, "Real Time and Dependability Concepts," chapter 16, pp.411-446 in Sape Mullender (ed.), *Distributed Systems,* ACM Press, 1993.

The Complexity of Early Deciding Set Agreement: How can Topology help?

Rachid Guerraoui and Bastian Pochon
Distributed Programming Laboratory
EPFL, Switzerland

Contact author :	Bastian Pochon
Email :	Bastian.Pochon@EPFL.ch
Phone :	+41-21-6935267
Address :	Laboratoire de Programmation Distribuée Bâtiment INR EPFL CH-1015 Lausanne, Switzerland
Category :	Regular paper

Abstract

The aim of this paper is to pose a challenge to the experts of (algebraic) topology techniques. We present an early deciding algorithm that solves the set agreement problem, i.e., the problem which triggered research on applying topology techniques to distributed computing. We conjecture the algorithm to be optimal, and we discuss the need and challenges of applying topology techniques to prove the lower bound.

The Complexity of Early Deciding Set Agreement: How can Topology help?

Rachid Guerraoui and Bastian Pochon
Distributed Programming Laboratory
EPFL, Switzerland

Abstract

The aim of this paper is to pose a challenge to the experts of (algebraic) topology techniques. We present an early deciding algorithm that solves the set agreement problem, i.e., the problem which triggered research on applying topology techniques to distributed computing. We conjecture the algorithm to be optimal, and we discuss the need and challenges of applying topology techniques to prove the lower bound.

1 Introduction

Results about the set agreement problem are intriguing, in the sense that they present an intrinsic trade-off between the number of processes in a system, the degree of coordination that these processes can reach, and the number of failures that can be tolerated [3]. Set agreement is a generalization of the widely studied consensus problem [4], in which each process is supposed to propose a value, and eventually decide on some value that was initially proposed, such that every correct eventually decides (just like in consensus). In contrast with consensus however, processes may not decide on more than k distinct values. Hence set agreement is also referred to as k-set agreement.

K-set agreement was introduced in [2]. The paper also introduced k-set agreement algorithms in the asynchronous model[1] when less than k processes may crash. In [6], techniques borrowed from algebraic topology were first used to prove the impossibility of k-set agreement in an asynchronous model where k processes may crash. In [3, 5], tight lower bounds were derived for set agreement in the synchronous model prone to process crash. The framework presented in [5] uses the tools from algebraic topology introduced in [6] and allows for proving lower bounds in both the asynchronous and the synchronous models.

Early deciding algorithms are those the efficiency of which depends on the effective number of failures in a given run, rather than on the (total) number of failures that can be tolerated. The effective number of failures is traditionally denoted by f, whereas the total number of failures that are tolerated is denoted by t. In practice, failures rarely happen, and it makes sense to devise algorithms that decide earlier when fewer failures occur. For uniform consensus, Charron-Bost and Schiper [1] have shown that there is a significant improvement on the efficiency when considering the effective number of failures. More precisely, they propose a uniform consensus algorithm in which every process that decides, decides by round $f + 2$ in any run with f failures. This bound is shown to be tight [1, 7].

To the best of our knowledge, no result for set agreement have been presented in the context of early deciding algorithms. In the present paper, we give an early deciding set agreement algorithm.

[1]In the asynchronous model, there is no bound on process relative speed and message communication delay.

We conjecture this algorithm to be optimal, and we discuss the need and challenges of applying topology techniques to prove the lower bounds.

The rest of the paper is organized as follows. Section 2 gives our system model. Section 3 presents our early deciding algorithm. Section 4 discusses the optimality of this result.

2 Model

We consider a set of $N = n + 1$ processes $\Pi = \{p_0, \ldots, p_n\}$. Processes communicate by message-passing. We consider that communication channels are reliable. Processes execute in a synchronous, round-based model [8]. A run is a sequence of rounds. Every round is composed of three phases. In the first phase, every process broadcasts a message to all the other processes. In the second phase, every process receives all the messages sent to it during the round. In the third phase, every process may perform a local computation, before starting the next round. Processes may fail by crashing. A process that crashes does not execute any step, and is said to be *faulty*. Processes that do not crash are said to be *correct*. When process p_i crashes in round r, a subset of the messages that p_i sends in round r (possibly the empty set) is received by the end of round r. A message broadcast in round r by a process that does not crash in round r is received, at the end of round r, by every process that reaches the end of round r. We consider that there are at most $t < N$ processes that may fail in any run.

3 An Algorithm

Figure 1 presents an early deciding k-set agreement algorithm. For $t < N - k$ (or equivalently, $t \leq n - k$), this algorithm achieves the following bounds: (1) for $0 \leq \lfloor f/k \rfloor \leq \lfloor t/k \rfloor - 2$, every process that decides, decides by round $\lfloor f/k \rfloor + 2$, and (2) for $\lfloor f/k \rfloor \geq \lfloor t/k \rfloor - 1$, every process that decides, decides by round $\lfloor f/k \rfloor + 1$. Note that this is a strict generalization of the tight lower bounds on uniform consensus [1, 7].[2]

In the algorithm, every process p_i sends its estimate value est_i in every round. At the end of every round, p_i updates est_i with the minimum estimate value received from any other process. The intuition behind set agreement achieved by the algorithm is as follows. In round r, if p_i observes that $k-1$ processes, or less, crash in that round, then process p_i knows all but at most $k-1$ values among the smallest values remaining in the system. Process p_i can thus safely decide on est_i if p_i reaches the end of the next round.

We give an intuition of why the algorithm is faster when $\lfloor f/k \rfloor = \lfloor t/k \rfloor - 1$. Note that in this case, every process that decides, decides by round $\lfloor f/k \rfloor + 1$. At the end of round $\lfloor t/k \rfloor - 1$, the processes have more than k distinct estimate values only if there remain $2k - 1$ processes or less that are still allowed to crash. In round $\lfloor t/k \rfloor - 1$, every process that detects $k - 1$ or less new crashes may safely decide at the end of round $\lfloor t/k \rfloor$. The reason is the following. First, if $k - 1$ or less processes crash in round $\lfloor t/k \rfloor$, then at most $k - 1$ distinct estimate values remain in the system, and it is safe to decide for any process. In contrast, if more than $k - 1$ processes crash in round $\lfloor t/k \rfloor$, then $k - 1$ or less processes may still crash. Denote by x the number of processes that detect less than $k - 1$ process crashes in round $\lfloor t/k \rfloor$. These x processes decide at the end of round $\lfloor t/k \rfloor$. Assume that they immediately crash after deciding. Thus there are at most $k - 1 - x$ processes that may still crash in the last round $\lfloor t/k \rfloor + 1$. At the end of that round $\lfloor t/k \rfloor + 1$, at

[2] For uniform consensus, the tight lower bound is $f + 2$, for $0 \leq f \leq t - 2$, and $f + 1$, for $f \geq t - 1$ [1].

```
At process p_i:
  halt := ∅ ; decided := deciding := false
  S^r := ∅, 1 ≤ r ≤ ⌊t/k⌋ + 1

  procedure propose(v_i)
     est_i := v_i
     for r from 1 to ⌊t/k⌋ + 1 do
        if decided or deciding then send (r, DEC, est_i) to all
        else send (r, EST, est_i) to all
        if deciding then
           decide(est_i) ; return
        else if decided then
           return
        else if received any (r, DEC, est_j) then
           est_i := est_j ; deciding := true
        else
           S^r := {(est_j, j) | (r, EST, est_j) is received in round r from p_j}
           halt := Π \ ∪_{(est_j, j) ∈ S^r} {j}
           est_i := min{est_j | (est_j, j) ∈ S^r}
           if r = ⌊t/k⌋ and |S^r| ≥ N − k⌊t/k⌋ + 1 then
              decided := true ; decide(est_i)
           else if |halt| < rk then
              deciding := true
     decide(est_i)
     return
```

Figure 1: An early deciding k-set agreement algorithm (code for process p_i)

most $k-x$ values may be decided (if $k-1-x$ processes crash). In total, processes decide at most on $x+(k-x)$ distinct values.

In the following proofs, we denote the local copy of a variable *var* at process p_i by var_i, and the value of var_i at the end of round r by var_i^r. $crashed^r$ denotes the set of processes that crash *before* completing round r, $ests^r$ denotes the set of estimate values of every process at the end of round r. By definition, round 0 ends when the algorithm starts. No process decides by round 0. We first prove three general claims about the algorithm of Figure 1.

Claim 1 $ests^r \subseteq ests^{r-1}$.

Proof: The proof of the claim is straightforward: for any process p_i, $est_i^r \in ests^{r-1}$. □

Claim 2 *If at the end of round $0 \le r \le \lfloor t/k \rfloor$ no process has decided, and at most l processes crash in round $r+1$, then $|ests^{r+1}| \le l+1$.*

Proof: Consider that the conditions of the claim hold and assume by contradiction that $|ests^{r+1}| \ge l+2$. By assumption, there are $l+2$ processes with distinct estimate values at the end of round $r+1$. Denote by $q_0, \dots, q_{l+1}$ these processes, such that $est_{q_i}^{r+1} \le est_{q_{i+1}}^{r+1}$, for $0 \le i \le l+1$. Processes $q_0, \dots, q_l$ do not send $est_{q_0}^{r+1}, \dots, est_{q_l}^{r+1}$ in round $r+1$; otherwise, q_{l+1} receives one of the smallest

$l+1$ estimate values in round $r+1$. Thus there are $l+1$ processes which send values corresponding to $est_{q_0}^{r+1}, \ldots, est_{q_l}^{r+1}$ in round $r+1$ and which crash in round $r+1$; otherwise, q_{l+1} receives one of the smallest $l+1$ estimate value in round $r+1$. This contradicts our assumption that at most l processes crash in round $r+1$. $\square$

Claim 3 *If, at the end of round* $1 \leq r \leq \lfloor t/k \rfloor$*, no process has decided, and* $|ests^r| \geq k+1$*, then* $|crashed^r| \geq rk$*.*

Proof: We prove the claim by induction. For the base case $r=1$, assume that the conditions of the claim hold. That is, at the end of round 1, there exist $k+1$ distinct processes $q_0, \ldots, q_k$ with distinct estimate values. By Claim 2, $|crashed^1| \geq k$. Assume the claim for round $r-1$, and assume the conditions of the claim hold at round r. We prove the claim for round r. By assumption, there are $k+1$ processes $q_0, \ldots, q_k$ at the end of round r with $k+1$ distinct estimates. By Claim 1, $k+1$ processes necessarily reach the end of round $r-1$ with $k+1$ distinct estimates. Thus Claim 3 holds at round $r-1$ (induction hypothesis), and thus, $|crashed^{r-1}| \geq (r-1)k$. By Claim 2, at least k processes crash in round r. Thus $|crashed^r| \geq k + |crashed^{r-1}| \geq rk$. $\square$

The next proposition asserts the correctness of the algorithm.

Proposition 4 *The algorithm in Fig. 1 solves* k*-set agreement.*

Proof: Validity and Termination are obvious. To prove k-ket agreement, we consider the lowest round r in which some process decides. Let p_i be one of the processes that decides in round r. We consider three mutually exclusive cases: (1) p_i decides in round $2 \leq r \leq \lfloor t/k \rfloor - 1$, (2) p_i decides in round $r = \lfloor t/k \rfloor$, and (3) p_i decides in round $r = \lfloor t/k \rfloor + 1$. (In the algorithm, no process decides before round 2.)

Case 1. p_i necessarily decides at line 9, and thus executes line 21 in round $r-1$, where *deciding* is set to *true*. (Because no process decides before p_i, p_i may not receive any DEC message before deciding; and because $r \leq \lfloor t/k \rfloor - 1$, p_i may not decide at line 19.) In round $r-1$, p_i executes line 21 only if p_i evaluates $|crashed^{r-1}| < rk$ at line 20. Thus, from Claim 3, there are at most k distinct estimates at the end of round $r-1$, which ensures agreement.

Case 2. There are two cases to consider: (1) p_i decides at line 9, after executing line 21 at the end of round $r-1$, or (2) p_i decides at line 19. (Because no process decides before p_i, p_i may not receive any DEC message before deciding.) In case (1), p_i executes line 21 in round $r-1$ only if p_i evaluates $|crashed^{r-1}| < rk$ at line 20. Thus, from Claim 3, there are at most k distinct estimates at the end of round $r-1$, which ensures agreement. In case (2), we consider $ests^{r-1}$. If $|ests^{r-1}| \leq k$, agreement is ensured thereafter. Thus consider that $|ests^{r-1}| \geq k+1$. By Claim 3, there exist $k+1$ distinct processes with different estimates at the end of round $r-1$ only if $|crashed^{r-1}| \geq k(r-1) = k(\lfloor t/k \rfloor - 1) \geq t - 2k + 1$, or, equivalently, only if at most $2k-1$ processes may crash in the two subsequent rounds (rounds $\lfloor t/k \rfloor$ and $\lfloor t/k \rfloor + 1$). In round $\lfloor t/k \rfloor$, p_i decides at line 19 only if p_i receives at least $n - k\lfloor t/k \rfloor + 1$ messages. Thus, by Claim 2, the processes that decide at the end of round $\lfloor t/k \rfloor$, including p_i, decide on at most k distinct values. Denote by x the number of processes that effectively crash in round $\lfloor t/k \rfloor$, and by y the number of processes that decide at the end of round $\lfloor t/k \rfloor$. We distinguish two cases: (a) $x \leq k-1$, and (b) $x \geq k$. In case (a), by Claim 2, $k-1$ values or less remain in the system at the end of round $\lfloor t/k \rfloor$; agreement is then ensured. In case (b), at most $2k - 1 - x \leq k-1$ processes may crash among the processes that decide at the end of round $\lfloor t/k \rfloor$ and the processes that take part to round $\lfloor t/k \rfloor + 1$. We claim that the total number of distinct decision values is at most k. Indeed, denote by y_{crash} the

5

number of processes that decide at the end of round $\lfloor t/k \rfloor$ and then immediately crash. In round $\lfloor t/k \rfloor + 1$, at most $k - 1 - y_{crash}$ may crash. By Claim 2 processes that decide at the end of round $\lfloor t/k \rfloor + 1$ may decide on at most $k - y_{crash}$ distinct estimate values. Hence the maximum number of decided values is $(k - y_{crash}) + y_{crash} = k$.

Case 3. By contradiction, consider that, at the end of round $\lfloor t/k \rfloor + 1$, there exist $k+1$ distinct processes $q_0, \ldots, q_k$ with different estimates, and which decide on their estimates. By Claim 1, there exist $k+1$ processes with distinct estimates at the end of round $r-1$. By Claim 3 and because $r = \lfloor t/k \rfloor + 1$, $|crashed^{r-1}| > k(r-1) = k\lfloor t/k \rfloor > t - k$. By Claim 2, there exist k processes that crash in round $\lfloor t/k \rfloor + 1$. Thus $|crashed^{r}| \geq k + |crashed^{r-1}| = k + k\lfloor t/k \rfloor > t$. A contradiction. $\Box$

The next proposition asserts the efficiency of the algorithm.

Proposition 5 *In any run with $0 \leq f \leq t$ failures, any process that decides, decides*

1. *by round $\lfloor f/k \rfloor + 2$, if $0 \leq \lfloor f/k \rfloor \leq \lfloor t/k \rfloor - 2$, and*

2. *by round $\lfloor f/k \rfloor + 1$, if $\lfloor f/k \rfloor \geq \lfloor t/k \rfloor - 1$.*

Proof: We proceed by separating both cases.

Case 1. Assume a run with f failures, such that $\lfloor f/k \rfloor \leq \lfloor t/k \rfloor - 2$. By contradiction, assume that there exists a process p_i for which $|halt_i^r| \geq rk$, for $r = \lfloor f/k \rfloor + 1$. (If $|halt_i^r| < rk$, then p_i decides at line 9 in the next round.) Process p_i does not decide in round r; in particular, p_i does not receive any DEC message in round r. We have $|halt_i^r| \geq rk = (\lfloor f/k \rfloor + 1)k = \lfloor f/k \rfloor k + k > f$. A contradiction.

Case 2. Assume a run with f failures, such that $\lfloor f/k \rfloor \geq \lfloor t/k \rfloor - 1$. First assume that $\lfloor f/k \rfloor = \lfloor t/k \rfloor - 1$, and assume by contradiction that there exists a process p_i that does not decide by round $r = \lfloor f/k \rfloor + 1$. Thus p_i does not receive any DEC message in round r. Assume by contradiction that p_i does not decide at line 19. Thus $|S^r| < N - k\lfloor t/k \rfloor + 1$, and $f > k\lfloor t/k \rfloor - 1$. This implies in turn that $\lfloor f/k \rfloor > \lfloor t/k \rfloor - 1$. A contradiction. When $\lfloor f/k \rfloor = \lfloor t/k \rfloor$, then any process that decides, decides by round $\lfloor f/k \rfloor + 1 = \lfloor t/k \rfloor + 1$. $\Box$

4 Discussion

We conjecture our early deciding set agreement algorithm to be tight. For the case $k = 1$, we fall back on uniform consensus, for which the lower bound of $f + 2$, for $0 \leq f \leq t - 2$, and $f + 1$, for $f \geq t - 1$, is known to be tight [1, 7]. For $k > 1$, we envisage a proof based on notions of algebraic topology, along the lines of [6, 5]. We discuss here why the techniques presented in [6, 5] do not apply, and we propose a possible line of research to address this open question.

The principle behind the proofs in [6, 5] is (1) to associate a so-called *protocol complex* to the set of all executions of the processes of a full-information protocol in a given model, and (2) to observe that such a protocol complex presents a topological obstruction that prevents it to be mapped onto the output complex of k-set agreement.[3] The obstruction that is used is $(k-1)$-connectivity. Indeed, Theorem 6 in [5] relates the $(k-1)$-connectivity of a protocol complex for k-set agreement in any model, with the impossibility of solving k-set agreement in that model.

Connectivity leads to impossibility because we assume that the processes all need to decide at the end of the same round. Indeed, one can applies Sperner's lemma to show that there exists at

[3] The output complex represents the set of all possible final states of the processes, according to the specification of k-set agreement.

least one execution where more than k values are decided, when the protocol complex is $(k-1)$-connected [6, 5]. On the other hand, in the algorithm presented in this paper, the processes may actually decide faster than $\lfloor t/k \rfloor + 1$, the tight lower bound for k-set agreement [3, 5].[4] Why is that possible? Is there any contradiction?

In fact, there is no contradiction. Processes may actually decide faster than the lower bound of $\lfloor t/k \rfloor + 1$, because, in the early deciding case, the processes are not forced to necessarily decide *all* at the end of the *same* round. In other word, even so the protocol complex is still $(k-1)$-connected after, say, round $r < \lfloor t/k \rfloor + 1$, *some* processes may already decide, provided that these processes span a subcomplex within the full protocol complex that is, at least, not $(k-1)$-connected.

The lower bound proof we envisage is (1) to consider, inductively on f and within the protocol complex of a full-information protocol in the synchronous model prone to process crash, the subcomplex spanned by those processes which *see* f failures, or less, after $(\lfloor f/k \rfloor + 1)$ rounds, and (2) to show that these processes still cannot decide at that round. For the latter point, it is not clear whether $(k-1)$-connectivity is a strong enough condition, or if a stronger (more restrictive) property on protocol complexes is required.

References

[1] B. Charron-Bost and A. Schiper. Uniform consensus harder than consensus. Technical Report DSC/2000/028, École Polytechnique Fédérale de Lausanne, Switzerland, May 2000.

[2] S. Chaudhuri. More choices allow more faults: set consensus problems in totally asynchronous systems. *Information and Computation*, 105(1):132–158, July 1993.

[3] S. Chaudhuri, M. Herlihy, N. A. Lynch, and M. R. Tuttle. Tight bounds for k-set agreement. *Journal of the ACM (JACM)*, 47(5):912–943, 2000.

[4] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

[5] M. Herlihy, S. Rajsbaum, and M. Tuttle. Unifying synchronous and asynchronous message-passing models. In *Proceedings of the* 17^{th} *ACM Symposium on Principles of Distributed Computing*, pages 133–142, 1998.

[6] M. Herlihy and N. Shavit. The topological structure of asynchronous computability. *Journal of the ACM (JACM)*, 46(6):858–923, 1999.

[7] I. Keidar and S. Rajsbaum. On the cost of fault-tolerant consensus when there are no faults – a tutorial. Technical report, MIT Technical Report MIT-LCS-TR-821, 2001. (Preliminary version in SIGACT News, Distributed Computing Column, 32(2):45–63, 2001.

[8] N. A. Lynch. *Distributed Algorithms*. Morgan-Kaufmann, 1996.

[4]For example, in the failure-free run, all processes decide by the end of round 2, for any t and any k.

CONTEXT FOR MODELS OF CONCURRENCY

PETER BUBENIK

ABSTRACT. Many categories have been used to model concurrency. Using any of these, the challenge is to reduce a given model to a smaller representation which nevertheless preserves the relevant computer-scientific information. That is, one wants to replace a given model with a simpler model with the same dihomotopy-type. Unfortunately, the obvious definition of dihomotopy equivalence is too coarse. This paper introduces the notion of *context* to refine the notion of dihomotopy equivalence.

1. INTRODUCTION

Various algebraic topological models are being used for studying concurrency. Among them are precubical complexes [Gou95], d-spaces [Gra03, Gra02], local pospaces [FGR99], and FLOW [Gau03]. For a given concurrent system, each of these categories provides a model which captures the relevant computer-scientific properties of the system.

These categories are large in two senses. They are large 'locally' in that a given model contains many paths which correspond to executions which are essentially equivalent. They are also large 'globally' in that a given concurrent system has a large number of models within the category. The size of these categories is a strength in terms of their descriptive power. However, for calculational purposes one would like to reduce these models to a smaller, possibly even discrete, representation.

A major goal of current research in this area is to introduce equivalences to obtain such smaller representations, which nevertheless still retain the relevant computer-scientific properties.

On the local front progress has been made in reducing the path space of a given model using dihomotopy equivalences of paths and the fundamental category [Gra03]. One global approach is to pass to the component category [FRGH04, Rau03]. In this paper we introduce another global approach, which is perhaps more geometric and which is compatible with the model categoric approach of [Bub04].

In the classical (undirected) topological case, the solution to this 'global' problem is well-understood. The equivalent spaces are the homotopy equivalent ones, or perhaps the weak-homotopy equivalent ones. So for example, all of the contractible spaces (those homotopy equivalent to a point) are equivalent.

In the directed case there is a similar notion of dihomotopy equivalence (which will be defined in the next section). However this notion is too coarse.

Date: August 19, 2004.

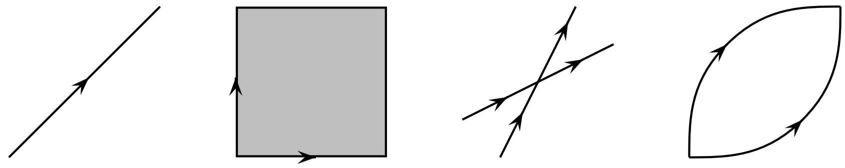


FIGURE 1. $\vec{I}$, $\vec{I}\times\vec{I}$, $\vec{X}$, and $\vec{O}$

Example 1.1. Let $\vec{I}$ be the unit interval $[0,1]$ with a direction given by the usual ordering of the real numbers. Let $\vec{I}\times\vec{I}$ be $[0,1]\times[0,1]$ with the ordering $(x,y)\leq(x',y')$ if and only if $x\leq x'$ and $y\leq y'$. Let $\vec{X}$ be the space in Figure 1 given by attaching two copies of $\vec{I}$ at their centers. Then as will be shown explicitly in Example 2.6, $\vec{I}$, $\vec{I}\times\vec{I}$ and $\vec{X}$ are all dihomotopy equivalent to a point. However $\vec{I}$ models an execution with one initial state and one final state while $\vec{X}$ models an execution with two initial states and two final states.

Clearly a stronger notion of equivalence is needed. Since $\vec{I}$ and $\vec{I}\times\vec{I}$ both have one initial state and one final state and all execution paths seem to be essentially equivalent it seems natural that we should look for a definition of equivalence under which these are equivalent. However even this 'equivalence' has a pitfall.

For a notion of equivalence to be practical it should continue to hold under certain 'pastings'. Roughly speaking, if we make the same addition to equivalent models we should still have equivalent models (This will be made precise in the next section).

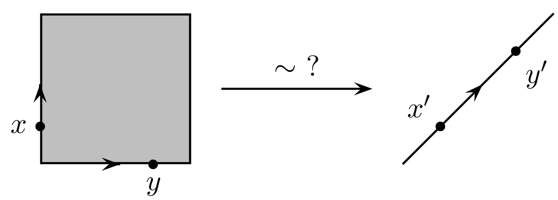


FIGURE 2. A hypothetical equivalence

Example 1.2. Assume we have an equivalence $\vec{I}\times\vec{I}\to\vec{I}$ as in Figure 2. Consider the following pasting on $\vec{I}\times\vec{I}$. Let $\vec{O}$ be the space in Figure 1 constructed by attaching two copies of $\vec{I}$ at their initial points and at the final points.[1] Let $\vec{O}_1$ and $\vec{O}_2$ be two copies of $\vec{O}$. For $i=1,2$ let $a_i,b_i\in\vec{O}_i$ denote the initial and final points of $\vec{O}_i$. Now choose two points $x,y\in\vec{I}\times\vec{I}$ such that neither $x\leq y$ nor $y\leq x$. Let $x',y'\in\vec{I}$ be the images of x and y under the assumed equivalence (Figure 2). Then either $x'\leq y'$ or $y'\leq x'$, since $\vec{I}$ is totally ordered.

If $x'\leq y'$ then identify b_1 and x and identify a_2 and y. Call this space B and denote C the space obtained by collapsing $\vec{I}\times\vec{I}\subset B$ to $\vec{I}$ using the given equivalence (Figure 3). Then there is an execution path from a_1 to b_2 in C but not in B. So the models B and C are not equivalent. A similar

[1]This is M.Grandis' *ordered circle* $\uparrow O^1$ [Gra03, Section 1.2].

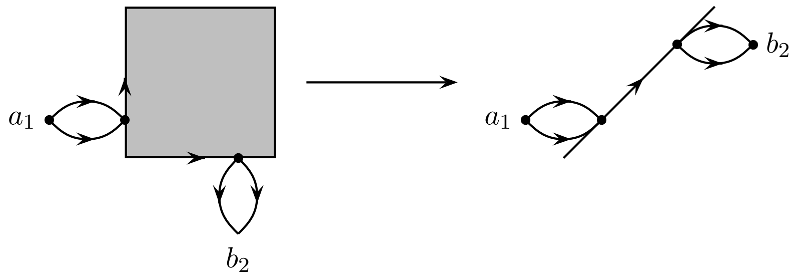


FIGURE 3. A map $B \to C$ which should not be an equivalence

construction is possible if $y' \leq x'$. Thus from this point of view $\vec{I} \times \vec{I}$ and $\vec{I}$ should not be equivalent.

This gives a good indication of the current state of affairs for determining a global notion equivalence. We don't even know whether or not $\vec{I} \times \vec{I}$ and $\vec{I}$ should be equivalent.

In this paper we introduce the idea of *context*. Whether or not $\vec{I}$ and $\vec{I} \times \vec{I}$ are equivalent depends on the context. If we permit pastings as in Example 1.2, then they are not equivalent. However if we only permit pastings to the initial and final points of $\vec{I}$ and $\vec{I} \times \vec{I}$ then they are equivalent. Again, we will make this precise in the next section. From the computer-scientific point of view this can be interpreted as follows. We cannot expect equivalent concurrent systems to still be equivalent after arbitrary (but equal) changes. However, if equal additions are made in a suitably modular way, then the resulting systems should still be equivalent.

2. Context for dihomotopy equivalences

In this section we make precise the intuitive ideas presented in the introduction.

Definition 2.1.

- A *partial order* on a topological space U is a reflexive, transitive, anti-symmetric relation $\leq$. If U has a partial order $\leq$ which is a closed subset of $U \times U$ under the product topology, then call U a *pospace*.
- A *dimap* $f : (U_1, \leq_1) \to (U_2, \leq_2)$ is a continuous map $f : U_1 \to U_2$ such that $x \leq_1 y$ implies that $f(x) \leq_2 f(y)$.
- A product of pospaces $(U_1, \leq_1)$ and $(U_2, \leq_2)$ is a pospace whose underlying topological space is $U_1 \times U_2$ and whose order relations is given by $(x, y) \leq (x', y')$ if and only if $x \leq_1 x'$ and $y \leq_2 y'$.
- A subspace A of a pospace U inherits a pospace structure under the definition $x \leq_A y$ if and only if $x \leq_U y$. This is called a *sub-pospace*.

Definition 2.2. Let **Pospace** be the the category whose object are pospaces and whose morphisms are dimaps.

For the sake of simplicity we will work with pospaces but one should be able to easily extend or adapt the constructions presented here for other models of concurrency.

Let $\vec{I} = ([0,1], \leq)$ where $\leq$ is the usual ordering of $\mathbb{R}$. A *dipath* in a pospace B is a dimap $\vec{I} \to B$. If $\vec{I}_1$ and $\vec{I}_2$ are two copies of $\vec{I}$, then let $\vec{X} = (\vec{I}_1 \amalg \vec{I}_2)/\sim$ where $(\frac{1}{2})_1 \sim (\frac{1}{2})_2$ (see Figure 1).

Definition 2.3.
- Given dimaps $f, g : B \to C \in$ **Pospace**, ϕ is a *dihomotopy*[2] from f to g if $\phi : B \times \vec{I} \to C \in$ **Pospace**, $\phi|_{B\times\{0\}} = f$ and $\phi|_{B\times\{1\}} = g$. In this case write $\phi : f \to g$.
- Write $f \simeq g$ if there is a chain of dihomotopies $f \to f_1 \leftarrow f_2 \to \ldots \leftarrow f_n \to g$. This is an equivalence relation.
- A dimap $f : B \to C$ is a *dihomotopy equivalence* if there is a dimap $g : C \to B$ such that $g \circ f \simeq \mathrm{Id}_B$ and $f \circ g \simeq \mathrm{Id}_C$. In this case write $B \simeq C$.

Our explicit dihomotopies will often be of the following form.

Definition 2.4. Let $f, g : B \to C$ be two dimaps. If such a map exists let the *linear interpolation* between f and g be the map $H : B \times \vec{I} \to C$ given by $H(b,t) = (1-t)f(b) + tg(b)$.

Remark 2.5. Note that there is no guarantee that such a map exists. However one can check that it does for the cases we will consider.

Example 2.6. We will show that under Definition 2.3, $\vec{I}$, $\vec{I} \times \vec{I}$, and $\vec{X}$ are dihomotopy equivalent to a point. Let $f : \vec{I} \to *$, $g : * \to \vec{I}$ be the constant map and the inclusion of the point to $1 \in \vec{I}$. Then $f \circ g = \mathrm{Id}_*$ and it remains to show that $\mathrm{Id}_{\vec{I}} \simeq g \circ f$. Let $H : \vec{I} \times \vec{I} \to \vec{I}$ be the linear interpolation between $\mathrm{Id}_{\vec{I}}$ and $g \circ f$. That is,

$$\begin{aligned} H(x,t) &= (1-t)x + t \\ &= x + t(1-x) \end{aligned}$$

Then H is a dimap and is the desired homotopy $\mathrm{Id}_{\vec{I}} \to g \circ f$.

In exactly the same way one can show that the constant map $f : \vec{I} \times \vec{I} \to *$ is a dihomotopy equivalence with $g : * \to \vec{I} \times \vec{I}$ given by $g(*) = (1,1)$.

To show that the constant map $f : \vec{X} \to *$ is a dihomotopy equivalence with $g(*) = (\frac{1}{2})_1 = (\frac{1}{2})_2$ is slightly more complicated. Again $f \circ g = \mathrm{Id}_*$. To show $\mathrm{Id}_{\vec{X}} \simeq g \circ f$ we will construct a chain of dihomotopies $\mathrm{Id}_{\vec{X}} \xrightarrow{H_1} h \xleftarrow{H_2} g \circ f$. Let $h : \vec{X} \to \vec{X}$ be given by

$$x \mapsto \begin{cases} \frac{1}{2} & \text{if } x < \frac{1}{2} \\ x & \text{otherwise} \end{cases}$$

Let H_1 be the linear interpolation between $\mathrm{Id}_{\vec{X}}$ and h and let H_2 be the linear interpolation between $g \circ f$ and h. Then H_1 and H_2 are dimaps and are the desired dihomotopies.

We will show that in the right *context* it is no longer true that $\vec{I}$, $\vec{I} \times \vec{I}$, and $\vec{X}$ are dihomotopy equivalent to a point.

[2]This is the notion of dihomotopy in [Gra03] which is stronger than the notion of dihomotopy in [FGR99] (which uses I instead of $\vec{I}$).

Definition 2.7. Let the *context* be an object $A \in \mathbf{Pospace}$. Instead of working in the category $\mathbf{Pospace}$ we will work in the category $\mathbf{A} \downarrow \mathbf{Pospace}$ of pospaces under A. The objects of $\mathbf{A} \downarrow \mathbf{Pospace}$ are dimaps $A \xrightarrow{\iota_B} B$ where $B \in \mathrm{Ob}\,\mathbf{Pospace}$. The morphisms in $\mathbf{A} \downarrow \mathbf{Pospace}$ are dimaps

$$\begin{tikzcd} & A \arrow[dl, "\iota_B"'] \arrow[dr, "\iota_C"] & \\ B \arrow[rr, "f"] & & C \end{tikzcd}$$

such that $f \circ \iota_B = \iota_C$.

Example 2.8. For example if $A = S^0 = \{a, b\}$ then $B \in \mathrm{Ob}\,\mathbf{A} \downarrow \mathbf{Pospace}$ is a pospace with two marked points. An important example is $\vec{I}$ with $\iota_{\vec{I}}(a) = 0$ and $\iota_{\vec{I}}(b) = 1$.

Definition 2.9.
- Given dimaps $f, g : B \to C \in \mathbf{A} \downarrow \mathbf{Pospace}$, ϕ is a *dihomotopy* from f to g if $\phi : B \times \vec{I} \to B \in \mathbf{Pospace}$, $\phi|_{B \times \{0\}} = f$, $\phi|_{B \times \{1\}} = g$, and for all $a \in A$, $\phi(\iota_B(a), t) = \iota_C(a)$. In this case write $\phi : f \to g$.
- Write $f \simeq g$ if there is a chain of dihomotopies $f \to f_1 \leftarrow f_2 \to \ldots \leftarrow f_n \to g$. This is an equivalence relation.
- A dimap $f : B \to C$ is a *dihomotopy equivalence* if there is a dimap $g : C \to B$ such that $g \circ f \simeq \mathrm{Id}_B$ and $f \circ g \simeq \mathrm{Id}_C$. In this case write $B \simeq C$.

We can think of this as *dihomotopy rel A*. In case the context A is one point or two points we get pointed and bipointed dihomotopies. However we will see that this notion is useful for more general contexts.

Example 2.10. Let us return to the example above. In the context of its end points $\vec{I}$ is no longer dihomotopic to a point. There is a dimap

$$\begin{tikzcd} & S^0 \arrow[dl, "\iota_{\vec{I}}"'] \arrow[dr, "\iota_*"] & \\ \vec{I} \arrow[rr, "f"] & & * \end{tikzcd}$$

making the diagram commute, but there is no map $g : * \to \vec{I}$ making the diagram commute. The same statement is true for $\vec{I} \times \vec{I}$ and $\vec{X}$.

Example 2.11. In the context of $S^0 = \{a, b\}$ let $\iota_{\vec{I}}(a) = 0$, $\iota_{\vec{I}}(b) = 1$, $\iota_{\vec{I} \times \vec{I}}(a) = (0, 0)$, and $\iota_{\vec{I} \times \vec{I}}(b) = (1, 1)$. We claim that in this context $\vec{I}$ and $\vec{I} \times \vec{I}$ are dihomotopy equivalent. Let $f : \vec{I} \times \vec{I} \to \vec{I}$ and $g : \vec{I} \to \vec{I} \times \vec{I}$ be given by $f(x, y) = \max(x, y)$ and $g(x) = (x, x)$. Then f and g are both dimaps, $f \circ g = \mathrm{Id}_{\vec{I}}$ and $g \circ f(x, y) = (\max(x, y), \max(x, y))$. It remains to construct a dihomotopy rel S^0 from $\mathrm{Id}_{\vec{I} \times \vec{I}}$ to $g \circ f$.

Let ϕ be the linear interpolation (see Definition 2.4) of $\mathrm{Id}_{\vec{I} \times \vec{I}}$ and $g \circ f$. That is,

$$\begin{aligned} \phi(x, y, t) &= (1 - t)(x, y) + t(\max(x, y), \max(x, y)) \\ &= (x + t(\max(x, y) - x), y + t(\max(x, y) - y)). \end{aligned}$$

Then ϕ is the desired dihomotopy rel $\{a, b\}$.

Hence $\vec{I} \times \vec{I}$ and $\vec{I}$ are dihomotopy equivalent in the given context.

We will now introduce some definitions and prove some lemmas that will allow us to relate dihomotopy rel A to the *fundamental category*. Furthermore it will enable us to quickly see that certain spaces are not dihomotopy equivalent in a given context.

Definition 2.12. Let $B \in$ **Pospace** and let $x, y \in B$.

- Recall that a *dipath* is a dimap $\gamma : \vec{I} \to B$.
- Given dipaths $\gamma_1, \gamma_2 : \vec{I} \to B$ such that $\gamma_1(0) = \gamma_2(0) = x$ and $\gamma_1(1) = \gamma_2(2) = y$. Then γ_1 and γ_2 are dihomotopy equivalent if $\gamma_1 \simeq \gamma_2$ in $\mathbf{S^0} \downarrow$ **Pospace** where $\iota_{\vec{I}}(a) = 0$, $\iota_{\vec{I}}(b) = 1$, $\iota_B(a) = x$, and $\iota_B(b) = y$. In this case write $\gamma_1 \simeq \gamma_2$.
- Let $\vec{\pi}_1(B)(x, y)$ be the set of dihomotopy equivalence classes of dimaps from x to y. The *fundamental category* of B is the category $\vec{\pi}_1(B)$ which has the same objects as B but the morphisms between x and y are the elements of $\vec{\pi}_1(B)(x, y)$.[3]

Lemma 2.13. *Given dihomotopic dipaths $\gamma \simeq \gamma' : \vec{I} \to B$ and a dimap $f : B \to C$, then $f \circ \gamma \simeq f \circ \gamma'$ are dihomotopic dipaths.*

Proof. Since $\gamma \simeq \gamma'$ there is a chain of dihomotopies $\gamma \xrightarrow{H_1} \gamma_1 \xleftarrow{H_2} \gamma_2 \xrightarrow{H_3} \dots \xleftarrow{H_n} \gamma_n \xrightarrow{H_{n+1}} \gamma'$. Then $f \circ \gamma \xrightarrow{f \circ H_1} f \circ \gamma_1 \xleftarrow{f \circ H_2} f \circ \gamma_2 \xrightarrow{f \circ H_3} \dots \xleftarrow{f \circ H_n} f \circ \gamma_n \xrightarrow{f \circ H_{n+1}} f \circ \gamma'$ is a chain of dihomotopies from $f \circ \gamma$ to $f \circ \gamma'$. □

Corollary 2.14. *For a dimap $f : B \to C$ and $x, y \in B$ there is an induced map $\vec{\pi}_1(f) : \vec{\pi}_1(B)(x, y) \to \vec{\pi}_1(C)(f(x), f(y))$ mapping $[\gamma] \mapsto [f \circ \gamma]$. That is, a dimap $f : B \to C$ induces a functor $\vec{\pi}_1(f) : \vec{\pi}_1(B) \to \vec{\pi}_1(C)$.*

Lemma 2.15. *Given dihomotopy equivalent dimaps $f \simeq g : B \to C \in$ $\mathbf{A} \downarrow$ **Pospace** and a dipath $\gamma : \vec{I} \to B$ such that $\gamma(0) = \iota_B(a)$ and $\gamma(1) = \iota_B(b)$ where $a, b \in A$ then $f \circ \gamma \simeq g \circ \gamma$ are dihomotopy equivalent dipaths.*

Proof. Since $f \simeq g$ there is a chain of dihomotopies $f \xrightarrow{H_1} f_1 \xleftarrow{H_2} f_2 \xrightarrow{H_3} \dots \xleftarrow{H_n} f_n \xrightarrow{H_{n+1}} g$. For $1 \leq i \leq n+1$, let $H'_i = H_i \circ (\gamma \times \vec{I})$. Then $f \circ \gamma \xrightarrow{H'_1} f_1 \circ \gamma \xleftarrow{H'_2} f_2 \circ \gamma \xrightarrow{H'_3} \dots \xleftarrow{H'_n} f_n \circ \gamma \xrightarrow{H_{n+1}} g \circ \gamma$ is a chain of dihomotopies of dipaths. □

Proposition 2.16. *If $f : B \to C \in \mathbf{A} \downarrow$ **Pospace** is a dihomotopy equivalence then for all $a, b \in A$ the induced map $\vec{\pi}_1(f)(a, b) : \vec{\pi}_1(B)(\iota_B(a), \iota_B(b)) \to \vec{\pi}_1(C)(\iota_C(a), \iota_C(b))$ is an isomorphism.*

Proof. Let $a, b \in A$ and let $\gamma, \gamma' : \vec{I} \to B$ be dipaths such that $\gamma(0) = \gamma'(0) = \iota_B(a)$ and $\gamma(1) = \gamma'(1) = \iota_B(b)$. Assume that $f : B \to C \in$ $\mathbf{A} \downarrow$ **Pospace** is a dihomotopy equivalence. Then there is a dimap $g : C \to B \in \mathbf{A} \downarrow$ **Pospace** such that $g \circ f \simeq \mathrm{Id}_B$ and $f \circ g \simeq \mathrm{Id}_C$.

Assume that $f \circ \gamma \simeq f \circ \gamma'$. Using Lemma 2.13 $\gamma = \mathrm{Id}_B \circ \gamma \simeq g \circ f \circ \gamma \simeq g \circ f \circ \gamma' \simeq \mathrm{Id}_B \circ \gamma' = \gamma'$. Therefore $\vec{\pi}_1(f)(a, b)$ is injective.

[3]This differs from the definition of fundamental category in [FRGH04] where the equivalence classes of dimaps use I and not $\vec{I}$.

Also let $\phi : \vec{I} \to C$ be a dipath in C with $\phi(0) = \iota_C(a)$ and $\phi(1) = \iota_C(b)$. Then $g \circ \phi$ is a dimap in B with $g \circ \phi(0) = \iota_B(a)$ and $g \circ \phi(1) = \iota_B(b)$. By Lemma 2.15 $\phi = \mathrm{Id}_C \circ \phi \simeq f \circ g \circ \phi$. Therefore $\vec{\pi}_1(f)(a,b)$ is surjective. Thus $\vec{\pi}_1(f)(a,b)$ is an isomorphism as claimed. $\square$

Example 2.17. Let $A = S^0 = \{a, b\}$ and choose any points $x, y \in \vec{I} \times \vec{I}$ such that $x \not\leq y$ and $y \not\leq x$. Then $\vec{\pi}_1(\vec{I} \times \vec{I})(x,y)$ and $\vec{\pi}_1(\vec{I} \times \vec{I})(y,x)$ are empty. However for any dimap $f : \vec{I} \times \vec{I} \to \vec{I}$ (see Figure 2), either $f(x) \leq f(y)$ or $f(y) \leq f(x)$ since I is totally ordered. Therefore one of $\vec{\pi}_1(\vec{I})(f(x), f(y))$ and $\vec{\pi}_1(\vec{I})(f(y), f(y))$ is nonempty. So in the context of $\iota_{\vec{I}\times\vec{I}}(a) = x$ and $\iota_{\vec{I}\times\vec{I}}(b) = y$, $\vec{I} \times \vec{I}$ is not dihomotopy equivalent to $\vec{I}$ since there can be no dihomotopy equivalence $f : \vec{I} \times \vec{I} \to \vec{I}$ such that $\vec{\pi}_1(f)(a,b)$ is an isomorphism.

Example 2.18. Let $\vec{X}$ be the space defined earlier (see Figure 1). In the context of its four endpoints $(0)_1$, $(0)_2$, $(1)_1$, and $(1)_2$, $\vec{X}$ is not dihomotopy equivalent to $\vec{I}$ since there are no dipaths from $(0)_1$ to $(0)_2$ and from $(1)_1$ to $(1)_2$.

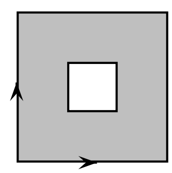


FIGURE 4. $\vec{I} \times \vec{I}$ with a square removed

Example 2.19. In this example we show that in the context of the points $(0,0)$ and $(1,1)$, $\vec{I} \times \vec{I}$ with a square removed from its interior is dihomotopy equivalent to its boundary.

Let $A = S^0 = \{a, b\}$. Let B be the sub-pospace of $\vec{I} \times \vec{I}$ in Figure 4 given by $\{(x,y) \in \vec{I} \times \vec{I} \mid \text{it is not true that } \frac{1}{3} < x < \frac{2}{3}, \frac{1}{3} < y < \frac{2}{3}\}$. Let $\iota_B(a) = (0,0)$ and let $\iota_B(b) = (1,1)$. Let C be the boundary of $\vec{I} \times \vec{I}$ with $\iota_C(a) = (0,0)$ and $\iota_C(b) = (1,1)$.

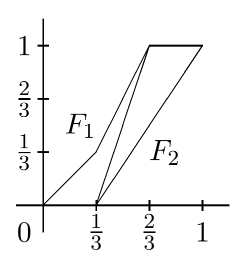


FIGURE 5. The graphs of F_1, F_2, and $F_2 \circ F_1$.

Let $F_1 : [0,1] \to [0,1]$ be given by the mapping

$$x \mapsto \begin{cases} x & \text{if } x < \frac{1}{3} \\ 2x - \frac{1}{3} & \text{if } \frac{1}{3} \leq x \leq \frac{2}{3} \\ 1 & \text{if } x > \frac{2}{3} \end{cases}$$

Let $F_2 : [0,1] \to [0,1]$ be given by the mapping

$$x \mapsto \begin{cases} 0 & \text{if } x < \frac{1}{3} \\ \frac{3}{2}x - \frac{1}{2} & \text{if } \frac{1}{3} \leq x \leq 1 \end{cases}$$

See Figure 5 for graphs of F_1, F_2, and $F_2 \circ F_1$.

Let $f : B \to C$ and $g : C \to B$ be given by $f(x,y) = (F_2 \circ F_1(x), F_2 \circ F_1(y))$ and $g(x,y) = (x,y)$. Also let $h : B \to B$ be given by $h(x,y) = (F_1(x), F_1(x))$. One can check that f, g, and h are dimaps.

We will now give explicit dihomotopies rel A showing that $g \circ f \simeq \mathrm{Id}_B$ rel A and $f \circ g \simeq \mathrm{Id}_C$ rel A. Let

$$H_1(x,y,t) = (1-t)(x,y) + t(F_1(x), F_1(y)).$$

Then $H_1 : \mathrm{Id}_B \xrightarrow{\simeq} h$ is a dihomotopy rel A. Similarly let

$$H_2(x,y,t) = (1-t)(F_2 \circ F_1(x), F_2 \circ F_1(y)) + t(F_1(x), F_1(y)).$$

Then $H_2 : g \circ f \xrightarrow{\simeq} h$ is a dihomotopy rel A. Therefore $g \circ f \simeq \mathrm{Id}_B$ rel A as claimed. Furthermore since C is a sub-pospace of B and $f \circ g = f = g \circ f$, the above dihomotopies restrict to C showing that $f \circ g \simeq \mathrm{Id}_C$ rel A.

We remark that using Definition 2.4, H_1 is a linear interpolation between Id_B and h, and H_2 is a linear interpolation between $g \circ f$ and h.

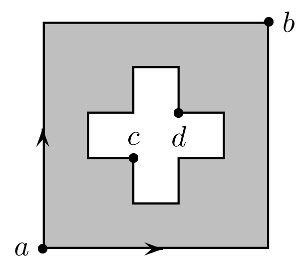


FIGURE 6. The swiss flag with labeled points $\{a, b, c, d\}$

Example 2.20. The swiss flag.

In this example we give an explicit dihomotopy between the famous swiss flag pospace in Figure 6 and the one-dimensional sub-pospace in Figure 7 in the context of four points.

Let A be the discrete pospace $\{a, b, c, d\}$. Let B be the sub-pospace of $\vec{I} \times \vec{I}$ given in Figure 6 with the cross removed and $\iota_B(a) = (0,0)$, $\iota_B(b) = (1,1)$, $\iota_B(c) = (\frac{2}{5}, \frac{2}{5})$, and $\iota_B(d) = (\frac{3}{5}, \frac{3}{5})$. Let C be the subspace of B given in Figure 7 with the same marked points.

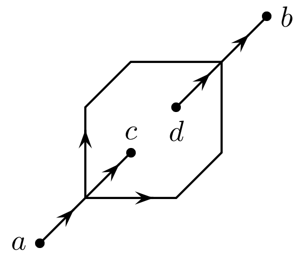


FIGURE 7. A sub-pospace of the swiss flag with the same labeled points $\{a, b, c, d\}$

Let $g : C \to B$ be the dimap given by $g(x, y) = (x, y)$. Let $f : B \to C$ be the dimap given by $f(x, y) = f_4 \circ f_3 \circ f_2 \circ f_1(x, y)$ where f_1, f_2, f_3, and f_4 are defined below. As in the previous example we will give a chain of dihomotopies rel A to show that $\mathrm{Id}_B \simeq g \circ f$. Since C is a subspace of B and $g \circ f = f = f \circ g$ this will restrict to a chain of dihomotopies rel A which show that $\mathrm{Id}_C \simeq f \circ g$. As a result we will have that $B \simeq C$.

$$f_1(x,y) = \begin{cases} (\max(x,y), \max(x,y)) & \text{if } 0 \le x \le \frac{1}{5}, 0 \le y \le \frac{1}{5} \\ (\frac{1}{5}, y) & \text{if } 0 \le x \le \frac{1}{5}, \frac{1}{5} < y \\ (x, \frac{1}{5}) & \text{if } 0 \le y \le \frac{1}{5}, \frac{1}{5} < x \\ (x,y) & \text{otherwise} \end{cases}$$

$$f_2(x,y) = \begin{cases} (\min(x,y), \min(x,y)) & \text{if } \frac{4}{5} \le x \le 1, \frac{4}{5} \le y \le 1 \\ (\frac{4}{5}, y) & \text{if } \frac{4}{5} \le x \le 1, y < \frac{4}{5} \\ (x, \frac{4}{5}) & \text{if } \frac{4}{5} \le y \le 1, x < \frac{4}{5} \\ (x,y) & \text{otherwise} \end{cases}$$

$$f_3(x,y) = \begin{cases} (\max(x, y - \frac{2}{5}), \max(x + \frac{2}{5}, y)) & \text{if } \frac{1}{5} \le x \le \frac{2}{5}, \frac{3}{5} \le y \le \frac{4}{5} \\ (\max(x, y + \frac{2}{5}), \max(x - \frac{2}{5}, y)) & \text{if } \frac{1}{5} \le y \le \frac{2}{5}, \frac{3}{5} \le x \le \frac{4}{5} \\ (\max(x,y), \max(x,y)) & \text{if } \frac{3}{5} \le x \le \frac{4}{5}, \frac{3}{5} \le y \le \frac{4}{5} \\ (\frac{2}{5} + 2(x - \frac{2}{5}), y) & \text{if } \frac{2}{5} \le x \le \frac{3}{5}, y = \frac{4}{5} \\ (x, \frac{2}{5} + 2(y - \frac{2}{5}), y) & \text{if } \frac{2}{5} \le y \le \frac{3}{5}, x = \frac{4}{5} \\ (x,y) & \text{otherwise} \end{cases}$$

$$f_4(x,y) = \begin{cases} (\min(x,y), \min(x,y)) & \text{if } \frac{1}{5} \leq x \leq \frac{2}{5}, \frac{1}{5} \leq y \leq \frac{2}{5} \\ (\frac{3}{5} - 2(\frac{3}{5} - x), y) & \text{if } \frac{2}{5} \leq x \leq \frac{3}{5}, y = \frac{1}{5} \\ (x, \frac{3}{5} - 2(\frac{3}{5} - y)) & \text{if } \frac{2}{5} \leq y \leq \frac{3}{5}, x = \frac{1}{5} \\ (x,y) & \text{otherwise} \end{cases}$$

Let H_1, H_2, H_3, and H_4 be the linear interpolations (see Definition 2.4) between Id_B and f_1, $f_2 \circ f_1$ and f_1, $f_2 \circ f_1$ and $f_3 \circ f_2 \circ f_1$, and f and $f_3 \circ f_2 \circ f_1$. Then these give a chain of dihomotopies

$$\mathrm{Id}_B \xrightarrow{H_1} f_1 \xleftarrow{H_2} f_2 \circ f_1 \xrightarrow{H_3} f_3 \circ f_2 \circ f_1 \xleftarrow{H_4} f = g \circ f.$$

Therefore $\mathrm{Id}_B \simeq g \circ f$. Restricting to C gives a chain of dihomotopies showing $\mathrm{Id}_C \simeq f = f \circ g$. Hence B is dihomotopy equivalent to C rel $\{a, b, c, d\}$.

3. Pushouts of dihomotopy equivalences

In this section we elaborate on the statement made in the introduction that dihomotopy equivalences should be preserved by 'pastings'. In fact we discuss the construction of a homotopy theory for concurrency. In order that we do not lose focus from the main ideas of this paper, we will defer the details of the definitions and constructions of this section to the appendix.

An excellent framework for a homotopy theory on a category is given by a *model structure* on the category [Hov99]. A category with a model structure and all small limits and colimits is called a *model category*. A model structure has three special classes of morphisms: *fibrations*, *cofibrations*, and *weak equivalences* which satisfy certain axioms (see Appendix A for the full definition).

The category **Pospace** has all small limits and colimits. However it is too restrictive to model many concurrent systems (for example pospaces cannot contain loops). Though all of our examples are in **Pospace** a better framework for concurrency is the category **LoPospc** of *local pospaces*. A *local pospace* is a topological space such that each point has a neighborhood which is a pospace and that these local orders are compatible (for a precise definition see Appendix B).

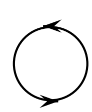


FIGURE 8. The local pospace $\vec{S}^1$

Example 3.1. An example of a local pospaces is the directed circle $\vec{S}^1$ in Figure 8 obtained by identifying the endpoints of $\vec{I}$. While $\vec{S}^1$ does not have a transitive, anti-symmetric order, locally it has the structure of the pospace $\vec{I}$.

Unfortunately, unlike **Pospace**, **LoPospc** does not contain all small colimits. However there is a formal method of enlarging a category to one with all small limits and colimits.[4] Furthermore this larger category has a canonical model structure! [Dug01] For details on how this theory can be applied to **LoPospc** see the appendix and [Bub04]. In the appendix we give a more precise version of the following theorem (Theorem B.4) which is proved in [Bub04].

Theorem 3.2. *Let* $\mathbf{C} = \mathbf{LoPospc}$*. Then* $\mathbf{C}$ *is a subcategory of a model category* $\mathbf{UC}$*. The morphisms in* $\mathbf{C}$ *that are cofibrations are the monomorphisms and the morphisms in* $\mathbf{C}$ *that are weak equivalences are the isomorphisms.*

From the point of view of just $\mathbf{C}$, this model structure is almost trivial. However one can *localize* $\mathbf{UC}$ with respect to a set M of morphisms in $\mathbf{C}$ to obtain a new category $\mathbf{UC/M}$. $\mathbf{UC/M}$ has the same objects and cofibrations as $\mathbf{UC}$ but the morphisms in M are now weak equivalences [Dug01]. The problem is to choose a good set of morphisms M. For example, we can take M to be the set of dihomotopy equivalences in $\mathbf{C}$.

One of the key properties of $\mathbf{UC}$ and $\mathbf{UC/M}$ is that they are *left proper*. That is, the pushout of a weak equivalence over a cofibration is a weak equivalence.

$$\begin{tikzcd}
G \arrow[r, "\sim", "f"'] \arrow[d, tail, "j"'] & C \arrow[d, tail] \\
D \arrow[r, "\sim", "g"'] & E
\end{tikzcd}$$

In particular in $\mathbf{UC/M}$ if $f \in M$ then g is a weak equivalence.

Example 3.3. Recall the dihomotopy equivalence $f : \vec{I} \times \vec{I} \to \vec{I}$ of Example 2.11. Also recall the inclusions of $\vec{I} \times \vec{I}$ and $\vec{I}$ into B and C (see Figure 3) given in Example 1.2 where attachments are made at the points $x, y \in \vec{I} \times \vec{I}$ and $x', y' \in \vec{I}$ (see Figure 2). We have the following pushout diagram.

$$\begin{tikzcd}
\vec{I} \times \vec{I} \arrow[r, "\sim", "f"'] \arrow[d, tail, "j"'] & \vec{I} \arrow[d, tail] \\
B \arrow[r, "g"'] & C
\end{tikzcd}$$

Since the inclusion j is a cofibration, we get a weak equivalence between B and C. However as discussed in Example 1.2, from a certain point of view B should not be equivalent to C.

The solution to this problem is to work with $\mathbf{A} \downarrow \mathbf{LoPospc}$ instead of **LoPospc** where the choice of context $A \in \mathrm{Ob}\,\mathbf{LoPospc}$ depends on the pushouts that one would like to consider.

In the example above the right context is clearly the points $x, y \in \vec{I} \times \vec{I}$ and $x', y' \in \vec{I}$. So $A = \{a, b\}$, $\iota_{\vec{I}\times\vec{I}}(a) = x$, $\iota_{\vec{I}\times\vec{I}}(b) = y$, $\iota_{\vec{I}}(a) = x'$, and $\iota_{\vec{I}}(b) = y'$. As discussed in Example 2.17 the map f is not a dihomotopy equivalence rel A. So we are not forced to conclude that there is a weak equivalence between B and C.

[4] Again more details in the appendix (one passes to the category of simplicial presheaves [Dug01]).

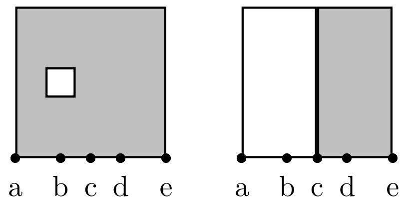


FIGURE 9. The spaces B and B'. Subspaces of $\vec{I}\times\vec{I}$ with a square removed and labelled points $\{a,b,c,d,e\}$

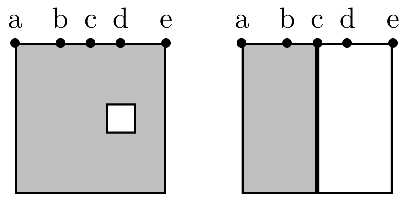


FIGURE 10. The spaces C and C'. Subspaces of $\vec{I}\times\vec{I}$ with a square removed and labelled points $\{a,b,c,d,e\}$

In the following two examples we examine the 'pastings' of two copies of $\vec{I}\times\vec{I}$ with a square removed. We show how choosing the right context allows us to find a one-dimensional sub-pospace which is dihomotopy equivalent to the pushout.

Unlike the previous section, we will not give explicit dihomotopy equivalences in these two examples.

Example 3.4. Let A be the discrete space $\{a,b,c,d,e\}$. Let B be the subspace of $\vec{I}\times\vec{I}$ in Figure 9 with the square $\{(x,y) \mid \frac{1}{5}<x<\frac{2}{5}, \frac{2}{5}<y<\frac{3}{5}\}$ removed. Let $\iota_B(a)=(0,0)$, $\iota_B(b)=(\frac{3}{10},0)$, $\iota_B(c)=(\frac{1}{2},0)$, $\iota_B(d)=(\frac{7}{10},0)$, and $\iota_B(e)=(1,0)$.

Let C be the subspace of $\vec{I}\times\vec{I}$ in Figure 10 with the square $\{(x,y) \mid \frac{3}{5}< x<\frac{4}{5}, \frac{2}{5}<y<\frac{3}{5}\}$ removed. Let $\iota_B(a)=(0,1)$, $\iota_B(b)=(\frac{3}{10},1)$, $\iota_B(c)=(\frac{1}{2},1)$, $\iota_B(d)=(\frac{7}{10},1)$, and $\iota_B(e)=(1,1)$.

Let B' be the subspace of $\vec{I}\times\vec{I}$ in Figure 9 with the square $\{(x,y) \mid 0<x<\frac{1}{2}, 0<y<1\}$ removed. Then there is a dihomotopy equivalence $f: B \xrightarrow{\simeq} B'$ rel A. One can construct the required dihomotopies by stretching the region $\frac{2}{5}\leq y\leq\frac{3}{5}$ first to $y=1$ and then to $y=0$. Next one stretches the region $\frac{1}{5}\leq x\leq\frac{2}{5}$ first to $x=\frac{1}{2}$ and then to $x=0$. All this is done while leaving the five marked points fixed.

Similarly there is a dihomotopy equivalence $g: C \xrightarrow{\simeq} C'$ rel A where C' is the subspace of $\vec{I}\times\vec{I}$ in Figure 10 with the square $\{(x,y) | \frac{1}{2}<x<1, 0<y<1\}$ removed.

Let D be the space obtained by attaching B along its bottom edge to the top edge of C. Notice that $D \in \operatorname{Ob}\mathbf{A}\downarrow\mathbf{Pospace}$ and the inclusions $i: B\to D$ and $j: C\to D$ are dimaps in $\mathbf{A}\downarrow\mathbf{Pospace}$.

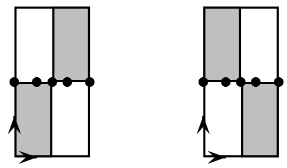


FIGURE 11. The pospaces F and F'

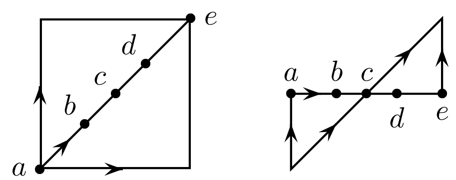


FIGURE 12. The pospaces G and G'

Now take the following pushout.

$$\begin{tikzcd} B \arrow[r, "\sim", "f"'] \arrow[d, tail, "i"'] & B' \arrow[d, tail] \\ D \arrow[r, "\sim", "f'"'] & E \end{tikzcd}$$

Then E is the pospace obtained by attaching the bottom edge of B' to the top edge of C. Since C includes into E we can take the following pushout.

$$\begin{tikzcd} C \arrow[r, "\sim", "g"'] \arrow[d, tail, "i"'] & C' \arrow[d, tail] \\ E \arrow[r, "\sim", "f''"'] & F \end{tikzcd}$$

Now F is the pospace in Figure 11 obtained by attaching the bottom edge of B' to the top edge of C'.

Finally F is dihomotopy equivalent rel A to the space G in Figure 12. Consider F and G as sub-pospaces of $\vec{I} \times \vec{I}$. The dihomotopy is obtained by first collapsing the square $[\frac{1}{2}, 1] \times [\frac{1}{2}, 1]$ using $(x, y) \mapsto (\max(x, y), \max(x, y))$, and then collapsing the square $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$ using $(x, y) \mapsto (\min(x, y), \min(x, y))$

Thus in the context of A, D is equivalent to G.

Example 3.5. Let A, B, C, B' and C' be as in the previous example, except that the marked points on B and B' are taken to be on the top edge, and the marked points on C and C' are taken to be on the bottom edge. Let D' be the space obtained by attaching C along its bottom edge to the top edge of B.

Then as in the previous example D' is weak equivalent to F' where F' is the pospace in Figure 11 obtained by attaching the bottom edge of C' to the top edge of B'.

Finally F' is dihomotopy equivalent rel A to the following space G' in Figure 12. Consider F' and G' as sub-pospaces of $\vec{I} \times \vec{I}$. The dihomotopy is obtained by collapsing the regions $[\frac{1}{2}, 1] \times [0, \frac{1}{2}]$ using $(x, y) \mapsto (x, \frac{1}{2})$, and then collapsing the square $[0, \frac{1}{2}] \times [\frac{1}{2}, 1]$ using $(x, y) \mapsto (x, \frac{1}{2})$.

Thus in the context of A, D' is equivalent to G'.

Example 3.6. Finally we give an example which requires a non-discrete context. Let $X = \vec{I} \times \vec{I}$. We will show that if we want to use X to construct a certain space Z then the appropriate context is the undirected unit interval I.

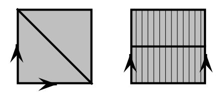


FIGURE 13. X and Y with the images of I marked

Let $\varphi : I \to X$ be the inclusion of the anti-diagonal, given by $t \mapsto (t, 1-t)$ (see Figure 13). Let $Y = I \times \vec{I}$ and let $\psi : I \to Y$ be the inclusion of the central line, given by $t \mapsto (t, \frac{1}{2})$ (see Figure 13). Define the pospace Z obtained by gluing X and Y together along the images of I, using the following pushout.

$$\begin{tikzcd} I \arrow[r, "\varphi"] \arrow[d, "\psi"'] & X \arrow[d, "\iota_X"] \\ Y \arrow[r, "\iota_Y"'] & Z \end{tikzcd}$$

We claim that if we want to consider this pushout then the proper context is $A = I$ with $\varphi : A \to X$ as given above.

For $\alpha \in I$ let $p_\alpha = \iota_X(\varphi(\alpha))$, $p^0_\alpha = \iota_Y(\alpha, 0)$ and $p^1_\alpha = \iota_Y(\alpha, 1)$. Notice that for $s \neq t \in I$ there does not exist a path in Z from p^0_s to p^1_t.

Now let A be some context and let $f : X \to X'$ be some dihomotopy rel A. Let Z' be defined by the following pushout.

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\iota_X"'] & X' \arrow[d] \\ Z \arrow[r, "g"'] & Z' \end{tikzcd}$$

Assume there exists $s \neq t \in I$ such that $f(p_s) = f(p_t)$. We claim that there is a path from $g(p^0_s)$ to $g(p^1_t)$. In Y (and hence in Z) there is a path from p^0_s to p_s and a path from p_t to p^1_t. The concatenation of the images of these paths under g gives the desired path in Z'.

Therefore the map g should not be an equivalence, and thus there should not have been an equivalence f such that $f(p_s) = f(p_t)$ for some $s \neq t \in I$. We can prevent this difficulty if we use the context $A = I$ together with $\varphi : A \to X$.

4. ACKNOWLEDGEMENT

I would like to thank Eric Goubault, Kathryn Hess, Krzysztof Worytkiewicz, and Emmanuel Haucourt for introducing me to the study of concurrency using topology and category theory.

Appendix A. Model Categories

In this section we define model categories, and show how a given small category can be embedded into a *universal model category*. For more details see [Dug01, Bub04].

Definition A.1. A *model category* is a category $\mathbf{C}$ with three distinguished classes of morphisms: weak equivalences, cofibrations, and fibrations satisfying the following conditions:

(1) $\mathbf{C}$ contains all small limits and colimits.
(2) If there exist morphisms f, g and $g \circ f$ and two of them are weak equivalences then so is the third.
(3) Weak equivalences, cofibrations, and fibrations are closed under retracts.
(4) Given any commutative diagram

$$\begin{tikzcd} A \arrow[r] \arrow[d, "i"'] & X \arrow[d, "p"] \\ B \arrow[r] & Y \end{tikzcd}$$

such that i is a cofibration and p is a fibration, then if either i or p is also a weak equivalence then there exists a map $B \to X$ making the diagram commute.
(5) Any map may be factored as a cofibration followed by a fibration which is a weak equivalence, and as a cofibration which is a weak equivalence followed by a fibration.

Next we define the category of simplicial presheaves.

Definition A.2.
- The simplicial category $\mathbf{\Delta}$ is the category whose objects are $[n] = \{0, 1, \dots, n\}$ for $n \geq 0$ and whose morphisms are maps $f : [n] \to [k]$ such that $x \leq y$ implies that $f(x) \leq f(y)$.
- The category of simplicial sets $\mathbf{sSet}$ is the category $\mathbf{Set}^{\mathbf{\Delta}^{\mathrm{op}}}$ whose objects are contravariant functors from $\mathbf{\Delta}$ to the category of sets $\mathbf{Set}$ and whose morphisms are natural transformations.
- Let $\mathbf{C}$ be a small category. Then $\mathbf{sPre}(\mathbf{C})$ is the category $\mathbf{sSet}^{\mathbf{C}^{\mathrm{op}}}$ whose objects are the contravariant functors from $\mathbf{C}$ to $\mathbf{sSet}$ and whose morphisms are natural transformations.

Remark A.3. An important fact is that there is an embedding $\mathbf{C} \to \mathbf{sPre}(\mathbf{C})$.

The category $\mathbf{sSet}$ has a model structure in which the cofibrations are the monomorphisms and the weak equivalences are the morphisms f such that $|f|$ the geometric realization of f is a weak equivalence in the category of topological spaces (that is, it induces isomorphisms between homotopy groups). For more details see [Hov99].

The category of simplicial presheaves has a canonical model structure, called the *cofibrant model structure*, where the weak equivalences and the cofibrations are defined objectwise. That is, a morphism f in $\mathbf{sPre}(\mathbf{C})$ is a weak equivalence or cofibration if and only if for each $X \in \mathrm{Ob}\,\mathbf{C}$ the morphism $f(X)$ is a weak equivalence or cofibration in $\mathbf{sSet}$.

Now one can localize this model category with respect some set of morphisms M to get a new model category $\mathbf{sPre}(\mathbf{C})/\mathbf{M}$. This model category has the same objects, but in addition to the previous weak equivalences, the morphisms in M are now weak equivalences. For example if $\mathbf{C} = \mathbf{LoPospc}$ then one could localize with respect to all dihomotopy equivalences (it makes sense to say this because of the embedding of $\mathbf{C}$ in $\mathbf{sPre}(\mathbf{C})$).

Appendix B. Local pospaces

In this section we give a precise definition of the category **LoPospc** of local pospaces and use it to give a more precise version of Theorem 3.2. Local pospaces are defined in [FGR99, Bub04]. Here we follow [Bub04].

Definition B.1.
- Given a topological space M, an *order atlas* on M is an open cover[5] $U = \{U_i\}$ indexed by a set I such that each U_i is a pospace and that the orders are compatible. That is, given $x, y \in U_i \cap U_j$, $x \leq_i y$ if and only if $x \leq_j y$.
- Let $U = \{U_i\}$ and $V = \{V_j\}$ be two order atlases. Then V is said to be a *refinement* of U if for any U_i and any $x \in U_i$ there exists a V_j containing x which is a sub-pospace of U_i.
- Two order atlases are said to be *equivalent* if they have a common refinement. One can check that this defines an equivalence relation.
- Define a *local pospace* to be a topological space together with an equivalence class of order atlases.
- Define a *dimap of local pospaces* $f : (M, \bar{U}) \to (N, \bar{V})$ to be a continuous map $f : M \to N$ such that for any choice of $V = \{V_j\} \in \bar{V}$ there is some choice of $U = \{U_i\} \in \bar{U}$ such that for all i, j the partial map $f : U_i \to V_j$ is a dimap of pospaces.

Definition B.2. Define **LoPospc** to be the category whose objects are local pospaces whose underlying topological spaces are subsets of $\mathbb{R}^n$ for some n, and whose morphisms are dimaps between local pospaces.

Remark B.3. Notice that we have restricted the class of local pospaces in our category. This is done precisely so that **LoPospc** is a small category, which is used to apply the machinery of Appendix A. For the purposes of concurrency, this does not seem to be a significant limitation. Furthermore, it may be possible that any local pospace can be 'found' in $\mathbf{sPre}(\mathbf{LoPospc})$.

Nevertheless, a consequence of this, is that the category **Pospace** in Definition 2.2 is not a subcategory of **LoPospc**. Of course one could define a new category $\mathbf{Pospace}'$ whose objects are those pospaces whose underlying topological spaces are subsets of $\mathbb{R}^n$ for some n. Then $\mathbf{Pospace}'$ is a subcategory of **LoPospc**. All of our examples are in $\mathbf{Pospace}'$.

We can now give a more precise version of Theorem 3.2.

Theorem B.4 ([Bub04])**.** *There exists a model structure on* $\mathbf{sPre}(\mathbf{LoPospc})$ *such that cofibrations are the monomorphisms. Furthermore the morphisms in* **LoPospc** *which are weak equivalences in* $\mathbf{sPre}(\mathbf{LoPospc})$ *are just the isomorphisms.*

[5]That is, each U_i is an open subset of M, and $M = \cup_{i \in I} U_i$.

References

[Bub04] Peter Bubenik. Towards a model category structure for local pospaces. 2004, http://igat.epfl.ch/bubenik/papers/.

[Dug01] Daniel Dugger. Universal homotopy theories. *Adv. Math.*, 164(1):144–176, 2001.

[FGR99] Lisbeth Fajstrup, Eric Goubault, and Martin Raussen. Algebraic topology and concurrency. *to appear in Theoretical Computer Science*, 1999. Also preprint R-99-2008, Dept. of Mathematical Sciences, Aalborg University, Aalborg, Denmark.

[FRGH04] L. Fajstrup, M. Raussen, E. Goubault, and E. Haucourt. Components of the fundamental category. *Appl. Categ. Structures*, 12(1):81–108, 2004. Homotopy theory.

[Gau03] Philippe Gaucher. A model category for the homotopy theory of concurrency. *Homology Homotopy Appl.*, 5(1):549–599, 2003.

[Gou95] Eric Goubault. *The geometry of concurrency*. PhD thesis, Ecole Normale Supèrieure, 1995, http://www.dmi.ens.fr/ goubault.

[Gra02] Marco Grandis. Directed homotopy theory. II. Homotopy constructs. *Theory Appl. Categ.*, 10:No. 14, 369–391 (electronic), 2002.

[Gra03] Marco Grandis. Directed homotopy theory. I. *Cah. Topol. Géom. Differ. Catég.*, 44(4):281–316, 2003.

[Hov99] Mark Hovey. *Model categories*, volume 63 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 1999.

[Rau03] Martin Raussen. State spaces and dipaths up to dihomotopy. *Homology Homotopy Appl.*, 5(2):257–280 (electronic), 2003. Algebraic topological methods in computer science (Stanford, CA, 2001).

Institut de Géométrie, Algèbre et Topologie, Bâtiment BCH, 1015-Lausanne, Switzerland

E-mail address: peter.bubenik@epfl.ch

URL: http://igat.epfl.ch/bubenik/

A framework for component categories

Emmanuel Haucourt

June 2, 2004

Abstract

This paper provides further developments in the study of component categories which have been intrduced in [FGHR04]. In particular, the component category functor is seen as a left adjoint hence preserves pushouts. This property is applied to prove a Van Kampen like theorem for component categories[1]. The original purpose of component categories is to suitably reduce the size of fundamental categories[2]. We take advantage of this fact to define the cohomology of a directed geometrical shape as the cohomology of its component category[3].

1 Introduction

Given a small category $\mathcal{C}$ and a subcategory Σ of $\mathcal{C}$, we define the **quotient category** $\mathcal{C}/_\Sigma$ applying results developed in [BBP99]. Indeed, the size of $\pi_1(\overrightarrow{X})/_\Sigma$ decreases as the one of Σ increases. As one can expect, if $\Sigma = \pi_1(\overrightarrow{X})$ then $\pi_1(\overrightarrow{X})/_\Sigma$ is $\{*\}$. Then the **component category** of a pospace $\overrightarrow{X}$ is defined as $\pi_1(\overrightarrow{X})/_\Sigma$ where Σ is the **greatest weak-equivalences subcategory** of $\mathcal{C}$ and $\pi_1(\overrightarrow{X})$ the fundamental category of $\overrightarrow{X}$. We have in mind that Σ is made of the dipaths[4] of $\overrightarrow{X}$[5] along which "no choice is made" so we do not lose information removing them[6].

The previous construction can be done in a category whose objects are taken in the class of pairs $(\mathcal{C}, \Sigma)$, such a pair is called a system over $\mathcal{C}$, where $\mathcal{C}$ is an object of a subcategory of **CAT** and $\Sigma \subseteq \mathcal{C}$. The idea is to equip the objects $\mathcal{C}$ of a sub-category of **CAT** with a sub-poset of the poset of all subcategories of $\mathcal{C}$. Then we define the quotient functor sending $(\mathcal{C}, \Sigma)$ on $\mathcal{C}/_\Sigma$. The component category is obtained when Σ is optimal i.e. when the size of $\mathcal{C}/_\Sigma$ is minimal without loss of relevant information. Several examples are given, involving different subcategories of **CAT**, and we define component categories of **pospaces**, **local pospaces** and **d-spaces**.

Some proofs of technical points are skipped and the paper is organized in the following way:

(i) Pospaces, local pospaces and d-spaces are defined. Concrete but informal examples are given to make the reader understand what component categories should be.

(ii) General congruences and some related tools are described.

(iii) A general theorem describes a framework in which the component category functor can be defined. As we shall see, this theorem makes the component category functor a left adjoint.

(iv) The previous theorem is applied to define the component category of pospaces, local pospaces and d-spaces. We check that we have obtained what was expected.

(v) Preservation of colimits by component category functor is applied to prove Van Kampen like theorems for component category (instead of fundamental category). Examples are given.

[1]This last point is very important to make effective calculations.
[2]in concrete examples fundemental category is as "big" as $\mathbb{R}$ while the component category is "finitely generated".
[3]Cohomology of small categories is defined in [BW85] and [Bau91].
[4]or execution traces from a computer scientist point of view.
[5]seen as the space of states of a computer on which a program runs.
[6]precisely, they are not removed but turned into identities.

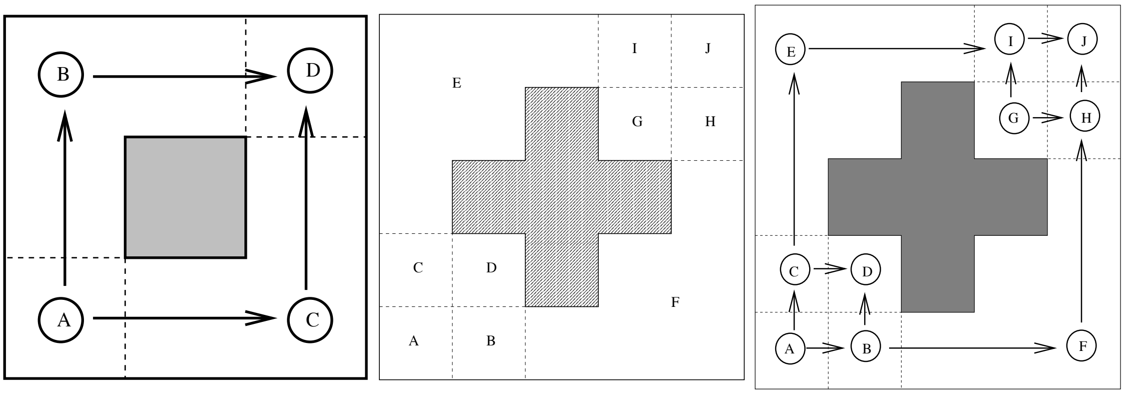


Figure 1: Square with centered hole and Swiss flag

(vi) A form of directed cohomology is defined as the cohomology of the component category.

2 Geometrical intuition of component categories through examples

Component categories first appear in [FGHR04] in order to reduce the size of the fundamental category. Pospaces are certainly the simplest model of directed topology one may find.

Definition 1 (Pospaces) *A* **pospace** *is a triple $(X, \tau_X, \leq_X)$ where (X, τ_X) is a topological space, $(X, \leq_X)$ is a poset and $\leq_X$ is a closed subset of $(X, \tau_X) \times (X, \tau_X)$. A* **dimap** *from a pospace $(X, \tau_X, \leq_X)$ to a pospace $(Y, \tau_Y, \leq_Y)$ is a set-theoretic function f from X to Y inducing a continuous map from (X, τ_X) to (Y, τ_Y) and an increasing map from $(X, \leq_X)$ to $(Y, \leq_Y)$. The collection of po-spaces together with dimaps between them form a category denoted* **POSPC**. *Isomorphisms of* **POSPC** *are called* **dihomeomorphisms** *and are bijective (one-to-one, onto) dimaps whose inverse is also a dimap. Monomorphisms (respectively epimorphisms) are one-to-one (respectively onto) dimaps.*

The unit segment $\overrightarrow{[0,1]}$ with classical topology and order is a pospace as well as all its products with product topology and order. $\overrightarrow{[0,1]}$ is in fact the "standard" example in the sense that it is the cogenerator[7] of the category of compact pospaces[8]. The examples of figure 1 are built up from the unit square with classical topology and order in which "holes" have been dug. In each case the underlying spaces are divided into "components" which give the set of objects of the component category, their borders are drawn with the dotten lines. Two components sharing a frontier are "neighbours" and we put a unique "prime arrow" between neihbour components, the source component being the left most bottom most one. The morphisms of the component category are "generated" by those "prime arrows". In the first example, the component category is free, in the "swiss flag" example (figure 1) it is not the case any more because we have $\xrightarrow{BD} \circ \xrightarrow{AB} = \xrightarrow{CD} \circ \xrightarrow{AC}$. The two examples of figure 2 are not dihomeomorphic since their component categories are clearly not isomorphic, it suffices to compare how many morphisms go from the left most bottom most object to the right most upper most one.
Before going further in the study of examples, let me emphasize the fact that we "read" the dipaths of the pospace in its component category. In mathematical terms, we have a lifting property which says that any morphism of the fundamental category is represented by a unique morphism of the component category, conversely, any morphism of the component category represents a morphism of the fundamental one. This property can be found in [FGHR04], it is also given for free provided we define the component category by

[7] see [Bor94a] for the definition.
[8] i.e. the underlying topological space is compact.

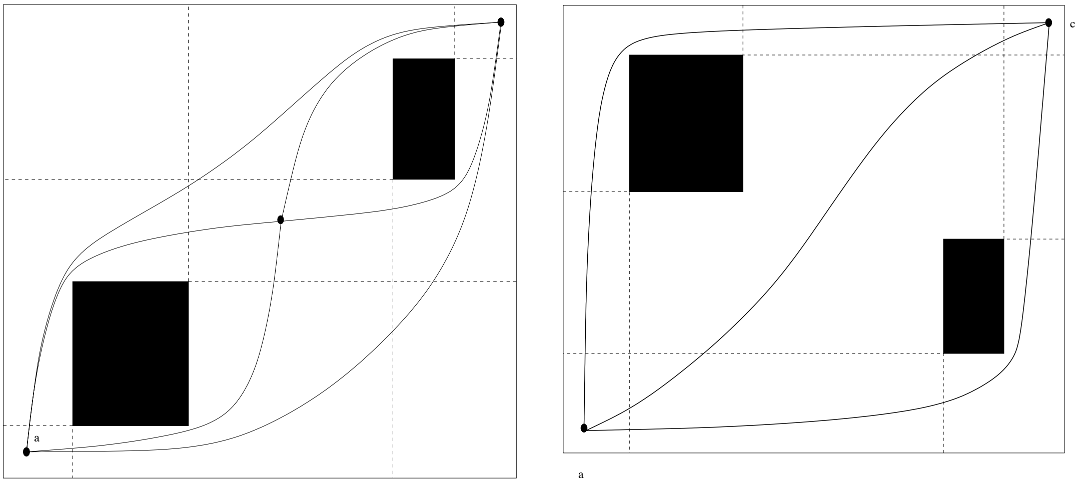


Figure 2: Two possible configurations of two holes in a square

means of generalized congruences, see [BBP99] and the description of the component category given in the rest of the paper. Next examples are 3−dimensional, the unit cube (with classical topology and order) with a centered hole is shown on the left side of figure 3. The right side picture depicts its components, whose border are represented by "walls":

In figure 5, the blue parallelepipeds are holes and the red cube is a deadlock area, i.e. any dipath entering in it will not go beyond the deep right upper corner of the red cube. On the right side, the corresponding component category is depicted, but the conventions of representation are different, vertices are components, edges are elementary arrows and faces represent relations between morphisms. By the way, this convention of representation induces a "dimensional duality", components are 3−dimensional subpospaces of the cube and they are represented by points, which are 0−dimensional. Faces of the components are 2−dimensional subpospaces and they are represented by "elementary arrows" (hence $1 - dimensional$) from component to the neighbour it shares the face with. A segment shared by four faces is a 1− dimensional subpospace and is represented by a relation

$$\begin{tikzcd} & \cdot & \\ \cdot \arrow[ur, "\beta"] & \Leftrightarrow & \cdot \arrow[ul, "\delta"'] \\ & \cdot \arrow[ul, "\alpha"] \arrow[ur, "\gamma"'] & \end{tikzcd}$$

between the four "elementary arrows" representing the four faces. This relation can be seen as a $2 - dimensional$ arrow provided we turn the component category into a 2−category adding a trivial groupoid between $\beta \circ \alpha$ and $\delta \circ \gamma$. One can even go further with a point of the pospace shared by six segments all of them being shared by four faces, which makes us reach 3−categories, see figure 4.

This "duality" property has been practically applied by Eric Goubault to write a program which provides a 3-dimensional "view" of the component category of the 3-dimensional pospaces. A detailed description of the method is available in [Gou03a]. The right side picture of figure 5 has been produced by this program.

3 Generalized congruences

This section is devoted to generalized congruences have been formalized in [BBP99].

Definition 2 (Generalized Congruences [BBP99]) *A generalized congruence on a small category $\mathcal{C}$, is an equivalence relation $\sim_o$ on $Ob(\mathcal{C})$ and a partial equivalence relation $\sim_m$ on $Mo(\mathcal{C})^+$ (the set of all non-empty finite sequences of morphisms of $\mathcal{C}$) satisfying the following conditions ($\cdot$ is the usual concatenation, the α's, β's and γ's range over $Mo(\mathcal{C})$):*

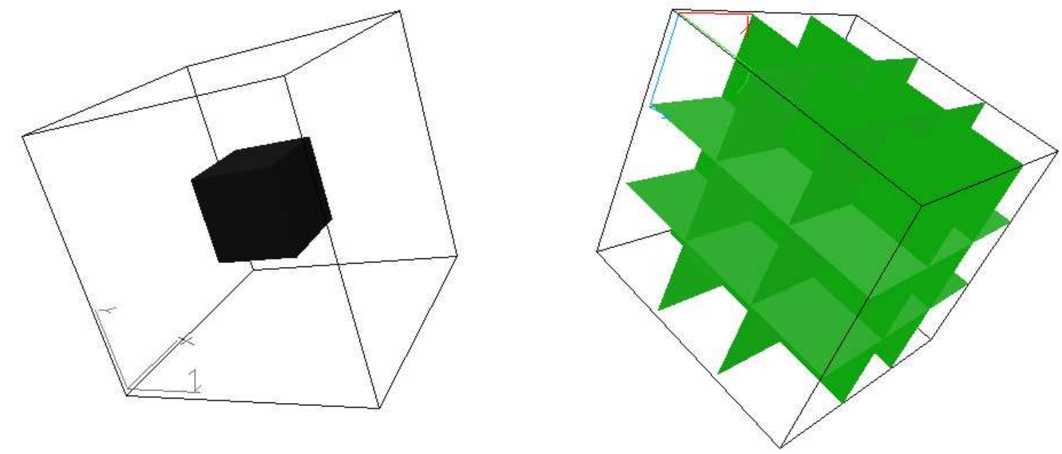
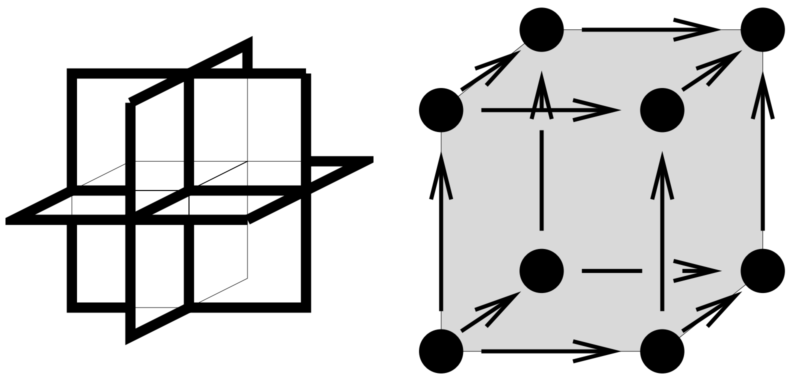


Figure 3: The cube with a centered hole



The point at the center of the left side figure is represented by a 3−morphism (the grey filling on the right side figure) in a suitable 3−category built up from the component category.

. This example is related to 3−philosophers problem.

Figure 4: Dimensional duality

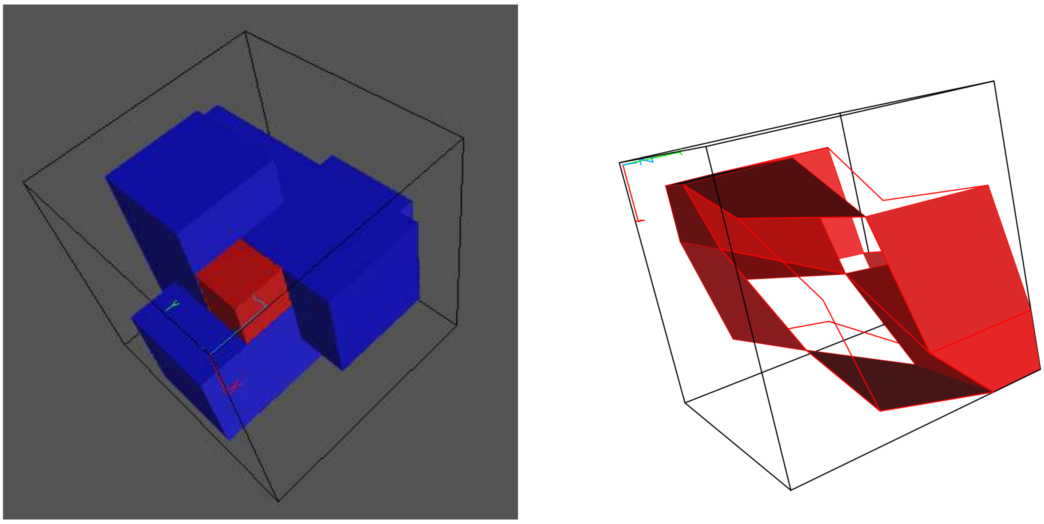


Figure 5: Three philosophers diner

- $(\beta_n, ..., \beta_0) \cdot (\alpha_p, ..., \alpha_0) \sim_m (\gamma_q, ..., \gamma_0) \Rightarrow tgt(\alpha_p) \sim_o src(\beta_0)$
- $(\beta_n, ..., \beta_0) \sim_m (\alpha_p, ..., \alpha_0) \Rightarrow tgt(\beta_n) \sim_o tgt(\alpha_p)$ *and* $src(\beta_0) \sim_o src(\alpha_0)$
- $x \sim_o y \Rightarrow id_x \sim_m id_y$
- $(\beta_n, ..., \beta_0) \sim_m (\alpha_p, ..., \alpha_0)$ *and* $(\delta_q, ..., \delta_0) \sim_m (\gamma_r, ..., \gamma_0)$ *and* $tgt(\beta_n) \sim_o src(\delta_0) \Rightarrow$ $(\delta_q, ..., \delta_0) \cdot (\beta_n, ..., \beta_0) \sim_m (\gamma_r, ..., \gamma_0) \cdot (\alpha_p, ..., \alpha_0)$
- $src(\beta) = tgt(\alpha) \Rightarrow (\beta \circ \alpha) \sim_m (\beta, \alpha)$

Theorem 1 (Quotient Category [BBP99]) *Given* $(\sim_o, \sim_m)$ *a generalized congruence on a small category* $\mathcal{C}$*, we define the* **quotient category** $\mathcal{C}/_\sim$ *by*

- $Ob(\mathcal{C}_{/\sim}) := \{[x]_{\sim_o} / x \in Ob(\mathcal{C})\}$
- $src([(\gamma_n, ..., \gamma_0)]_{\sim_m}) = [src(\gamma_0)]_{\sim_o}$ *and* $tgt([(\gamma_n, ..., \gamma_0)]_{\sim_m}) = [tgt(\gamma_n)]_{\sim_o}$
- $[(\beta_n, ..., \beta_0)]_{\sim_m} \circ [(\alpha_p, ..., \alpha_0)]_{\sim_m} = [(\beta_n, ..., \beta_0) \cdot (\alpha_p, ..., \alpha_0)]_{\sim_m}$

Moreover, there is a **quotient** *functor* $Q_\sim : \mathcal{C} \to \mathcal{C}_\sim$*, defined by* $Q_\sim(x) = [x]_{\sim_o}$ *and* $Q_\sim(\gamma) = [\gamma]_{\sim_m}$*.* $Q_\sim$ *enjoys the following universal property, for any functor* $f : \mathcal{C} \to \mathcal{C}_2$*, if* $\sim \subseteq \sim_f$ *then* $\exists! g : \mathcal{C}_{/\sim} \to \mathcal{C}_2$ *making the following diagram commutes*

$$\begin{tikzcd} & \mathcal{C}_{/\sim} \arrow[dr, dashed, "g"] \arrow[d, phantom, "="] & \\ \mathcal{C} \arrow[ur, "Q_\sim"] \arrow[rr, "f"'] & & \mathcal{C}_2 \end{tikzcd}$$

Still, we have the following facts :

- g *is a monomorphism iff* $\sim_f = \sim$
- $\sim_{Q_\sim} = \sim$
- $Q_\sim$ *is an extremal epimorphism*

Lemma 1 ([BBP99]) *Generalized congruences on a given small category, ordered by componentwise inclusion form a complete lattice whose meets are componentwise intersections. The total relation which identifies all objects and all non-empty finite sequences of morphisms is a generalized congruence, precisely $\top$ of the lattice, while $(=_{Ob(\mathcal{C})}, \emptyset)$ is $\bot$. Thus, for an arbitrary pair of relations R_o on $Ob(\mathcal{C})$ and R_m on $Mo(\mathcal{C})^+$, there is a least generalized congruence containing (R_o, R_m).*

4 The Component Category functor

4.1 Loop Free, One Way and Directed categories

Pureness first apprears in [FGHR04] and is an unavoidable technical tool to study component categories, indeed, good properties of $\mathcal{C}/_\Sigma$ directly depend on pureness of Σ. In ideas, if Σ consists of execution paths[9] along with nothing happens then if $\beta \circ \alpha \in \Sigma$ it is expectable that $\beta, \alpha \in \Sigma$ too. It is also a convenient way to define loop free, one way and directed categories.

Definition 3 *A sub-category $\mathcal{B}$ of $\mathcal{C}$ is* **pure** *in $\mathcal{C}$ iff $\forall f, g$ morphisms of $\mathcal{C}$ with $src(g) = tgt(f)$, $g \circ f \in \mathcal{B} \Rightarrow f, g \in \mathcal{B}$.*

Pureness is a kind of generalization of convexity in poset framework, indeed, a subposet $(A, \leq_A)$ of a poset $(X, \leq_X)$ is convex iff A' is a pure subcategory of X', where A' and X' are the small categories coresponding to A and X.

Definition 4 *A* **loop free cateory** *is a category whose subcategory of endomorphisms is* **pure** *and* **discrete** [10].
A **one way cateory** *is a category whose subcategory of isomorphisms is* **pure** *and* **discrete**.[11]
A **directed cateory** *or* **d-category** *is a category whose subcategory of isomorphisms is* **pure**. *Loop free, one way and directed small categories respectively form full, complete and co-complete subcategories of* **CAT** *respectively denoted* **LFCAT**, **OWCAT**, **dCAT**.

The fundamental category of a pospace is obviously loop free, the one of a local pospace is one way, but it is much harder to prove, and I conjecture that the one of a d-space is directed, it is in fact the reason why I called them "directed", roughly speaking, it comes from the fact that dX is stable under direparametrization (see definition 12).

Conjecture 1 *We have the inclusion functors* $\mathbf{LFCAT} \hookrightarrow \mathbf{OWCAT} \hookrightarrow \mathbf{dCAT} \hookrightarrow \mathbf{CAT}$ *each of which having a left adjoint* [12]. *Moreover, we have the following commutative diagram*

$$\begin{tikzcd} \mathbf{POSPC} \arrow[r, hook] \arrow[d, "\pi_1"] & \mathbf{LPOSPC} \arrow[r, hook] \arrow[d, "\pi_1"] & \mathbf{dSPC} \arrow[d, "\pi_1"] \\ \mathbf{LFCAT} \arrow[r, hook] & \mathbf{OWCAT} \arrow[r, hook] & \mathbf{dCAT} \end{tikzcd}$$

4.2 Weak Equivalences Subcategory

Next materials are directly related to the choice of a Σ such that $\mathcal{C}/_\Sigma$ is the component category of $\mathcal{C}$. As we shall see, all the rest of the subsection, in particular the existence of a non empty Weak Equivalences subcategory, holds for any directed category $\mathcal{C}$. Then the component category of a pospace/local pospace/directed space, is defined as the component category of its fundamental category.

[9] it is a computer science point of view.

[10] i.e. for all diagram $\begin{tikzcd} \cdot \arrow[r, bend left, "\alpha"] & \cdot \arrow[l, bend left, "\beta"] \end{tikzcd}$ in $\mathcal{C}$, α and β are identities. Hence $\mathcal{C}$ has no "loops", whence the name.

[11] A one way category might have loops, but each loop is either clockwise or unclockwise never both at the time.

[12] in fact, **LFCAT**, **OWCAT**, **dCAT** are respectively reflective subcategories of **OWCAT**, **dCAT**, **CAT** but it is of no use in this paper.

4.2.1 Yoneda inversible morphisms, Left/Right extension properties and Weak Equivalences Subcategories

Definition 5 ([FGHR04]) *Let $\mathcal{C}$ be a category. A morphism σ of $\mathcal{C}$ is said to be* **Yoneda revertible** *iff* $\forall x \in Ob(\mathcal{C}), (\mathcal{C}[x, src(\sigma)] \neq \emptyset \Rightarrow \gamma \in \pi_1(\mathcal{C})[x, src(\sigma)] \mapsto \sigma \circ \gamma)$ *is bijective and* $\forall y \in Ob(\mathcal{C}), (\mathcal{C}[tgt(\sigma), y] \neq \emptyset \Rightarrow \gamma \in \mathcal{C}[tgt(\sigma), y] \mapsto \gamma \circ \sigma)$ *is bijective.*

Definition 5 is closely related to representable functors of $\mathcal{C}$ and Yoneda's lemma (see [Bor94a]), however, the restriction $\forall x \in Ob(\mathcal{C}), (\mathcal{C}[x, src(\sigma)] \neq \emptyset$... and $\forall y \in Ob(\mathcal{C}), (\mathcal{C}[tgt(\sigma), y] \neq \emptyset$... cannot be removed, otherwise, a Yoneda inversible morphism would necessarily be an isomorphism which is silly for loop free and one way categories whose only isomorphisms are identities. From a computer science point of view, the subtle difference between Yoneda inversible morphisms and isomorphisms give an theoretical method for deadlock detection, but we will not develop this remark here. In all examples given in section 2, any dipath joining two points of the same component give rise to a Yoneda inversible morphism of the fundamental category.

Lemma 2 *Let $\mathcal{C}$ be any (small) category, x, y objects of $\mathcal{C}$ and $\sigma_1, \sigma_2 \in \mathcal{C}[x, y]$ Yoneda inversible, $\exists f_1, f_2 \in Iso(\mathcal{C})[y, y], \sigma_2 = f_1 \circ \sigma_1, \sigma_1 = f_2 \circ \sigma_2$ and $\exists g_1, g_2 \in Iso(\mathcal{C})[x, x], \sigma_2 = \sigma_1 \circ g_1, \sigma_1 = \sigma_2 \circ g_2$.*

PROOF. $\mathcal{C}[y, y] \neq \emptyset$ hence by definition of Yoneda inversible applied to σ_1, $\exists! f_1 \in \mathcal{C}[y, y]$ such that $\sigma_2 = f_1 \circ \sigma_1$. Exchanging σ_1 and σ_2, $\exists! f_2 \in \mathcal{C}[y, y]$ such that $\sigma_1 = f_2 \circ \sigma_2$. In particular, $\sigma_2 = f_1 \circ (f_2 \circ \sigma_2) = (f_1 \circ f_2) \circ \sigma_2$ and $\sigma_1 = f_2 \circ (f_1 \circ \sigma_1) = (f_2 \circ f_1) \circ \sigma_1$, but, by definition of Yoneda inversible, id_y is the only morphism of $h \in \mathcal{C}[y, y]$ such that $\sigma_2 = h \circ \sigma_2$. It is also only morphism of $h \in \mathcal{C}[y, y]$ such that $\sigma_1 = h \circ \sigma_1$. It follows that $f_1 \circ f_2 = f_2 \circ f_1 = id_y$ i.e. $f_1, f_2 \in Iso(Mo(\mathcal{C}))$. It works the same way for the g's.

Corollary 1 *Let $\mathcal{C}$ be a (small) category such that $Iso(\mathcal{C})$ is dicrete, then given x, y objects of $\mathcal{C}$, $\mathcal{C}[x, y] \cap \{Yoneda\ inversibles\}$ is either $\emptyset$ or a singleton.*

Remark 1 *Any isomorphism is Yoneda inversible morphism, and a composition of Yoneda inversible morphisms is a Yoneda inversible morphism. Moreover, if $\mathcal{L}$ is loop-free and σ is a Yoneda inversible morphism of $\mathcal{L}$ then $\mathcal{L}[src(\sigma), tgt(\sigma)] = \{\sigma\}$.*

To prove the last point, note that $\gamma \in \mathcal{L}[src(\sigma), src(\sigma)] \longmapsto \sigma \circ \gamma \in \mathcal{L}[src(\sigma), tgt(\sigma)]$ is a bijection. Up to now, this definition has only proved its relevance in loop-free cases. First, we recall from [FGHR04] that **the Σ-zigzag connected component of x in $\mathcal{L}$** denoted C_x is the subcategory of $\mathcal{L}$ whose objects are those connected to x by a zigzag of morphisms of Σ and satisfying for all objects y, z of C_x, $C_x[y, z] = \mathcal{L}[y, z] \cap \Sigma$.

Definition 6 R*ight* **E***xtension* **P***roperty*
Σ has the right extension property with respect to $\mathcal{C}$ iff $\forall \gamma : y' \longrightarrow x'$, $\forall \sigma : x \longrightarrow x' \in \Sigma$, $\exists \sigma' : y \longrightarrow y'$, $\exists \gamma' : y \longrightarrow x$ such that $\sigma \circ \gamma' = \gamma \circ \sigma'$, i.e. the following diagram is commutative:

$$\begin{tikzcd}
& y \arrow[dl, dashed, "\exists\sigma'\in\Sigma"'] \arrow[dr, dashed, "\exists\gamma'"] & \\
y' \arrow[dr, "\forall\gamma"'] & & x \arrow[dl, "\forall\sigma\in\Sigma"] \\
& x' &
\end{tikzcd}$$

Left **E**extension **P**rpoerty is obtained "dualizing" definition 6

Definition 7 (Eric Goubault) [13] *Let $\mathcal{C}$ be a small category, $\Sigma \subseteq Mo(\mathcal{C})$ is a WE-subcategory iff (by definition) Σ is stable under composition (of $\mathcal{C}$) and satisfies*

[13] in directed categories framework, this definition is equivalent to Eric Goubault's one.

1 $Iso(\mathcal{C}) \subseteq \Sigma \subseteq Yoneda(\mathcal{C})$[14]

2 Σ *is stable under pushouts and pullbacks (with any morphism in* $\mathcal{C}$*), it means that we* Σ *has both* **REP** *and* **LEP** *with respect to* $\mathcal{C}$ *and further the commutative squares provided by* **REP** *and* **LEP** *can be chosen in order to be respectively* **pullback** *and* **pushout** *squares in* $\mathcal{C}$*.*

Eric Goubault, in [Gou03b], has changed the definition of Weak Equivalences subcategory of [FGHR04][15] replacing left and right extension axiom by pushout/pullback stability axiom, providing an extremly handy tool. Indeed, any WE-subcategory of any small category $\mathcal{C}$ is pure (will be proved later) and has left and right extension properties (it is obvious). Moreover, if $Iso(\mathcal{C})$ is pure in $\mathcal{C}$ (i.e. $\mathcal{C}$ is directed) then $\mathcal{C}$ has a $\subseteq$-biggest WE-subcategory.

4.2.2 Locale of the Weak Equivalences of a small category

We give several results which will be combined to prove that the collection of WE-subcategories of a small category $\mathcal{C}$ such that $Iso(\mathcal{C})$ is pure in $\mathcal{C}$ forms locale. We recall that a **locale** is a poset $(L, \leq_L)$ such that $\forall U \subseteq L, U$ has a least upper bound and a greatest lower bound (it is a complete lattice) and $\forall (b_j)_{j\in J} \in L^J$ $\forall a \in L,\ a \wedge \left(\bigvee_{j\in J} b_j\right) = \bigvee_{j\in J}(a \wedge b_j)$ (see [Bor94b] or [Joh82]). Lemma 3 is due to Eric Goubault, it is the reason for definition 7. Indeed, in [FGHR04], we had to enforce the pureness of Σ by an axiom, unfortunately, the resulting definition was not "stable" in the sense that the subcategory generated by two pure subcategory is not, in general, pure.

Lemma 3 *Let* $\mathcal{C}$ *be a small category such that* $Iso(\mathcal{C})$ *is pure in* $\mathcal{C}$*. Then any WE-subcategory of* $\mathcal{C}$ *is pure in* $\mathcal{C}$*.*

PROOF. Take $\sigma \in \Sigma$ and $f_1, f_2 \in Mo(\mathcal{C})$ such that $\sigma = f_2 \circ f_1$. By 3^{rd} point of definition 7, we have a $\sigma' \in \Sigma$ and f_1' which form a pushout square and a unique $g \in Mo(\mathcal{C})$ making the push-out diagram

$$\begin{tikzcd} & \cdot & \\ & \cdot \arrow[u, "g"] & \\ \cdot \arrow[ur, dashed, "f_1'"] \arrow[uur, bend left, "id"] \arrow[rr, phantom, "pushout"] & & \cdot \arrow[ul, dashed, "\sigma'"'] \arrow[uul, bend right, "f_2"'] \\ & \cdot \arrow[ul, "\sigma"] \arrow[ur, "f_1"'] & \end{tikzcd}$$

commutative. By pureness of $Iso(\mathcal{C})$ in $\mathcal{C}$, f_1' and g are isomorphisms, hence by 2^{nd} point of definition 7, belongs to Σ. So by 1^{st} point of definition 7, $f_2 = g \circ \sigma' \in \Sigma$. The same way, using the pull-back (instead of push-out) extension property, one proves that $f_1 \in \Sigma$. Thus Σ is pure in $\mathcal{C}$.

Lemma 4 *Let* $\mathcal{C}$ *be a small category. If* $Iso(\mathcal{C})$ *is pure in* $\mathcal{C}$ *then* $Iso(\mathcal{C})$ *is a WE-subcategory of* $\mathcal{C}$*.*

PROOF. 1^{st} point of definition 7 is obviously satisfied. So is the 2^{nd} because if $g \circ f = id$ then, by pureness of $Iso(\mathcal{C})$, $f, g \in Iso(\mathcal{C})$. Any isomorphism is Yoneda inversible (remark 1) hence the 4^{th} point in satisfied. Let $\sigma \in Iso(\mathcal{C})$ and $f \in Mo(\mathcal{C})$ be, then we have

$$\begin{tikzcd} & \cdot & \\ \cdot \arrow[ur, dashed, "f\circ\sigma^{-1}"] \arrow[rr, phantom, "pushout"] & & \cdot \arrow[ul, dashed, "id"'] \\ & \cdot \arrow[ul, "\sigma"] \arrow[ur, "f"'] & \end{tikzcd}$$

and

$$\begin{tikzcd} & \cdot & \\ \cdot \arrow[ur, "\sigma"] \arrow[rr, phantom, "pullback"] & & \cdot \arrow[ul, "f"'] \\ & \cdot \arrow[ul, dashed, "\sigma^{-1}\circ f"] \arrow[ur, dashed, "id"'] & \end{tikzcd}$$

so the 3^{rd} point is also satisfied.

Lemma 5 *If* $(\Sigma_j)_{j\in J}$ *is a non empty family of WE subcategories of a small category* $\mathcal{C}$ *then* $\bigcap_{j\in J} \Sigma_j$ *is a WE-subcategory of* $\mathcal{C}$*.*

[14] $Iso(\mathcal{C})$ and $Yoneda(\mathcal{C})$ are subcategories of $\mathcal{C}$ respectively generated by isomorphisms and Yoneda inversible morphisms of $\mathcal{C}$.

[15] definition of [FGHR04] was itself inspired by the notion of calculus of fractions, see [GZ67] and [Bor94a].

Proof. $\bigcap_{j\in J}\Sigma_j$ obviously enjoys the $1^{st}, 2^{nd}$ and 4^{th} point of definition 7. Suppose $\sigma \in \bigcap_{j\in J}\Sigma_j$ and $f \in Mo(\mathcal{C})$ with $src(f) = src(\sigma)$. Take $j_1, j_2 \in J$, since $\sigma \in \Sigma_{j_1}$ we have a push out square

$$\begin{tikzcd} & x_1 & \\ {} \arrow[ur, dashed, "f'_1"] & \text{pushout} & {} \arrow[ul, dashed, "\sigma'_1\in\Sigma_{j_1}"'] \\ & {} \arrow[ul, "\sigma"] \arrow[ur, "f"'] & \end{tikzcd}$$

and also

$$\begin{tikzcd} & x_2 & \\ {} \arrow[ur, dashed, "f'_2"] & \text{pushout} & {} \arrow[ul, dashed, "\sigma'_1\in\Sigma_{j_2}"'] \\ & {} \arrow[ul, "\sigma"] \arrow[ur, "f"'] & \end{tikzcd}$$

because $\sigma \in \Sigma_{j_2}$. By uniqueness (up to isomorphism) of the pushout, we have an isomorphism τ from x_2 to x_1 such that $\sigma'_1 = \tau\circ\sigma'_2$. By 2^{nd} point of definition 7 and for any isomorphism is a retract (both left and right), $\tau \in \Sigma_{j_2}$ which is stable under composition (1^{st} point), thus $\sigma'_1 = \tau\circ\sigma'_2 \in \Sigma_{j_2}$.

By the same argument, $\forall j \in J, \sigma'_1 \in \Sigma_j$ i.e. $\sigma'_1 \in \bigcap_{j\in J}\Sigma_j$ and we have

$$\begin{tikzcd} & {} & \\ {} \arrow[ur, dashed, "f'_1"] & \text{pushout} & {} \arrow[ul, dashed, "\sigma'_1\in\bigcap_{j\in J}\Sigma_j"'] \\ & {} \arrow[ul, "\sigma\in\bigcap_{j\in J}\Sigma_j"] \arrow[ur, "f"'] & \end{tikzcd}$$

. The same proof holds for pull-backs.

Lemma 6 *If $(\Sigma_j)_{j\in J}$ is a non empty family of WE subcategories of a small category $\mathcal{C}$ then $\biguplus_{j\in J}\Sigma_j$ is a WE-subcategory of $\mathcal{C}$. Where $\biguplus_{j\in J}\Sigma_j$ is the least sub-category of $\mathcal{C}$ including all the Σ_j's.*

Proof. By definition, $\biguplus_{j\in J}\Sigma_j = \{\sigma_n\circ...\circ\sigma_1/for\ n\in\mathbb{N}^*\ \{j_1,...,j_n\}\subseteq J\ and\ \forall k\in\{1,...,n\}\sigma_k\in\Sigma_{j_k}\}$, 1^{st} point of definition 7 immediately follows. The 2^{nd} one is obvious for the family is non empty and the 4^{th} one because a composition of Yoneda inversible morphisms is Yoneda inversible (see remark 1). Take $\sigma_n\circ...\circ\sigma_1 \in \biguplus_{j\in J}\Sigma_j$ with $n\in\mathbb{N}^*$, $\{j_1,...,j_n\}\subseteq J$, $\forall k\in\{1,...,n\}\sigma_k\in\Sigma_{j_k}$ and $f\in Mo(\mathcal{C})$ with $src(\sigma_1) = src(f)$. We have

$$\begin{tikzcd} {} & & & \\ {} \arrow[u, "f"] \arrow[r, "\sigma_1\in\Sigma_{j_1}"'] & {} \arrow[r, dotted, no head] & {} \arrow[r, "\sigma_n\in\Sigma_{j_n}"'] & {} \end{tikzcd}$$

. With a finite induction (apply consecutively the 3^{rd} point of definition 7 for $\Sigma_{j_1},...,\Sigma_{j_n}$), we have

$$\begin{tikzcd} {} \arrow[r, dashed, "\sigma'_1\in\Sigma_{j_1}"] & {} & {} \arrow[r, dashed, "\sigma'_n\in\Sigma_{j_n}"] & {} \\ {} \arrow[u, "f"] \arrow[r, "\sigma_1\in\Sigma_{j_1}"'] \arrow[ur, phantom, "\text{pushout}"] & {} \arrow[u, dashed, "f_1"'] \arrow[r, dotted, no head] & {} \arrow[u, dashed, "f_{n-1}"] \arrow[r, "\sigma_n\in\Sigma_{j_n}"'] \arrow[ur, phantom, "\text{pushout}"] & {} \arrow[u, dashed, "f_n"'] \end{tikzcd}$$

. Now, it is a general fact that a "composition" of push-out squares is a push-out square (see [Bor94a]) hence

$$\begin{tikzcd} {} \arrow[r, dashed, "\sigma'_n\circ...\circ\sigma'_1\in\biguplus_{j\in J}\Sigma_j"] & {} \\ {} \arrow[u, "f"] \arrow[r, "\sigma_n\circ...\circ\sigma_1\in\biguplus_{j\in J}\Sigma_j"'] \arrow[ur, phantom, "\text{pushout}"] & {} \arrow[u, dashed, "f_n"'] \end{tikzcd}$$

. It works analoguously for pull-backs, thus the 3^{rd} point of definition 7 is satisfied.

Lemma 7 *Let $\mathcal{C}$ be a (small) category. If $\mathcal{A}$ is a pure subcategory $\mathcal{C}$ then for all families $(\mathcal{C}_j)_{j\in J}$ of subcategories of $\mathcal{C}$, $\mathcal{A}\cap\left(\biguplus_{j\in J}\mathcal{C}_j\right) = \biguplus_{j\in J}(\mathcal{A}\cap\mathcal{C}_j)$*

Proof. The inclusion $\mathcal{A}\cap\left(\biguplus_{j\in J}\mathcal{C}_j\right) \supseteq \biguplus_{j\in J}(\mathcal{A}\cap\mathcal{C}_j)$ is always satisfied. Indeed, if f is an element of the right member, then one has $n\in\mathbb{N}^*$, $\{j_1,...,j_n\}\subseteq J$, $\forall k\in\{1,...,n\}\sigma_k\in\mathcal{A}\cap\Sigma_{j_k}$ and $f = \sigma_n\circ...\circ\sigma_1$. Now $\mathcal{A}$ is a subcategory of $\mathcal{C}$ and in particular $\forall k\in\{1,...,n\}\sigma_k\in\mathcal{A}$, hence $f\in Mo(\mathcal{A})$. Conversely, suppose that we have $n\in\mathbb{N}^*$, $\{j_1,...,j_n\}\subseteq J$, $\forall k\in\{1,...,n\}\sigma_k\in\Sigma_{j_k}$ and $f = \sigma_n\circ...\circ\sigma_1\in Mo(\mathcal{A})$, by pureness of $\mathcal{A}$, $\sigma_n,...,\sigma_1\in Mo(\mathcal{A})$. Then $\forall k\in\{1,...,n\}\sigma_k\in\mathcal{A}\cap\Sigma_{j_k}$ and f is an element of the left member.

Remark 2 *If $\mathcal{C}$ satisfies the following property: $\forall \gamma_1, \gamma_2 \in Mo(\mathcal{C}), \gamma_2 \circ \gamma_1 = \gamma_2 \Rightarrow \gamma_1 = id$ and $\gamma_2 \circ \gamma_1 = \gamma_1 \Rightarrow \gamma_2 = id$, then the converse of lemma 7 is true.*

PROOF. Take $\gamma_2 \circ \gamma_1 \in Mo(\mathcal{A})$ where $\gamma_2, \gamma_1 \in Mo(\mathcal{C})$. Set $\mathcal{C}_1 := \{\gamma_1\}$ and $\mathcal{C}_2 := \{\gamma_2\}$ and apply the distributivity for the family $\{\mathcal{C}_1, \mathcal{C}_2\}$. If $\gamma_1 \notin Mo(\mathcal{A})$ and $\gamma_2 \notin Mo(\mathcal{A})$ then $(\mathcal{A} \cap \mathcal{C}_1) \uplus (\mathcal{A} \cap \mathcal{C}_2) = \emptyset$ while $\mathcal{A} \cap (\mathcal{C}_1 \uplus \mathcal{C}_2) = \{\gamma_2 \circ \gamma_1\}$. If $\gamma_1 \notin Mo(\mathcal{A})$ and $\gamma_2 \in Mo(\mathcal{A})$ then $(\mathcal{A} \cap \mathcal{C}_1) \uplus (\mathcal{A} \cap \mathcal{C}_2) = \{\gamma_2\}$ while $\mathcal{A} \cap (\mathcal{C}_1 \uplus \mathcal{C}_2) = \{\gamma_2, \gamma_2 \circ \gamma_1\}$ and $\gamma_2 \neq \gamma_2 \circ \gamma_1$ by the property of $\mathcal{C}$, precisely, if we had $\gamma_2 = \gamma_2 \circ \gamma_1$, we would have $\gamma_1 = id_{src(\gamma_1)}$ hence $id_{src(\gamma_1)} \in Mo(\mathcal{A})$ because $\mathcal{A}$ is a subcategory of $\mathcal{C}$.
The required property is true if $\mathcal{C}$ is a groupoid or a loop-free category. In fact, having $\mathcal{A} \cap \left(\biguplus_{j \in J} \mathcal{C}_j\right) = \biguplus_{j \in J} (\mathcal{A} \cap \mathcal{C}_j)$ is equivalent to the existence of the right adjoint of the functor $\mathcal{A} \cap \_$: ({subcategories of $\mathcal{C}$}, $\subseteq$) $\longrightarrow$ ({subcategories of $\mathcal{C}$}, $\subseteq$), where the continuous lattice ({subcategories of $\mathcal{C}$}, $\subseteq$) is seen as a complete and co-complete small category. The equivalence directly comes from the special adjoint functor theorem . This equivalence is related to the link between locales and complete **Heyting algebras**, see [Bor94b] for further details.

Corollary 2 *Let $(\Sigma_j)_{j \in J}$ be a family of WE-subcategories of a (small) category $\mathcal{C}$ such that $Iso(\mathcal{C})$ is pure in $\mathcal{C}$ and Σ a WE-subcategory of $\mathcal{C}$. Then $\Sigma \cap \left(\biguplus_{j \in J} \Sigma_j\right) = \biguplus_{j \in J} (\Sigma \cap \Sigma_j)$.*

PROOF. By lemma 3, Σ is pure in $\mathcal{C}$, the result follows by lemma 7. Note that the hypothesis that all the Σ_j's are WE-subcategories is not used in the proof.

Remark 3 *$\bigcap$ and $\biguplus$ are associative over the family of subcategories of a small category $\mathcal{C}$.*

Theorem 2 *Let $\mathcal{C}$ be a small category such that $Iso(\mathcal{C})$ is pure in $\mathcal{C}$ (i.e. $\mathcal{C}$ is directed). Then, the family of WE-subcategories of $\mathcal{C}$ is not empty and, together with $\subseteq$ it forms a locale whose l.u.b. operator is $\biguplus$ and g.l.b operator is $\bigcap$. Moreover, the least element of this locale ("bottom") is $Iso(\mathcal{C})$.*

PROOF. Axioms of a locale are given by lemmas 5, 6 and corollary 2.
As it is explained in [Bor94b] and [Joh82], notion of locale generalizes notion of family of open subsets of a topological space, thus, theorem 2 gives us a kind of topology over $\mathcal{C}$ as soon as $Iso(\mathcal{C})$ is pure in $\mathcal{C}$. This pureness hypothesis is actually very "natural". Ideologicaly, if we want to consider an isomorphism of $\mathcal{C}$ as a path that can be run forward, which is the case when $\mathcal{C}$ is a fundamental category, it geometricaly makes sense to expect that all its subpaths can also be run forward i.e. are isomorphisms. When this "geometrical" assuption is fulfilled by a small category $\mathcal{C}$, roughly speaking, $\mathcal{C}$ describes the arc-wise connectedness of a "geometrical shape".

4.3 Quotient of a small category by one of its subcategory : $\mathcal{C}/_\Sigma$

Given Σ a subcategory of a small category $\mathcal{C}$, we can define $\mathcal{C}/_\Sigma := \mathcal{C}/_\sim$ where $\sim$ is the least generalized congruence on $\mathcal{C}$ containing $(\emptyset, \{(id_{tgt(\sigma)}, \sigma), (\sigma, src_{src(\sigma)})/\sigma \in Mo(\Sigma)\})$ (by lemma 1).

Theorem 3 (Description and universal property of $\mathcal{C}/\Sigma$)
Given a small category $\mathcal{C}$ and $\Sigma \subseteq Mo(\mathcal{C})$, closed under composition (in fact, take Σ a subcategory of $\mathcal{C}$). Let $(\sim_{o,\Sigma}, \sim_{m,\Sigma})$ be the least generalized congruence containing $(\emptyset, \{(id_{tgt(\sigma)}, \sigma), (\sigma, id_{src(\sigma)})/\sigma \in \Sigma\})$. Then $\forall x, y \in Ob(\mathcal{C}), x \sim_{o,\Sigma}$ iff there is a Σ-zig-zag between x and y. $\forall (\beta_n, ..., \beta_0), (\alpha_m, ..., \alpha_0)$ $\sim_{o,\Sigma}$-composable sequences (i.e. $src(\alpha_{i+1}) \sim_{o,\Sigma} tgt(\alpha_i)$ and $src(\alpha_{i+1}) \sim_{o,\Sigma} tgt(\alpha_i)$), we have $(\beta_n, ..., \beta_0) \sim_{m,\Sigma} (\alpha_m, ..., \alpha_0)$ iff there is a finite sequence of "elementary transformation" from $(\alpha_m, ..., \alpha_0)$ to $(\beta_n, ..., \beta_0)$, where an "elementary transformation" is either

- *$(\alpha_n, ..., \alpha_{i+1}, \sigma, \alpha_{i-1}, ..., \alpha_0) \sim^1_{m,\Sigma} (\alpha_n, ..., \alpha_{i+1}, id_{src(\sigma)}$ or $id_{tgt(\sigma)}, \alpha_{i-1}, ..., \alpha_0)$ if $\sigma \in \Sigma$*

  or

- *$(\alpha_n, ..., \alpha_{i+2}, \alpha_{i+1}, \alpha_i, \alpha_{i-1}, ..., \alpha_0) \sim^1_{m,\Sigma} (\alpha_n, ..., \alpha_{i+2}, \alpha_{i+1} \circ \alpha_i, \alpha_{i-1}, ..., \alpha_0)$ if $src(\alpha_{i+1}) = tgt(\alpha_i)$.*

$\mathcal{C}/\Sigma$ is caracterized by the following universal property, $\forall f \in \mathbf{CAT}[\mathcal{C},\mathcal{C}']$, if $\forall \sigma \in \Sigma, f(\sigma) = id$ then $\exists! g \in \mathbf{CAT}[\mathcal{C},\mathcal{C}/\Sigma]$ such that

$$\begin{tikzcd} & \mathcal{C}/\Sigma \arrow[dr, "g"] & \\ \mathcal{C} \arrow[ur, "Q_\Sigma"] \arrow[rr, "f"'] \arrow[rr, phantom, bend left=40, "commutes"] & & \mathcal{C}' \end{tikzcd}$$

Moreover, if $\mathcal{C}_1 \xrightarrow{f} \mathcal{C}_2$ satisfies $f(\Sigma_1) \subseteq \Sigma_2$ then $\exists!\ \mathcal{C}_1/\Sigma_1 \xrightarrow{h} \mathcal{C}_2/\Sigma_2$ making the following diagram commutes

$$\begin{tikzcd} \mathcal{C}_1 \arrow[r, "f"] \arrow[d, "Q_{\Sigma_1}"'] \arrow[dr, phantom, "="] & \mathcal{C}_2 \arrow[d, "Q_{\Sigma_2}"] \\ \mathcal{C}_1/\Sigma_1 \arrow[r, "h"'] & \mathcal{C}_2/\Sigma_2 \end{tikzcd}$$

Where Q_Σ is the quotient functor (refer to theorem 1) associated to the generalized congruence induced by Σ. g is also denoted $f_{/\Sigma_1,\Sigma_2}$, and in the same stream of notation h is denoted $f_{/\Sigma}$.

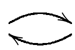
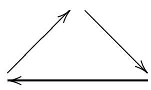
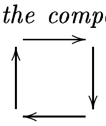
Definition of the component category

The **component category of a directed category** $\mathcal{C}$ is defined as $\mathcal{C}/\top_{WE(\mathcal{C})}$ where $\top_{WE(\mathcal{C})}$ is the biggest weak equivalence subcategory of $\mathcal{C}$. Given a pospace/local pospace/directed space X, the component category of X is defined as the component category of $\pi_1(X)$, the fundamental category of X.

It makes sense by theorem 2 and 1. Remark we have not the functoriality yet. Next theorem establishes a relation between connectedness [16] and component category of the fundamental groupoid of a topological space.

Theorem 4 *Let $\mathcal{G}$ be a groupoid, then $Mo(\mathcal{G})$ is the $\subseteq$-biggest WE-subcategory of $\mathcal{G}$. Moreover $\mathcal{G}/_{Mo(\mathcal{G})}$ is (isomorphic to) the set (prcisely a disrcete category seen as its set of objects) of zigzag connected component of $\mathcal{G}$. If $\mathcal{G} := \Gamma_1(X,\tau_X)$ the fundamental groupoid of topological space (X,τ_X), then $\mathcal{G}/_{Mo(\mathcal{G})}$ is the set of arc-wise connected components of (X,τ_X).*

Proof. Any isomorphism is both left and right retractions, thus, by 1^{st} property of WE-subcategories, if $\mathcal{G}$ has a WE-subcategory, it is necessarily $Mo(\mathcal{G})$ which is stable under composition. By remark 1, each morphism of a groupoid is Yoneda inversible hence 2^{nd} and 3^{rd} point of definition 7 are satisfied. Finally, it is a general fact that if σ is an isomorphism, then any morphism f such that $src(f) = src(\sigma)$ has a push-out along σ and any morphism g such that $tgt(g) = tgt(\sigma)$ has a pull-back along σ, thus we have the 3^{rd} point of definition 7.
Then each morphism of $\mathcal{G}$ is identified with the identity of its source and target. Two objects x, y of $\mathcal{G}$ are identified iff there is a zigzag between them (note that, since $\mathcal{G}$ is a groupoid, it is equivalent to $\mathcal{G}[x,y] \neq \emptyset$).

Remark 4 *Any free category is obviously a one-way category, so we can always define the component category of a free category. For example, the component category of the monoid $(\mathbb{N},+)$ seen as a small category is $(\mathbb{N},+)$. More generally, $(\mathbb{N},+)$ is also the component category of the free categories generated by the following graphs*  ,  ,  *... Compared to these examples, $\vec{S}^1$ can be seen as a continuous generalization[17].*

[16] in the classical algebraic topology sense.

[17] Still, note that the fundamental category of $\pi_1(\vec{S}^1)$ is not free as described in section 2.

Theorem 5 *Let $\mathcal{C}$ be a small category and Σ a wide subcategory of $\mathcal{C}$.*
If *$\mathcal{C}$ is loop-free and Σ is a pure subcategory of Yoneda inversible morphisms admitting left and right extension properties* **then** *$\mathcal{C}/\Sigma$ is loop-free.*
If *Σ is pure in $\mathcal{C}$* **then** *$\mathcal{C}/\Sigma$ is one-way.*

Theorem 6 *For any small category $\mathcal{C}$, $\mathcal{C}/_{\Sigma_{loop}}$ is loop-free. Where Σ_{loop} is the subcategory of $\mathcal{C}$ generated by morphisms σ such that $\exists \alpha, \beta\gamma \in Mo(\mathcal{C})$,*

$$\begin{tikzcd} \cdot \arrow[dr, "\alpha"'] & & & \cdot \arrow[lll, bend right, "\gamma"'] \\ & \cdot \arrow[r, "\sigma"'] & \cdot \arrow[ur, "\beta"'] & \end{tikzcd}$$

, <u>it is not necessarily commutative.</u>

Note that Σ_{loop} is a pure subcategory of $\mathcal{C}$

Definition 8 *A category is* **thin** *iff its biggest weak equivalences subcategory is discrete.* **TLFCAT**, **TOWCAT** *and* **TdCAT** *are the full sub categories of thin loop-free categories of* **LFCAT**, **OWCAT** *and* **dCAT**.

Conjecture 2 *Let $\mathcal{L}$ be a small loop-free category and $\Sigma_{\mathcal{L}}$ the biggest WE-subcategory of $\mathcal{L}$. Then $\mathcal{L}/\Sigma_{\mathcal{L}}$ is thin (see definition 8).*

4.4 Functoriality of component categories

Next theorem gives the general framework in which component category can be defined as a functor. As pointed out in the abstract, the idea is to equip any small category $\mathcal{C}$ of our scope of interest with a subcategory of distinguished morphisms (called "inessential" in [FGHR04]) which are unformally those along with "nothing happens".

Theorem 7 (General framework for component category functor)
Let $\mathbf{K}$ be a subcategory of $\mathbf{CAT}$ and Φ be an "assignment" which gives to each $\mathcal{C}$ object of $\mathbf{K}$ a subposet of $(Sb(\mathcal{C}), \subseteq)$ (which is the complete partial order of subcategories of $\mathcal{C}$) with "top" and "bottom" elements. Then we define $\mathbf{K}\Phi$ the categorgy whose objects are pairs $(\mathcal{C}, \Sigma)$ where $\mathcal{C}$ is an object of $\mathbf{K}$ and $\Sigma \in \Phi(\mathcal{C})$ and $\mathbf{K}\Phi[(\mathcal{C}_1, \Sigma_1), (\mathcal{C}_2, \Sigma_2)] := \{f \in \mathbf{K}[\mathcal{C}_1, \mathcal{C}_2] / \forall \sigma \in \Sigma_1,\ f(\sigma) \neq id \Rightarrow f(\sigma) \in \Sigma_2\}$.

(i) $\forall f \in \mathbf{K}[\mathcal{C}_1, \mathcal{C}_2] \forall \sigma \in \top_{\Phi(\mathcal{C}_1)},\ f(\sigma) \neq id \Rightarrow f(\sigma) \in \top_{\Phi(\mathcal{C}_2)}$

(ii) $\forall f \in \mathbf{K}[\mathcal{C}_1, \mathcal{C}_2] \forall \sigma \in \bot_{\Phi(\mathcal{C}_1)},\ f(\sigma) \neq id \Rightarrow f(\sigma) \in \bot_{\Phi(\mathcal{C}_2)}$

(iii) *For all $\mathcal{C}$ object of $\mathbf{K}$, $\forall \Sigma \in \Phi(\mathcal{C})$*

- *(a)* $\bot_{\Phi(\mathcal{C})} \subseteq id(\mathcal{C})$
- *(b)* *$Q_\Sigma : \mathcal{C} \longrightarrow \mathcal{C}/\Sigma$ is a morphism of $\mathbf{K}$ (hence $\mathcal{C}/\Sigma$ is an object of $\mathbf{K}$)*
- *(c)* *$\forall f \in \mathbf{K}\Phi[(\mathcal{C}, \Sigma), (\mathcal{C}', \Sigma')]$, $f/_\Sigma : \mathcal{C}/\Sigma \longrightarrow \mathcal{C}'$ and $f/_{\Sigma,\Sigma'} : \mathcal{C}/\Sigma \longrightarrow \mathcal{C}'/\Sigma'$ [18] are morphisms of $\mathbf{K}$*

Then we have

- *(iiia) $\Rightarrow$ (ii)*
- *If (i) is satisfied then R is well defined and $U \dashv R$*
- *If (ii) is satisfied then L is well defined and $L \dashv U$*
- *If (iii) is satisfied then $Comp\Phi$ is well defined and $Comp\Phi \dashv L$*

[18] see lemma 3 for notations $f/_\Sigma$ and $f/_{\Sigma,\Sigma'}$.

Where

$$\begin{tikzcd}[column sep=huge]
\mathbf{K}\Phi \arrow[r, bend left, "U"] \arrow[r, bend right=60, "Comp\Phi"'] & \mathbf{K} \arrow[l, dashed, bend right=70, "R"'] \arrow[l, dashed, "L"]
\end{tikzcd}$$

U is the obvious forgetful functor.
Given $\mathcal{C} \in \mathbf{K}$, $L(\mathcal{C}) := (\mathcal{C}, \perp_{\Phi(\mathcal{C})})$, $R(\mathcal{C}) := (\mathcal{C}, \top_{\Phi(\mathcal{C})})$, $Comp\Phi(\mathcal{C}, \Sigma) := \mathcal{C}/_{\Sigma}$.
Given $f : \mathcal{C}_1 \longrightarrow \mathcal{C}_2$, $R(f)$ is the induced morphism from $(\mathcal{C}_1, \top_{\Phi(\mathcal{C}_1)})$ to $(\mathcal{C}_2, \top_{\Phi(\mathcal{C}_2)})$ (i.e. $U(R(f)) = f$)
$L(f)$ is the induced morphism from $(\mathcal{C}_1, \perp_{\Phi(\mathcal{C}_1)})$ to $(\mathcal{C}_2, \perp_{\Phi(\mathcal{C}_2)})$ (i.e. $U(L(f)) = f$) and
for all $f \in \mathbf{K}\Phi[(\mathcal{C}_1\Sigma_1), (\mathcal{C}_2, \Sigma_2)]$, $Comp\Phi(f) := f_{\Sigma_1, \Sigma_2}$.

PROOF.
(iiia) $\Rightarrow$ *(ii)*:
Take $f \in \mathbf{K}[\mathcal{C}_1, \mathcal{C}_2]$ and $\sigma \in \perp_{\Phi(\mathcal{C}_1)}$ by (ii), σ is an identity so necessarily $f(\sigma)$ is an identity.
$(i) \Rightarrow U \dashv R$:
R is well defined because the object part does not raise any problem and (i) is exactly the assumption we need to ensure that morphism of $\mathbf{K}$ from $\mathcal{C}_1$ to $\mathcal{C}_2$ induces a morphism of $\mathbf{K}\Phi$ from $(\mathcal{C}_1, \top_{\Phi(\mathcal{C}_1)})$ to $(\mathcal{C}_2, \top_{\Phi(\mathcal{C}_2)})$. The unit of the ajunction is $\eta_{(\mathcal{C},\Sigma)} : (\mathcal{C}, \Sigma) \longrightarrow (\mathcal{C}, \top_{\Phi(\mathcal{C})})$, which is a morphism of $\mathbf{K}\Phi$ since $\Sigma \subseteq \top_{\Phi(\mathcal{C})}$. The co-unit is $\varepsilon_{\mathcal{C}} := id_{\mathcal{C}}$. Given $f : (\mathcal{C}, \Sigma) \longrightarrow (\mathcal{C}', \Sigma')$, put $g := U(f)$, it is clearly the only morphism of $\mathbf{K}$ such that $f = g \circ id_{\mathcal{C}}$ and $f = R(g) \circ \eta_{(\mathcal{C},\Sigma)}$. The naturality of η is obvious.
$(ii) \Rightarrow L \dashv U$:
L is well defined because the object part does not raise any problem and (ii) is exactly the assumption we need to ensure that morphism of $\mathbf{K}$ from $\mathcal{C}_1$ to $\mathcal{C}_2$ induces a morphism of $\mathbf{K}\Phi$ from $(\mathcal{C}_1, \perp_{\Phi(\mathcal{C}_1)})$ to $(\mathcal{C}_2, \perp_{\Phi(\mathcal{C}_2)})$. The unit of the ajunction is $\eta_{\mathcal{C}} = id_{\mathcal{C}}$ the co-unit is $\varepsilon_{(\mathcal{C},\Sigma)} : (\mathcal{C}, \perp_{\Phi(\mathcal{C})}) \longrightarrow (\mathcal{C}, \Sigma)$ which is a morphism of $\mathbf{K}\Phi$ because $\perp_{\Phi(\mathcal{C})} \subseteq \Sigma$. Given a morphism $f \in \mathbf{K}[\mathcal{C}_1, \mathcal{C}_2]$, setting $g := \varepsilon_{(\mathcal{C},\Sigma)} \circ L(f)$, we have $f = \eta_{\mathcal{C}} \circ L(g)$ i.e. $f = L(g)$.
$(iii) \Rightarrow CC\Phi \dashv L$:
The object part of $CC\Phi$ is well defined by $(iiib)$, the morphism part of $CC\Phi$ is well defined by $(iiic)$ ($f/_{\Sigma,\Sigma'} : \mathcal{C}/\Sigma \longrightarrow \mathcal{C}'/\Sigma'$ is a morphism of $\mathbf{K}$). The unit of the adjunction is the only morphism $\eta_{(\mathcal{C},\Sigma)}$: $(\mathcal{C}, \Sigma) \longrightarrow (\mathcal{C}/\Sigma, \perp_{\Phi(\mathcal{C}/\Sigma)})$ such that $U(\eta_{(\mathcal{C},\Sigma)}) = Q_{\Sigma}$, Q_{Σ} is in $\mathbf{K}$ by $(iiib)$, moreover $\forall \sigma \in \Sigma$, $Q_{\Sigma}(\sigma)$ is an identity, hence by definition of $\mathbf{K}\Phi$, $\eta_{(\mathcal{C},\Sigma)}$ is in $\mathbf{K}\Phi$. Let $f : (\mathcal{C}, \Sigma) \longrightarrow (\mathcal{C}', \perp_{\Phi(\mathcal{C}')})$ morphism of $\mathbf{K}\Phi$, it follows that $\forall \sigma \in \Sigma$, $f(\sigma) \neq id \Rightarrow f(\sigma) \in \perp_{\Phi(\mathcal{C}')}$, however, by $(iiia)$, $\perp_{\Phi(\mathcal{C}')} \subseteq id(\mathcal{C}')$ then $\forall \sigma \in \Sigma$, $f(\sigma)$ is an identity. So we can apply lemma 3, $f/_{\Sigma}$ is the only morphism of small categories from $\mathcal{C}/\Sigma$ to $\mathcal{C}'$ such that $U(f) = f/_{\Sigma} \circ Q_{\Sigma}$. It follows that $f/_{\Sigma}$ is the only morphism of $\mathbf{K}$ (cf $(iiic)$) such that $f = L(f/_{\Sigma}) \circ \eta_{(\mathcal{C},\Sigma)}$. Naturality of $\eta_{(\mathcal{C},\Sigma)}$ is a consequence of uniqueness property of lemma 3.

Definition of the component category functor by means of theorem 7

It suffices to set $\mathbf{K}$:=**LFCAT** and $\Phi(\mathcal{C}) := WE(\mathcal{C})$, (ii) and $(iiia)$ are satisfied because $\perp_{\Phi(\mathcal{C})} := \{id_x / x \in Ob(\mathcal{C})\}$. By theorem 5, $\forall \Sigma \in WP(\mathcal{C})$, $\mathcal{C}/\Sigma$ is a loop-free category, since **LFCAT** is a full sub-category of **CAT**, $(iiib)$ and $(iiic)$ are also satisfied. Note that (i) is not necessarily satisfied, hence we do not have, in general, the functor R.

We can do the same setting $\mathbf{K}$:=**OWCAT** and $\Phi(\mathcal{C}) := WE(\mathcal{C})$, (ii) and $(iiia)$ are satisfied because $\perp_{\Phi(\mathcal{C})} := \{id_x / x \in Ob(\mathcal{C})\}$. By theorem 5, $\forall \Sigma \in WP(\mathcal{C})$, $\mathcal{C}/\Sigma$ is a one-way category, since **OWCAT** is a full sub-category of **CAT**, $(iiib)$ and $(iiic)$ are also satisfied. Once again, (i) is not necessarily satisfied, hence we do not have, in general, the functor R.

For directed categories, things are slightly more intricate, the reason is that the least weak equivalences subcategory of a directed category $\mathcal{C}$ might contain isomorphisms which are not identities, hence ($iiia$) of theorem 7 is not necessarily satisfied. However, by theorem 1, **OWCAT** is a reflective subcategory of **dCAT** hence, if $L \dashv (\mathbf{dCAT} \hookrightarrow \mathbf{OWCAT})$ we define the compoent categroy functor as $Comp_{OW} \circ L$ where $Comp_{OW}$ is the component category functor defined in the case of one way categories. It is natural, isomorphisms are Yoneda inversible so they have to be turned into identities, the fact that we have to identify them before applying theorem 7 is just a technical twist which does not change the underlying philosophy of the method.

4.5 Comments and examples

4.5.1 Is there any relation with weak equivalences in model categories ?

In our context, morphisms of the weak eaquivalence subcategory of $\mathcal{C}$ are be called weak equivalences. However, these weak equivalences are far from model categories ones. There is a slight analogy between them, due to the pushout/pullback stability property but it does not really go further. In fact, the main difference is that, in model categories, the weak equivalences are (almost) always given by an intrisic property of the morphisms, for example in **SPC** the category of topological spaces, weak equivalences are continuous maps giving rise to isomorphisms bewteen homotopy groups in all dimensions. This definition just depends on the map and its domains and codomains, in some sense, it is local. On the other hand, weak equivalences in our context are defined as part of a subcategory which is defined in a global way. Let us consider $\vec{T} := \{(x,y)/0 \leq x,y; x+y \leq 1\}$ and $\vec{C} := \{(x,y)/0 \leq x,y \leq 1\}$ with classical topology and order. It is easy to check that in $\pi_1(\vec{T})$ as well as in $\pi_1(\vec{C})$, all morphisms are Yoneda inversible. $\pi_1(\vec{C})$ clearly has all pushouts and pullbacks hence any morphism of $\pi_1(\vec{C})$ is a weak equivalence while the only weak equivalences of $\pi_1(\vec{T})$ are identities. The reason is that for any non identical morphism σ of $\pi_1(\vec{T})$ one can find a morphism γ so that the (right) extension property is not fulfilled. The last example emphasizes the global and geometric aspect of our weak equivalence definition.

4.5.2 Detailed calculation of the component category of the "L" pospace

The idea is to find morphisms that are "obviously" not weak equivalences and to check the remaining one form a weak equivalence subcategory. Let L be the pospace depicted in figure 6 with classical topology and order. Given $(x,y) \leq (x',y')$ there is, up to dihomotopy, a unique morphism from (x,y) to (x',y'), hence any morphism is Yoneda inversible. Now suppose that a morphism σ crosses the vertical dotted segment, then take γ a morphism which crosses the horizontal one. Clearly, the right extension property is not satisfied by σ. Now it is easy to check that the subcategory made of the morphisms of $\pi_1(L)$ which do not cross any dotted segments are weak equivalences. By the way, note that if a morphism has its source or target exactly on the dotted line, it is still a weak equivalence. This is due to topological properties of components which have been deeper studied in [FGHR04].

4.5.3 Component category of the torus $\vec{T}^2$ with a hole

Take the directed square with a hole (see figure 1) then identity $[0,1] \times \{0\} \approx [0,1] \times \{1\}$ and $\{0\} \times [0,1] \approx \{1\} \times [0,1]$. We obtain a local pospace whose underlying topological space is a torus with a hole and where the local order is clockwise on the "small" and "large" generators, denote T this local pospace. Figure 7 represents T with the identifications described above. Then it comes that we have three components[19].

The component category is the free category generated by
$$\begin{tikzcd} B \arrow[r, bend right, "g_B"'] & A \arrow[l, bend right, "f_B"'] \arrow[r, bend left, "f_C"] & C \arrow[l, bend left, "g_C"] \end{tikzcd}$$

Indeed, we can prove that morphisms that crosses a dotted segment are not Yoneda inversible. Precisely,

[19] which are connected up to identification.

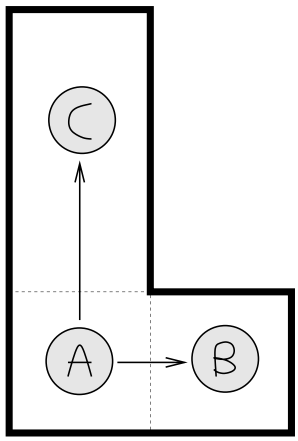


Figure 6: The “L” pospace

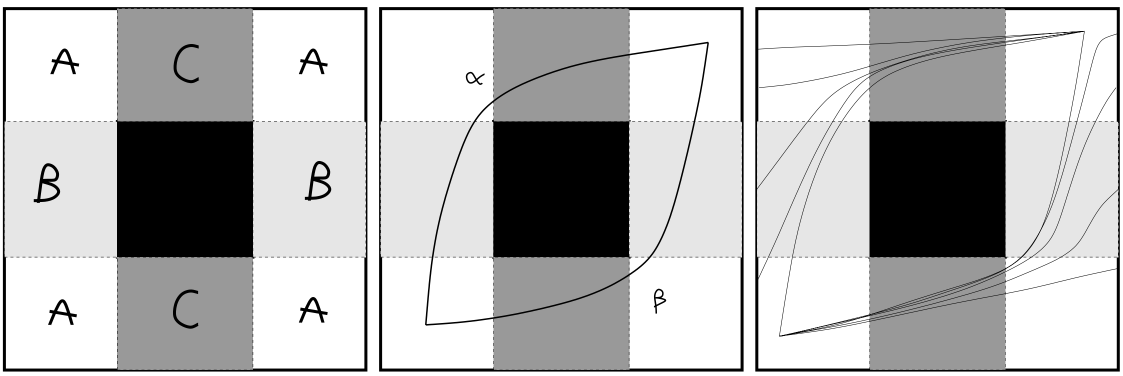


Figure 7: Directed torus $\overrightarrow{T}^2$ -unfold representation-

15

the morphisms of the fundamental category they induce are both monic and epic making the correponding set theoretic maps are one-to-one but not onto (see definition 5). The dipaths α and β on figure 7 are not dihomotopic, the right hand part shows the "only" dihomotopy one could image. In fact, α and β are not even homotopic, it is a classical algebraic topology problem.

5 Tool for calculation of component categories

The presentation given above could let the reader think that theorem 7 is useless to define component categories, and forgeting the functoriality question, he is right! The point is that, in concrete case, we want to be able to calculate component categories and, in order to do so, we need efficient tools. Ones of the most classical results towards calculation of fundamental groups, groupoids and categories are Van Kampen theorems[20]. The idea of the theorem is as follows, given a geometrical shape X (classical or directed), instead of directly calculating the fundamental object of X, split X into two parts, say A and B whose fundamental objects are known (or at least easier to calculate) then "glue" the fundamental objects of A and B to have the fundamental object of X. If you see a geometrical shape as a program and its fundamental object as an abstract interpretation (see [CC92]) of this program, then Van Kampen theorem becomes a kind of "compositionality" result. Technical details of Van Kampen theorem are of out of the scope of this paper, so we just give an unformal statement.
In theorems 8 and 9, $\star$**SPC** are $\star$**CAT** are taken by pair according to the following table
Table 5

$\star$**SPC**	$\star$**CAT**
POSPC	**LFCAT**
LPOSPC	**OWCAT**
dSPC	**dCAT**

where $\star$**SPC** is the domain of the fundamental category functor π_1 and $\star$**CAT** its codomain.

Theorem 8 (Van Kampen for fundamental category) *Let $\vec{X}_1, \vec{X}_2$ be sub-objects of $\vec{X}$ (object of $\star$**SPC**) such that the underlying topological space of $\vec{X}$ is the union of the interiors[21] of the underlying topological spaces of $\vec{X}_1$ and $\vec{X}_2$ and let $\vec{X}_0 := \vec{X}_1 \cap \vec{X}_2$. $i_1 : \vec{X}_0 \hookrightarrow \vec{X}_1, i_2 : \vec{X}_0 \hookrightarrow \vec{X}_2, j_1 : \vec{X}_1 \hookrightarrow \vec{X}$ and $j_2 : \vec{X}_2 \hookrightarrow \vec{X}$ the inclusion maps. Then we have the following push-out squares*

$$\begin{tikzcd} & \vec{X} & \\ \vec{X}_1 \arrow[ur, "j_1"] & push-out & \vec{X}_2 \arrow[ul, "j_2"'] \\ & \vec{X}_0 \arrow[ul, "i_1"] \arrow[ur, "i_2"'] & \end{tikzcd} \qquad \begin{tikzcd} & \pi_1(\vec{X}) & \\ \pi_1(\vec{X}_1) \arrow[ur, "\pi_1(j_1)"] & push-out & \pi_1(\vec{X}_2) \arrow[ul, "\pi_1(j_2)"'] \\ & \pi_1(\vec{X}_0) \arrow[ul, "\pi_1(i_1)"] \arrow[ur, "\pi_1(i_2)"'] & \end{tikzcd}$$

*respectively in $\star$**SPC** and $\star$**CAT**.*

Theorem 9 (Van Kampen for component category)
*Let $\vec{X}_1$, $\vec{X}_2$ be sub-objects of $\vec{X}_3$ (object of $\star$**SPC**) such that the underlying topological space of $\vec{X}_3$ is the union of the interiors of the underlying topological spaces of $\vec{X}_1$ and $\vec{X}_2$ and let $\vec{X}_0 := \vec{X}_1 \cap \vec{X}_2$. $i_1 : \vec{X}_0 \hookrightarrow \vec{X}_1, i_2 : \vec{X}_0 \hookrightarrow \vec{X}_2, j_1 : \vec{X}_1 \hookrightarrow \vec{X}_3$ and $j_2 : \vec{X}_2 \hookrightarrow \vec{X}_3$ the inclusion maps.*
Moreover, we suppose that Σ_1, Σ_2 are WE-subcategories of $\pi_1(\vec{X}_1)$, $\pi_1(\vec{X}_2)$, $\pi_1(j_1)(\Sigma_1) \biguplus \pi_1(j_2)(\Sigma_2)$ (also

[20]there are several versions depending on the framework : see [Mas91] and [Swi02] for groups, [Hig71] for groupoids, [Gra03] for categories.

[21]with respect to the underlying topology of $\vec{X}$.

*denoted Σ_3) is a WE-subcategory of $\pi_1(\vec{X}_3)$, $\pi_1(i_1)(\Sigma_0) \subseteq (\Sigma_1)$ and $\pi_1(i_2)(\Sigma_0) \subseteq (\Sigma_2)$ (i.e. $\pi_1(i_1), \pi_1(i_2)$ are morphisms of $\star$**CATΦ**).*

then

*i_1, i_2, j_1 and j_2 give rise to i'_1, i'_2, j'_1 and j'_2 morphisms of $\star$**CATΦ** and we have*

$$\begin{tikzcd}
& (\pi_1(\vec{X}_3), \Sigma_3) & \\
(\pi_1(\vec{X}_1), \Sigma_1) \arrow[ur, "j'_1"] & \textit{push out in} \ \star CAT\Phi & (\pi_1(\vec{X}_2), \Sigma_2) \arrow[ul, "j'_2"'] \\
& (\pi_1(\vec{X}_0), \Sigma_0) \arrow[ul, "i'_1"] \arrow[ur, "i'_2"'] &
\end{tikzcd}$$

and

$$\begin{tikzcd}
& Comp\Phi(\pi_1(\vec{X}_3), \Sigma_3) & \\
Comp\Phi(\pi_1(\vec{X}_1), \Sigma_1) \arrow[ur, "Comp\Phi(j'_1)"] & \textit{push out in} \ \star CAT & Comp\Phi(\pi_1(\vec{X}_2), \Sigma_2) \arrow[ul, "Comp\Phi(j'_2)"'] \\
& Comp\Phi(\pi_1(\vec{X}_0), \Sigma_0) \arrow[ul, "Comp\Phi(i'_1)"] \arrow[ur, "Comp\Phi(i'_2)"'] &
\end{tikzcd}$$

The proof of theorem 9 requires three cases, one for each line of table 5. **POSPC/LFCAT** case can be found in [Gou95]. **dSPC/dCAT** is available in [Gra03]. In all the cases one might define the fundamental category of a local pospace as the fundamental category of its corresponding directed space see theorem 1.

PROOF. Theorem 8 gives us pushout squares in $\star$**SPC** and **?CAT**:

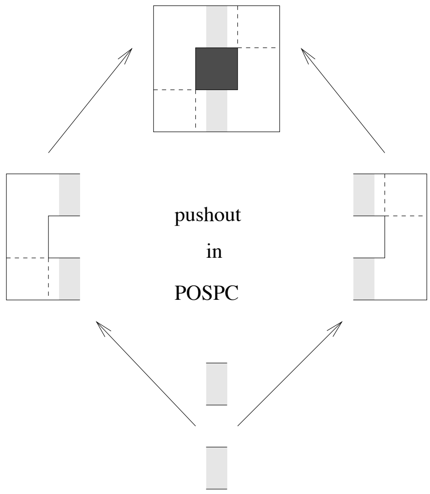
$$\begin{tikzcd}
& \vec{X} & \\
\vec{X}_1 \arrow[ur, "j_1"] & push\!-\!out & \vec{X}_2 \arrow[ul, "j_2"'] \\
& \vec{X}_0 \arrow[ul, "i_1"] \arrow[ur, "i_2"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& \pi_1(\vec{X}) & \\
\pi_1(\vec{X}_1) \arrow[ur, "\pi_1(j_1)"] & push\!-\!out & \pi_1(\vec{X}_2) \arrow[ul, "\pi_1(j_2)"'] \\
& \pi_1(\vec{X}_0) \arrow[ul, "\pi_1(i_1)"] \arrow[ur, "\pi_1(i_2)"'] &
\end{tikzcd}$$

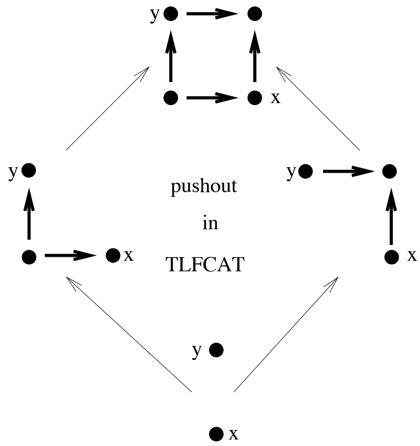
We have to prove that $\pi_1(\vec{X}_0)$, $\pi_1(\vec{X}_1)$, $\pi_1(\vec{X}_2)$ and $\pi_1(\vec{X}_3)$ respectively equiped with Σ_0, Σ_1, Σ_2 and Σ_3 give rise to a pushout square in $\star$**CATΦ**. Given $f_1 : (\pi_1(\vec{X}_1), \Sigma_1) \longrightarrow (\mathcal{L}, \Sigma)$ and $f_1 : (\pi_1(\vec{X}_2), \Sigma_2) \longrightarrow (\mathcal{L}, \Sigma)$ morphisms of $\star$**CATΦ** such that $f_1 \circ i_1 = f_2 \circ i_2$, by hypothesis, $\exists! h : \pi_1(\vec{X}_3) \longrightarrow \mathcal{L}$ (morphism of $\star$**CAT**) such that $f_1 = h \circ j_1$ and $f_2 = h \circ j_2$. It remains to see that h gives rise to a morphism of $\star$**CATΦ** i.e. $h(\Sigma_3) \subseteq \Sigma$. By hypothesis, $\Sigma_3 = j_1(\Sigma_1) \biguplus j_2(\Sigma_2)$ so any element of Σ_3 can be written $j_2(\alpha_{2n+1}) \cdot j_1(\alpha_{2n}) \cdot ... \cdot j_2(\alpha_1) \cdot j_1(\alpha_0)$ where $\forall k \in \{0, ..., n\}, \alpha_{2k} \in \Sigma_1$ and $\alpha_{2k+1} \in \Sigma_2$, so $h(j_2(\alpha_{2n+1}) \cdot j_1(\alpha_{2n}) \cdot ... \cdot j_2(\alpha_1) \cdot j_1(\alpha_0)) = (h \circ j_2)(\alpha_{2n+1}) \cdot (h \circ j_1)(\alpha_{2n}) \cdot ... \cdot (h \circ j_2)(\alpha_1) \cdot (h \circ j_1)(\alpha_0) = f_2(\alpha_{2n+1}) \cdot f_1(\alpha_{2n}) \cdot ... \cdot f_2(\alpha_1) \cdot f_1(\alpha_0) \in \Sigma$ since f_1, f_2 are morphisms of $\star$**CATΦ**, hence h gives rise to a morphism of $\star$**CATΦ** from $(\pi_1(\vec{X}_3), \Sigma_3)$ to $(\mathcal{L}, \Sigma)$. Thus we have a pushout square in $\star$**CATΦ**. Now by theorem 7, we know that $Comp\Phi$ is a left adjoint hence[22] preserves colimits and, in particular, pushouts.

Theorem 9 does not necessarily give the biggest WE-subcategory of $\pi_1(\vec{X}_3)$, so one has to guess what this biggest WE-subcategory is in order to choose appropriate Σ_1 and Σ_2, the choice of Σ_0 is not as important,

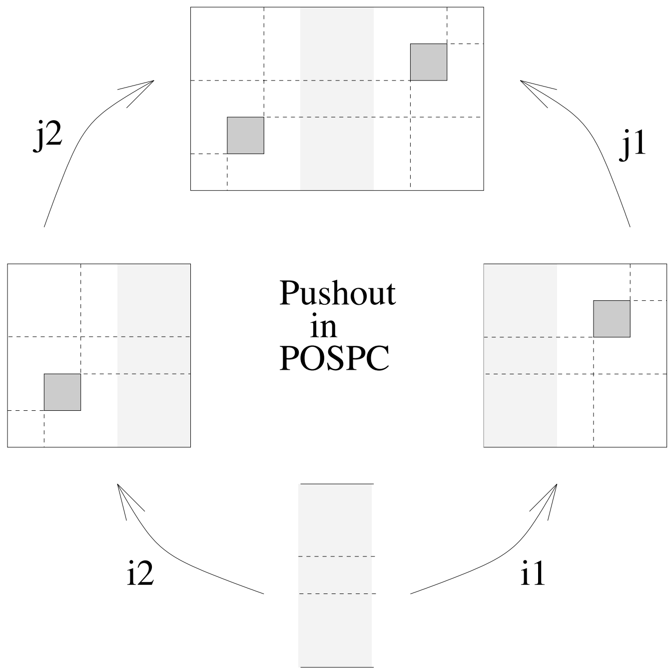
[22]general facts of category theory see [Bor94a].

and once Σ_1 and Σ_2 are given, it might be possible to take Σ_0 as the biggest WE-subcategory of $\pi_1(\overrightarrow{X}_0)$ satisfying $\pi_1(i_1)(\Sigma_0) \subseteq (\Sigma_1)$ and $\pi_1(i_2)(\Sigma_0) \subseteq (\Sigma_2)$. A very simple application of theorem 9 to calculate the component category of the first example given in section 2.

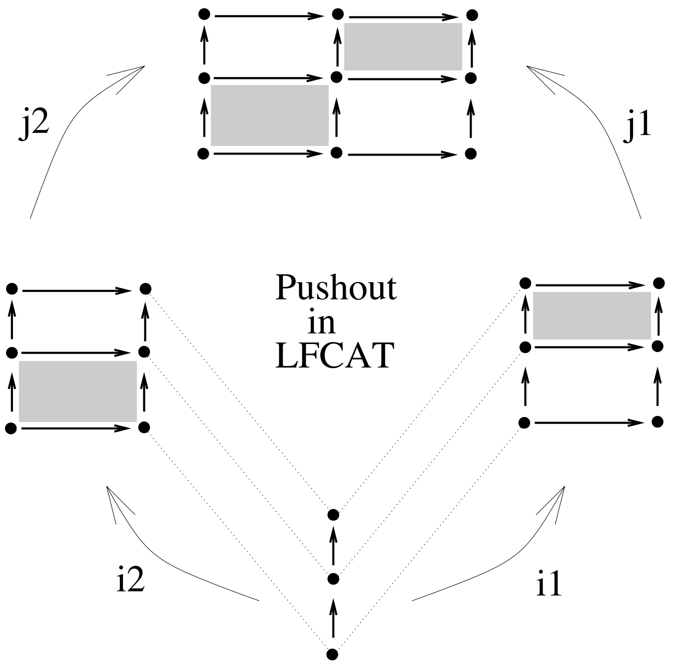




Let us come back to the example of the rectangle with two holes:



which gives, by theorem 9



In this figure, rectangle filled with grey color are <u>not</u> commutative. The holes of the geometrical shape are represented by non-commutative squares in the component category.

Applying theorem 9 we can also prove that the component category of the cube with a centered cubical hole has 26 objects[23]. It can be represented in $\mathbb{R}^3$ putting an object in the "center" of each vertex, edge and face ($8\ vertices + 12\ edges + 6\ faces = 26\ objects$). Morphisms are generated by arrows from a point to its "closer neighbours in the future", for example those of $(0,0,0)$ are $(0,0,\frac{1}{2})$, $(0,\frac{1}{2},0)$ and $(\frac{1}{2},0,0)$ while $(1,1,1)$ has no such neighbours. In order to have the hypothesis of theorem 9 satisfied, we split the cube into two parts so that, following notation of theorem 8, $X_0 :=]\frac{1}{2}-\epsilon, \frac{1}{2}+\epsilon[\times[0,1]\times[0,1]$. It is the analog of the previous example in three dimensions.

6 Towards directed cohomology

In [BW85] and [Bau91] a cohomology of small categories is presented by means of natural systems of factorization. The idea would be to define the cohomology of a directed geometrical object $\vec{X}$ as the cohomology of its fundamental category[24]. However, as we have already pointed it out, the fundamental category has often as many objects as $\mathbb{R}$. Still, there is only few of them which is relevant, and finding them amounts to calculate the component category. Thus, the cohomology of $\vec{X}$ could be defined as the cohomology of the component category of $\vec{X}$. For example, with this definition, the fourth and fifth examples given in section 2 are distinguished by their first cohomology groups.

In this paragraph, "cubical" pospace means a disjoint union of unit cubes of dimension n in which finitely many parallelepipeds[25] have been dug out. As we have remarked in the previous paragraph, the choice of a "good" natural system is influenced by "good" properties of the small category we want to calculate the cohomology groups.

[23]geometrically, picture the Rubik's cube, the interior cube is the hole, all other cubes give an object.

[24]it is abusive to write "the" cohomology of a small category because, as far as I know, it depends on the natural system one has put on the small category one wants to calculate "the" cohomology. Hence, it becomes a part of the art to choose a good natural system. In partical cases, the component category of $\vec{X}$ has good properties which induce an "obviously" interesting natural system.

[25]with faces parallel to the faces of the unit cube.

Definition 9 • *A morphism γ is said* **prime** *iff for any morphisms $\gamma_n, ..., \gamma_0$ such that $\gamma = \gamma_n \circ ... \circ \gamma_0, \exists! i \in \{1, ..., n\}, \gamma_i \neq id$.*

- *A category $\mathcal{C}$ is* **generated by primes** *iff any non trivial morphism of $\mathcal{C}$ can be written as a finite composition of prime morphisms.*
- *A category $\mathcal{C}$ is* **homogeneous** *iff $\mathcal{C}$ is generated by primes and for all composable sequences of prime morphisms $(\gamma_n, ..., \gamma_0)$ and $({\gamma'}_{n'}, ..., {\gamma'}_0)$ we have $(\gamma_n \circ ... \circ \gamma_0) = ({\gamma'}_{n'} \circ ... \circ {\gamma'}_0) \Rightarrow n = n'$. $n+1$ is the* **length** *of $\gamma_n \circ ... \circ \gamma_0$.*
- *A category $\mathcal{C}$ is said* **strongly homogeneous** *iff $\mathcal{C}$ is generated by primes and $\forall x, y \in Ob(\mathcal{C}) \exists N_{x,y}$ such that for all composable sequences of prime morphisms $(\gamma_n, ..., \gamma_0)$ with $src(\gamma_0) = x$ and $tgt(\gamma_n) = y$ we have $n = N_{x,y}$. In this case, length depends only on src and tgt.*
- *A category $\mathcal{C}$ is said* **bounded** *iff the length of the composable sequences of $\mathcal{C}$ whose elements are not trivial are bounded, i.e. $\exists N_{\mathcal{C}} \in \mathbb{N}$ such that for all composable sequences $(\gamma_n, ..., \gamma_0)$ satisfying $\gamma_i \neq id$, we have $n \leq N_{\mathcal{C}}$.*
- *A category $\mathcal{C}$ is said* **weakly bounded** *iff $\forall \gamma \in Mo(\mathcal{C}), Max(\{n \in \mathbb{N} / \exists (\alpha_n, ..., \alpha_0)$ such that $\alpha_n \circ ... \circ \alpha_0\}) < +\infty$*

The relations existing between these properties are given in the following diagram

$$\begin{tikzcd}
& \text{Free} \arrow[d, Rightarrow] \\
\text{Strongly Homogeneous} \arrow[r, Rightarrow] \arrow[ddd, Rightarrow, bend right] & \text{Homogeneous} \arrow[d, Rightarrow] \\
\text{Bounded} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{Weakly Bounded} \arrow[d, Rightarrow] \\
\text{Direct} \arrow[d, Rightarrow] & \text{generated by Primes} \arrow[d, Rightarrow] \\
\text{Loop-free} \arrow[r, Rightarrow] & \text{Iso is discrete} \arrow[d, Rightarrow] \\
& \text{Skeletal}
\end{tikzcd}$$

Prime morphisms generalize prime numbers, indeed, the monoid $(\mathbb{N}, +)$ seen as a small category has prime morphisms which are exactly the prime numbers. In fact, it is homogeneous by the famous prime number decomposition theorem. In particular, $\pi_1(\overrightarrow{S^1})$ is homogeneous. The notion of direct categories is related to model category theory, see [Hov99] or [Hir03] for further details.

Definition 10 *A* **linear extension** *of a small category $\mathcal{C}$ is a functor $f : \mathcal{C} \to \lambda$ such that $\forall \gamma \in Mo(\mathcal{C}), f(\gamma) = id \Rightarrow \gamma = id$ and where λ is an* **ordinal**[26] *i.e. a poset whose any non empty subset has a minimum.*
A **direct category** *is a small category having a linear extension. An* **inverse category** *is a small category whose dual is direct.*

Conjecture 3 *The component category of a cubical pospace is homogeneous. Moreover, if its underlying space is connected, the component category is bounded.*

[26]see [Hov99] or [Kun80] or any set theory textbook for the definition.

In general it is not strongly homogeneous as shown by the right side of figure 2 nor bounded because it is always possible to have a infinite disjoint union of connected cubical pospaces $C_0 \sqcup ... \sqcup C_n \sqcup ...$ such that $\forall n \in \mathbb{N}$, C_n has a composable sequence of prime morphisms of length n. Being homogeneous induces a natural system as follows. Given a small category $\mathcal{C}$, the **category of factorizations** of $\mathcal{C}$ (denoted $F\mathcal{C}$) is given by $Ob(F\mathcal{C}) = Mo(\mathcal{C})$ and $F\mathcal{C}[\alpha, \beta]$ is the collection of pairs $(\gamma_2, \gamma_1) \in \mathcal{C}[tgt(\alpha), tgt(\beta)] \times \mathcal{C}[src(\beta), src(\alpha)]$ such that $\beta = \gamma_2 \circ \alpha \circ \gamma_1$ [27]. Given a small category $\mathcal{C}$, a **natural system (of abelian groups)**[28] on $\mathcal{C}$ is a functor $D : F\mathcal{C} \rightarrow \mathbf{Ab}$, where $\mathbf{Ab}$ is the category of abelian groups and group morphisms between them.

Lemma 8 *Let $\mathcal{C}$ be a homogeneous small category. We define a natural system on $\mathcal{C}$ setting $D(\gamma) := Z^{length(\gamma)}$ and for*

$$\begin{tikzcd} \cdot \arrow[r, "\alpha"] \arrow[d, phantom, ""] & \cdot \arrow[d, "\gamma_2"] \\ \cdot \arrow[u, "\gamma_1"] \arrow[r, "\beta"'] \arrow[ur, phantom, "="] & \cdot \end{tikzcd}$$

$D(\gamma_2, \gamma_1) : (x_n, ..., x_1) \in Z^{length(\alpha)} \hookrightarrow (\underbrace{0, ..., 0}_{length(\gamma_2)\ times}, x_n, ..., x_1, \underbrace{0, ..., 0}_{length(\gamma_1)\ times}) \in Z^{length(\beta)}$*, with* $length(\gamma_2)$ *zeros on the left side of* x_n *and* $length(\gamma_1)$ *zeros on the right side of* x_1.

Instead of a (boring and) formal proof that we actually have a functor, observe the following example, suppose $length(\gamma_1) = 1$, $length(\gamma_2) = 2$, $length(\beta) = 6$, then necesarily, $length(\alpha) = 3$ and $D(\gamma_2, \gamma_1)$ is an abelian group embedding pictured by the following diagram:

$$\begin{tikzcd} \{0\} \arrow[d, "0"] & \times & \{0\} \arrow[d, "0"] & \times & Z \arrow[d, "id"] & \times & Z \arrow[d, "id"] & \times & Z \arrow[d, "id"] & \times & \{0\} \arrow[d, "0"] \\ Z & \times & Z & \times & Z & \times & Z & \times & Z & \times & Z \end{tikzcd}$$

It is important to notice that the image of a morphism of $F\mathcal{C}$ only depends on the length of γ_1, γ_2 and α.

7 Dealing with loops: the fundamental monoid

As one can notice, the category POSPC does not contain any satifactory model of the directed circle "$\overrightarrow{S^1}$". Indeed, the only authorized paths of $\overrightarrow{S^1}$ are the clockwise ones[29]. The problem is to modelize this idea. What order relation should equip S^1 in order to make it a pospaces whose dipaths are exactly the clockwise ones? Suppose that such a relation $\leq$ exists, in particular, $t \in \overrightarrow{I} \longmapsto (\cos(-2\pi t), \sin(-2\pi t))$ is clockwise, so we should have $\forall t \in [0,1],\ (0,1) \leq (\cos(-2\pi t), \sin(-2\pi t)) \leq (0,1)$ hence, by antisymmetry, $(\cos(-2\pi t), \sin(-2\pi t)) = (0,1)$ which is a contradiction. A naive solution consists of weakening the definition of a pospace asking $\leq$ be a preorder instead of an order relation. But then, by transitivity, $\forall t, t' \in [0,1],\ (\cos(2\pi t), \sin(2\pi t)) \leq (\cos(2\pi t'), \sin(2\pi t'))$ so $t \in \overrightarrow{I} \longmapsto (\cos(-2\pi t), \sin(-2\pi t))$ which is anticlockwise would also be directed. Marco Grandis approach consists of equiping a topological space X with a set of distinguished paths denoted dX and submited to some conditions. The elements of dX are naturaly called the directed paths. Then it suffices to equip S^1 with the set of all clockwise paths to obtain a model of directed circle. It is also possible to have a model of directed circle by covering S^1 with open subsets, each of which being suitably equiped with and order relation $\leq$ that <u>locally</u> makes S^1 a pospace.
Besides, the fact that a pospace does not have loops makes its fundamental category loop-free, in particular it has no endomorphisms. As a direct consequence, trying to define the "fundamental monoid" of a pospace $\overrightarrow{X}$ as $\pi_1(\overrightarrow{X})[x,x]$ is sound but pointless because $\pi_1(\overrightarrow{X})[x,x] = \{id_x\}$. Introducing loops in our models, the "fundamental monoid" becomes relevant.

[27] If $\mathcal{C}$ is small then so is $F\mathcal{C}$. Moreover, if $\mathcal{C}$ is loop-free then so is $F\mathcal{C}$.
[28] see [BW85] and [Bau91] for further details.
[29] obviously we could have choose the anticlokwise ones

Ideas related to the definition of local pospaces are borrowed from the ones of differential geometry and smooth manifold theory, for a deeper analogy see [LFR99] and [Sok02].

Definition 11 (Local Pospaces) *A* **local pospace** *is a triple $(X, \tau_X, \leq_X)$ such that (X, τ_X) is a topological space, $\leq_X$ a relation on X and $\forall x \in X \exists U$ an open neighbourhood of x such that $(U, \tau_U, \leq_U)$ is a pospace. τ_U and $\leq_U$ are respectively the restriction of τ_X and $\leq_X$ to U. An* **atlas** *of $(X, \tau_X, \leq_X)$ is an open covering $(U_i)_{i \in I}$ of (X, τ_X) such that $\forall i \in I, (U_i, \tau_{U_i}, \leq_{U_i})$ is a pospace. A* **local dimap** *$f : (X, \tau_X, \leq_X) \to (Y, \tau_Y, \leq_Y)$ is a continuous map between underlying topological spaces such that $\exists (U_j)_{j \in J}$ atlas of $(X, \tau_X, \leq_X)$ $\exists (V_j)_{j \in J}$ atlas of $(Y, \tau_Y, \leq_Y)$ satisfying $\forall j \in J$, $f_{U_j \to V_j} : x \in U_j \longmapsto f(x) \in V_j$ is a dimap (i.e. a morphism of* **POSPC***). Local pospaces and local dimaps organize themselves in a category denoted* **LPOSPC**.

As $\overrightarrow{[0,1]}$ is the standard example of pospace, the directed circle $\overrightarrow{S^1}$ is the standard example of local pospace, its relation is described by means of maps $\theta_0 : x \in]0, 2\pi[\longmapsto (\cos(x), \sin(x)) S^1$ and $\theta_1 : x \in]-\pi, \pi[\longmapsto (\cos(x), \sin(x)) S^1$ setting $\forall x, y \in]0, 2\pi[,\ \theta_0(x) < \theta_0(y)$ if $x < y$ and $\forall x, y \in]-\pi, \pi[,\ \theta_1(x) < \theta_1(y)$ if $x < y$.

The next definition is due to Marco Grandis in [Gra03] [30]

Definition 12 (d-spaces) *A* **directed space** *or* **d-space** *is a triple (X, τ_X, dX) where (X, τ_X) is a topological space and $dX \subseteq \{$paths of $(X, \tau_X)\}$ with the following conditions*

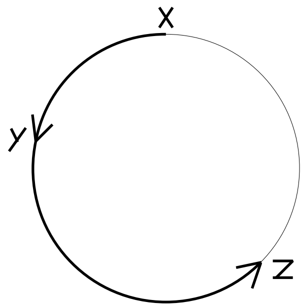
(i) $\{$*constant paths*$\} \subseteq dX$

(ii) for all $\theta : [0,1] \to [0,1]$ continuous and <u>*increasing*</u>*, for all $\gamma \in dX$, $\gamma \circ \theta \in dX$ (dX is stable under di-reparametrization)*

(iii) for all $\gamma_1, \gamma_2 \in dX, \gamma_2 \circ \gamma_1 \in dX$ (dX is stable under concatenation)

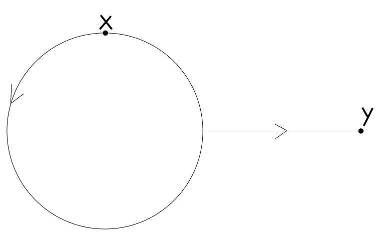
A d-map from (X, τ_X, dX) to (Y, τ_Y, dY) is a continuous map f from (X, τ_X) to (Y, τ_Y) such that $\forall \gamma \in dX$ $f \circ \gamma \in dY$ d-spaces and d-maps organize themselves in a category denoted **dSPC**.

Remark that we have the "obvious" inclusion functors $\mathbf{POSPC} \hookrightarrow \mathbf{LPOSPC} \hookrightarrow \mathbf{dSPC}$. Now let us focus on two examples:



Denoting $\pi_1(\overrightarrow{S^1})$ the fundamental category of $\overrightarrow{S^1}$, we have $\forall x \in \overrightarrow{S^1}$, $\pi_1(\overrightarrow{S^1})[x,x]$ isomorphic to $\mathbb{N}$. Compare $\mathbb{N}$ to the fundamental group of the circle. Precisely, $\pi_1(\overrightarrow{S^1})$ can be described the following way, for each $x, y \in S^1$ there is a distinguished arrow $\alpha_{x,y}$ and the family of distinguished arrows is submitted to the following axiom, $\forall x, y, z \in S^1$, $\alpha_{y,z} \circ \alpha_{x,y} = \alpha_{x,z}$, where $y \in (x, z)$. Here, (x, z) is the clockwise open arc from x to z. Intuitively, the distinguished arrow from $x \in \overrightarrow{S^1}$ to $y \in \overrightarrow{S^1}$ is the clockwise path from x to y on the directed circle, see the left side figure. Then $\forall \gamma \in \pi_1(\overrightarrow{S^1})[x,y]$ $\exists! n \in \mathbb{N}$ such that $\gamma = (\alpha_{y,y})^n \circ \alpha_{x,y}$ and $\forall \gamma \in \pi_1(\overrightarrow{S^1})[x,x]$ $\exists! n \in \mathbb{N}$ such that $\gamma = (\alpha_{x,x})^n$. Hence we could define the fundamental monoid of $\overrightarrow{S^1}$ as $(\mathbb{N}, +)$.

The idea of the fundamental monoid is attractive but does not work because, in general, it depends on the base point x:



The left side picture can easily be described as a local pospace or a directed space denoted $\overrightarrow{X}$ in both cases. Adapting the description of the fundamental category of $\overrightarrow{S^1}$, it is easy to describe the one of $\overrightarrow{X}$. Then we observe that $\pi_1(\overrightarrow{X})[x,x] \cong (\mathbb{N}, +)$ in MON - the category of monoids - while $\pi_1(\overrightarrow{X})[y,y] \cong \{\bullet\}$. The base point dependence makes impossible to define <u>the</u> fundamental monoid of $\overrightarrow{X}$ as the straightforward generalization of the fundamental group.

[30] [Gra03] also contains a definition of local pospace which differs from the presently given one.

In addition, the component category of $\overrightarrow{S}^1$ is its fundamental one. Indeed, none of the morphisms $\alpha_{x,y}$ of $\pi_1(\overrightarrow{S}^1)$ is Yoneda inversible. By definition, if $\alpha_{x,y}$ were Yoneda inversible then, since $\pi_1(\overrightarrow{S}^1)[y,x] \neq \emptyset$, we would have a morphism g from y to x such that $g \circ \alpha_{x,y} = id_x$, which is impossible. Hence, as any arrow of $\pi_1(\overrightarrow{S}^1)$ can be written as a composition of α's, none of them is Yoneda inversible.
More generaly, given a small category $\mathcal{C}$ and $\varepsilon \in \mathcal{C}[x,x]$, an endomorphism of $\mathcal{C}$ such that $\forall n \in \mathbb{N}\ \varepsilon^n \neq id_x$, if $\varepsilon = \gamma \circ \delta$, then γ and δ are not Yoneda inversible. The argument is exactly the same as in the case of $\pi_1(\overrightarrow{S}^1)$. It follows that, even if the definition remains sound, the component category does not reduce the size of a small category when it contains endomorphisms like ε. In particular it does not reduce the size of the fundamental category of a local pospace or a directed space which "contains" $\overrightarrow{S}^1$. Still the next result can provide a way to solve this problem:

Proposition 1 *Let $\mathcal{C}$ be a small category. Suppose that $\sigma : x \longrightarrow y$ is a morphism of $\mathcal{C}$:*

(i) **If** $\forall \delta \in \mathcal{C}[x,x]\ \exists! \gamma \in \mathcal{C}[y,y]$ *such that* $\sigma \circ \delta = \gamma \circ \sigma$ **then** *the map* $\Phi_\sigma : \delta \in \mathcal{C}[x,x] \longmapsto \gamma \in \mathcal{C}[y,y]$ *is a morphism of monoids*

(ii) **If** $\forall \gamma \in \mathcal{C}[y,y]\ \exists! \delta \in \mathcal{C}[x,x]$ *such that* $\gamma \circ \sigma = \sigma \circ \delta$ **then** *the map* $\Psi_\sigma : \gamma \in \mathcal{C}[y,y] \longmapsto \delta \in \mathcal{C}[x,x]$ *is a morphism of monoids*

(iii) **If** $\forall \delta \in \mathcal{C}[x,x]\ \exists! \gamma \in \mathcal{C}[y,y]$ *such that* $\sigma \circ \delta = \gamma \circ \sigma$ *and* $\forall \gamma \in \mathcal{C}[y,y]\ \exists! \delta \in \mathcal{C}[x,x]$ *such that* $\gamma \circ \sigma = \sigma \circ \delta$ **then** $\Psi_\sigma \circ \Phi_\sigma = Id_{\mathcal{C}[x,x]}$ *and* $\Phi_\sigma \circ \Psi_\sigma = Id_{\mathcal{C}[y,y]}$

PROOF. $\sigma \circ id_x = id_y \circ \sigma$, thus $\Phi_\sigma(id_x) = id_y$. Moreover, $\sigma \circ (\delta_2 \circ \delta_1) = \Big(\Phi_\sigma(\delta_2 \circ \delta_1)\Big) \circ \sigma$ and $\sigma \circ (\delta_2 \circ \delta_1) = (\sigma \circ \delta_2) \circ \delta_1 = \Big(\Phi_\sigma(\delta_2) \circ \sigma\Big) \circ \delta_1 = \Phi_\sigma(\delta_2) \circ (\sigma \circ \delta_1) = \Big(\Phi_\sigma(\delta_2)\Big) \circ \Big(\Phi_\sigma(\delta_1)\Big) \circ \sigma$. By uniqueness, $\Phi_\sigma(\delta_2 \circ \delta_1) = \Phi_\sigma(\delta_2) \circ \Phi_\sigma(\delta_1)$, hence Φ_σ is a morphism of monoids. The same holds for Ψ_σ dualizing everything. Suppose we have the hypothesis of the third point, then $\sigma \circ \delta = \Phi_\sigma(\delta) \circ \sigma = \sigma \circ \Psi_\sigma(\Phi_\sigma(\delta))$, hence, by uniqueness, $\Psi_\sigma(\Phi_\sigma(\delta)) = \delta$. The same way, $\Phi_\sigma(\Psi_\sigma(\gamma)) = \gamma$.

Proposition 2 *Let $\mathcal{C}$ be a small category. Suppose that $\sigma : x \longrightarrow y$ is a morphism of $\mathcal{C}$ such that $f_\sigma : \delta \in \mathcal{C}[x,x] \longmapsto \sigma \circ \delta \in \mathcal{C}[x,y]$ and $g_\sigma : \gamma \in \mathcal{C}[x,x] \longmapsto \gamma \circ \sigma \in \mathcal{C}[x,y]$ are bijective. Then the hypothesis of the third point of proposition 1 are satisfied.*

PROOF. Given $\delta \in \mathcal{C}[x,x]$, by definition of the bijections f and g, $\gamma := g_\sigma^{-1}(\sigma \circ \delta)$ is the only element of $\mathcal{C}[y,y]$ such that $\sigma \circ \delta = \gamma \circ \sigma$. Of course, given $\gamma \in \mathcal{C}[y,y]$, by defintion of the bijections f and g, $\delta := f_\sigma^{-1}(\gamma \circ \sigma)$ is the only element of $\mathcal{C}[x,x]$ such that $\gamma \circ \sigma = \sigma \circ \delta$. In particular, $\Phi_\sigma = g_\sigma^{-1} \circ f_\sigma$ and $\Psi_\sigma = f_\sigma^{-1} \circ g_\sigma$.

Corollary 3 *Any Yoneda inversible morphism satisfies the hypothesis of proposition 2*

For example, we remark that $\forall x, y \in S^1$, $\alpha_{x,y}$ satisfies the hypothesis of the third point of proposition 2 and of proposition 2, nevertheless, as we have already seen, they are not Yoneda inversible.

References

[Bau91] Hans Joachim Baues. *Combinatorial Homotopy and 4-Dimensional Complexes.* De Gruyter expositions in Mathematics. Walter de Gruyter, 1991.

[BBP99] M.A. Bednarczyk, A.M. Borzyszkowski, and W. Pawlowski. Generalised congruences-epimorphisms in cat. *Theory and Applications of Categories*, 5(11), 1999.

[Bor94a] Francis Borceux. *Handbook of Categorical Algebra 1 : Basic Category Theory*, volume 50 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1994.

[Bor94b] Francis Borceux. *Handbook of Categorical Algebra 3 : Categories of Sheaves*, volume 52 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1994.

[BW85] Hans-Joachim Baues and Gunther Wirsching. Cohomology of small categories. *Journal of Pure and Applied Algebra*, 38, 1985.

[CC92] P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2(4):511–547, August 1992.

[FGHR04] Lisbeth Fajstrup, Eric Goubault, Emmanuel Haucourt, and Martin Raussen. Component categories and the fundamental category. *APCS*, 12(1):81–108, February 2004.

[Gou95] Eric Goubault. *The Geometry of Concurrency.* PhD thesis, Ecole Normale Supérieure, 1995. also available at http://www.dmi.ens.fr/˜goubault.

[Gou03a] Eric Goubault. Inductive computation of components categories of mutual exclusion models. 2003.

[Gou03b] Eric Goubault. A note on inessential morphisms. december 2003.

[Gra03] Marco Grandis. Directed homotopy theory, i. the fundamental category. *Cahiers Top. Géom. Diff. Catég*, 44:281–316, 2003.

[GZ67] P. Gabriel and M. Zisman. *Calculus of fractions and homotopy theory.* Ergebnisse der Mathematik und ihrer Grenzgebiete. Springer Verlag, 1967.

[Hig71] Philip J. Higgins. *Categories and Groupoids.* Van Nostrand Reinhold, 1971.

[Hir03] P. Hirschhorn. *Model Categories and their Localizations*, volume 99 of *Mathematical Surveys and Monographs.* American Mathematical Society, 2003.

[Hov99] Mark Hovey. *Model Categories*, volume 63 of *Mathematical Surveys and Monographs.* American Mathematical Society, 1999.

[Joh82] Peter T. Johnstone. *Stone Spaces*, volume 3 of *Cambridge studies in advanced mathematics.* Cambridge University Press, 1982.

[Kun80] Kenneth Kunen. *Set Theory: An Introduction to Independence Proofs*, volume 102 of *Studies in Logic and The Foundations of Mathematics.* North-Holland, 1980.

[LFR99] Eric Goubault Lisbeth Fajstrup and Martin Raussen. Algebraic topology and concurrency. *submitted to Theoretical Computer Science*, 1999. also technical report, Aalborg University, available at http://www.ipipan.gda.pl/ stefan/AlgTop/reports.html.

[Mas91] William S. Massey. *A Basic Course in Algebraic Topology*, volume 127 of *Gradute Texts in Mathematics.* Springer-Verlag, 1991.

[Sok02] S Sokolowski. A case for po-manifolds : in chase after a good topological model for concurrency. Technical report, Institute of Computer Science, Gdansk Division, 2002.

[Swi02] Robert M. Switzer. *Algebraic Topology - Homology and Homotopy.* Springer, 2002. reprint of the 1975 edition.

A Dihomotopy Double Category of a Po-Space Extended Abstract

Ulrich Fahrenberg

Dept. of Mathematical Sciences, Aalborg University
9220 Aalborg East, Denmark. Email: uli@math.auc.dk

Abstract

We introduce a dihomotopy invariant of a po-space in dimension 2, its dihomotopy double category. This is a generalisation both of the fundamental category of a po-space, and of the double homotopy groupoid of a topological space. We conjecture a van Kampen theorem for the dihomotopy double category, thus making available a tool for calculations.

1 Introduction

The fundamental category, together with its component category, of a (local) po-space is a fundamental tool in the "geometric" analysis of concurrent systems, see [8,11,10,15,9,13] for some accounts of its uses.

In "ordinary" algebraic topology, as opposed to the "directed topology" of local po-spaces, one has analogues of the fundamental group in all dimensions. An analogue of the fundamental category in dimensions other than 1 has hitherto been missing.

The search for higher dihomotopy invariants is complicated by the fact that the proper counterpart to the fundamental category in ordinary topology is not the fundamental group, but the fundamental *groupoid*, cf. [2], of a topological space. Higher homotopy groupoids for topological spaces have been known since [4,5], but these are defined for *filtered* spaces and seem to resist any transfer to the directed setting.

Following up on a paper [12], Brown et.al. [3] define a notion of homotopy double groupoid for a topological space *without* a presupposed filtration. The present article constructs an analogue of this structure for directed topology.

2 The Singular Cubical Set of a Po-Space

A po-space is a partially ordered topological space such that the order is closed in the product topology. The category of po-spaces and monotone continuous

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

maps (*dimaps*) is denoted poTop. Note [14] that a po-space is Hausdorff.

The singular cubical set of a po-space X is the graded set $RX = \{R_n X\}_{n\in\mathbb{N}}$, where

$$R_n X = \mathsf{poTop}(\vec{I}^n, X)$$

together with face maps $\delta_i^\nu : R_n X \to R_{n-1} X$ $(i = 1, \ldots, n, \nu = 0, 1)$, degeneracies $\varepsilon_i : R_n X \to R_{n+1} X$ $(i = 1, \ldots, n+1)$, connections $\gamma_i^\nu : R_n X \to R_{n+1} X$ $(n \geq 1,\ i = 1, \ldots, n,\ \nu = 0, 1)$, and partially defined compositions $+_i : R_n X \times R_n X \to R_n X$ $(i = 1, \ldots, n)$. Here $\vec{I}$ denotes the (totally) ordered unit interval, $\vec{I}^n$ its n-fold product with the product order, and the natural numbers include zero.

With this structure, RX is a cubical set with connections and compositions in the sense of [4,1]. We will only need the cubical structure in dimensions 0 through 3, so we give an account of the definition of the structure only for these dimensions. For a full account see [4,1].

We will in general denote *points* in $R_0 X = X$ by x, y, *dipaths* in $R_1 X$ by a, b, *disquares* in $R_2 X$ by u, v, and *dicubes* in $R_3 X$ by α, β. The definition of the structure maps in dimensions 0 to 3 is as follows:

$$\begin{array}{ccc}
\delta_1^\nu a = a(\nu) & \delta_1^\nu u(s) = u(\nu, s) & \delta_2^\nu u(s) = u(s, \nu) \\
\delta_1^\nu \alpha(s,t) = \alpha(\nu, s, t) & \delta_2^\nu \alpha(s,t) = \alpha(s, \nu, t) & \delta_3^\nu \alpha(s,t) = \alpha(s, t, \nu) \\
\varepsilon_1 x(s) = x & \varepsilon_1 a(s,t) = a(t) & \varepsilon_2 a(s,t) = a(s) \\
\varepsilon_1 u(r,s,t) = u(s,t) & \varepsilon_2 u(r,s,t) = u(r,t) & \varepsilon_3 u(r,s,t) = u(r,s)
\end{array}$$

$$\begin{array}{cc}
\gamma_1^0 a(s,t) = a(\max(s,t)) & \gamma_1^1 a(s,t) = a(\min(s,t)) \\
\gamma_1^0 u(r,s,t) = u(\max(r,s), t) & \gamma_1^1 u(r,s,t) = u(\min(r,s), t) \\
\gamma_2^0 u(r,s,t) = u(r, \max(s,t)) & \gamma_2^1 u(r,s,t) = u(r, \min(s,t))
\end{array}$$

In that our n-cubes are directed, we miss the *reflection* maps, given by reversing n-cubes in one or more variables, of the standard singular cubical set [4]. This will have the effect that some relations defined by filling in n-cubes which are equivalences in [3], are not symmetric in our setting.

3 The Dipath Category

The one-dimensional part of our dihomotopy double category consists of dipaths modulo reparametrisation. Let $a, b \in R_1 X$, then we say that a and b are *thinly equivalent*, $a \sim_T b$, if there exist surjective dimaps $\phi, \psi : \vec{I} \to \vec{I}$ such that $a \circ \phi = b \circ \psi$. That is, a and b are thinly equivalent if there are reparametrisations of a and b that coincide.

Proposition 3.1 *The relation $\sim_T$ is an equivalence on $R_1 X$ and a congruence with respect to $+_1$. Also, $\varepsilon_1 \delta_1^0 a +_1 a \sim_T a \sim_T a +_1 \varepsilon_1 \delta_1^1 a$ for any*

$a \in R_1X$, and if $a, b, c \in R_1X$ are such that $\delta_1^1 a = \delta_1^0 b$ and $\delta_1^1 b = \delta_1^0 c$, then $a +_1 (b +_1 c) \sim_T (a +_1 b) +_1 c$.

We denote $\rho_1 X = R_1X/{\sim_T}$. As $a \sim_T b$ implies $\partial a = \partial b$, we can define face maps $\delta_1^\nu : \rho_1 X \to R_0X$ by $\delta_1^\nu \langle a\rangle = \langle \delta_1^\nu a\rangle$, and concatenation of paths passes to the quotient by $\langle a\rangle +_1 \langle b\rangle = \langle a +_1 b\rangle$, defined if and only if $\delta_1^1\langle a\rangle = \delta_1^0\langle b\rangle$.

Composing the degeneracies $\varepsilon_1 : R_0X \to R_1X$ with the quotient map yields maps $\varepsilon_1 : R_0X \to \rho_1X$, and the operation $+_1$ on $\rho_1 X$ is associative with units $\varepsilon_1\langle x\rangle$, $x \in X$. Hence the pair (ρ_1X, R_0X) is a category, the *dipath category* of X. By abuse of notation, we shall also denote this category by ρ_1X; note that, contrary to usual (non-directed) paths, dipaths are in general not reversible, hence ρ_1X is in general not a groupoid.

If $f : X \to Y$ is a dimap, and $a \sim_T b$ in X, then also $f \circ a \sim_T f \circ b$ in Y, hence f induces a morphism of categories $f^* : \rho_1X \to \rho_1Y$, and ρ_1 is a functor $\mathsf{poTop} \to \mathsf{Cat}$.

4 The Dihomotopy Double Category

Before we can enter dimension 2, we need to express the relation $\sim_T$ in a more "cubical" way. We say that a disquare $u \in R_2X$ is *thin* if it has a factorisation

$$u : \vec{I}^2 \to \vec{I} \to X$$

If, in addition, $\delta_2^0 u$ and $\delta_2^1 u$ are degenerate, we call u a *thin elementary dihomotopy* and write $u : \delta_1^0 u \sim_T^{\mathrm{e}} \delta_1^1 u$ or simply $\delta_1^0 u \sim_T^{\mathrm{e}} \delta_1^1 u$.

Lemma 4.1 *The relation $\sim_T$ is the transitive, symmetric closure of the relation $\sim_T^{\mathrm{e}}$.*

We want to define an equivalence relation on R_2X, relating disquares which are dihomotopy equivalent relative to the boundary ∂I^2. As we were quotienting out reparametrisations in dimension 1 however, we need to take this into account in defining our equivalence relation.

The equivalence relation $\equiv_T$ to be defined on R_2X will introduce a structure of *edge-symmetric double category* on the triple $(R_2X/{\equiv_T}, \rho_1X, R_0X)$. The following ought to be the shortest possible definition of edge-symmetric double category: A double category is an internal category in the category of small categories, and it is said to be edge-symmetric if the object set of the morphisms object equals the morphism set of the objects object.

As all the double categories in this paper are edge-symmetric, we shall henceforth omit the word "edge-symmetric."

We say that a dicube $\alpha \in R_3X$ has *thin boundary* if the four faces $\delta_2^\nu \alpha$, $\delta_3^\nu \alpha$ are thin elementary dihomotopies; note that this implies that the four horizontal edges $\delta_2^\nu \delta_2^\mu \alpha$ are degenerate. If α has thin boundary, we write $\alpha : \delta_1^0 \alpha \equiv_T^{\mathrm{e}} \delta_1^1 \alpha$ or just $\delta_1^0\alpha \equiv_T^{\mathrm{e}} \delta_1^1\alpha$, and we let $\equiv_T$ be the symmetric, transitive closure of $\equiv_T^{\mathrm{e}}$.

Proposition 4.2 *The relation $\equiv_T$ is an equivalence on R_2X and a congruence with respect to $+_1$ and $+_2$. If $u \equiv_T v$, then $\delta_i^\nu u \sim_T \delta_i^\nu v$, $i = 1, 2$, and if $a \sim_T b$, then $\varepsilon_i a \equiv_T \varepsilon_i b$, $i = 1, 2$.*

We denote the quotient $\rho_2 X = R_2X/{\equiv_T}$ and its elements by $\langle u \rangle$; by the lemma we can introduce face maps and degeneracies $\delta_i^\nu : \rho_2X \to \rho_1X$, $\varepsilon_i : \rho_1X \to \rho_2X$ by

$$\delta_i^\nu \langle u \rangle = \langle \delta_i^\nu u \rangle \qquad \varepsilon_i \langle a \rangle = \langle \varepsilon_i a \rangle$$

The triple (ρ_2X, ρ_1X, R_0X), which we by another abuse of notation also shall denote ρ_2X, is a cubical set with these mappings.

The following technical lemma will allow us to define compositions in ρ_2X.

Lemma 4.3 *Let $u, v \in R_2X$ such that $u(s,t) = P(\phi_1(s), \psi_1(t))$, $v(s,t) = P(\phi_2(s), \psi_2(t))$ for some $\phi_1, \psi_1, \phi_2, \psi_2 : \vec{I} \to \vec{I}$, $P \in R_2X$, and assume that $\phi_1(\nu) = \phi_2(\nu)$, $\psi_1(\nu) = \psi_2(\nu)$ for $\nu = 0, 1$. Then $u \equiv_T v$.*

Corollary 4.4 *Given $u, v \in R_2X$ such that $\delta_1^1 u \sim_T \delta_1^0 v$, then there exist $\hat{u} \equiv_T u$, $\hat{v} \equiv_T v$ such that $\delta_1^1 \hat{u} = \delta_1^0 \hat{v}$, and similarly with δ_1^ν replaced by δ_2^ν throughout.*

Proof. By lemma 4.1 we have surjective reparametrisation $\phi, \psi : \vec{I} \to \vec{I}$ such that $\delta_1^1 u \circ \phi = \delta_1^0 v \circ \psi$. Define $\hat{u}$, $\hat{v}$ by $\hat{u}(s,t) = u(s, \phi(t))$, $\hat{v}(s,t) = v(s, \psi(t))$. Then $\delta_1^1 \hat{u} = \delta_1^0 \hat{v}$, and $u \equiv_T \hat{u}$, $v \equiv_T \hat{v}$ by lemma 4.3. For the statement with δ_1^ν replaced with δ_2^ν, $\hat{u}$ and $\hat{v}$ are given by $\hat{u}(s,t) = u(\phi(s), t)$, $\hat{v}(s,t) = v(\psi(s), t)$.□

We are ready to define operations $+_1$, $+_2$ on ρ_2X. Let $\langle u \rangle, \langle v \rangle \in \rho_2X$, and assume first that $\delta_1^1 \langle u \rangle = \delta_1^0 \langle v \rangle$. Then $\delta_1^1 u \sim_T \delta_1^0 v$, hence by corollary 4.4 there are disquares $\hat{u} \equiv_T u$, $\hat{v} \equiv_T v$ such that $\delta_1^1 \hat{u} = \delta_1^0 \hat{v}$, and we can define $\langle u \rangle +_1 \langle v \rangle = \langle \hat{u} +_1 \hat{v} \rangle$. If $\delta_2^1 \langle u \rangle = \delta_2^0 \langle v \rangle$, we can apply the second statement of corollary 4.4 similarly.

Proposition 4.5 *The operations $+_1$, $+_2$ on ρ_2X are well-defined and introduce two category structures on the pair (ρ_2X, ρ_1X), with identities $\varepsilon_1 \langle a \rangle$, $\varepsilon_2 \langle a \rangle$, $\langle a \rangle \in \rho_1X$, respectively. Also, the maps $\delta_i^\nu : \rho_2X \to \rho_1X$, $\varepsilon_i : \rho_1X \to \rho_2X$ are morphisms of categories, and*

$$\big(\langle u \rangle +_1 \langle v \rangle\big) +_2 \big(\langle w \rangle +_1 \langle z \rangle\big) = \big(\langle u \rangle +_2 \langle w \rangle\big) +_1 \big(\langle v \rangle +_2 \langle z \rangle\big) \tag{1}$$

whenever both sides are defined.

In other words, the triple (ρ_2X, ρ_1X, R_0X) forms a double category, the *dihomotopy double category* of X. We shall again abuse notation and also denote this double category by ρ_2X; noting that a dimap $f : X \to Y$ maps cubes with thin boundary to cubes with thin boundary, we see that f induces a morphism of double categories $f^* : \rho_2X \to \rho_2Y$.

5 Ongoing and Future Work

We conjecture that the relation $\equiv_T$ is related to dihomotopy in dimension 2 in the following way, where $\sim$ is dihomotopy, i.e. the equivalence generated by the elementary relation "$u \sim^{\mathrm{e}} v \in R_2X$ iff there exists $\alpha \in R_3X$ such that $u = \delta_1^0\alpha$, $v = \delta_1^1\alpha$, and the four other faces of α are degenerate:"

Conjecture 5.1 *Given $u, v \in R_2X$ such that $\delta_2^\nu u$, $\delta_2^\nu v$ are degenerate, then $u \sim v$ if and only if $u \equiv_T v$.*

This is analogous to the result in [12,3].

The connections $\gamma_1^0, \gamma_1^1 : R_1X \to R_2X$ induce mappings $\gamma_1^0, \gamma_1^1 : \rho_1X \to \rho_2X$, which are connections in the sense of [7] and hence introduce a *thin structure* on ρ_2X. We conjecture that thinness in R_2X is equivalent to thinness in ρ_2X:

Conjecture 5.2 *An element $\langle u\rangle \in \rho_2X$ is thin if and only if there exists $v \in R_2X$ such that v is thin and $u \equiv_T v$.*

Conjecture 5.2 is the major stepping stone towards the following van Kampen theorem for the dihomotopy double category, analogous to [6]:

Conjecture 5.3 *Let $X \in \mathsf{poTop}$, and let $\mathcal{U}$ be a covering of X by po-spaces, i.e. such that $\bigcup_{U\in\mathcal{U}} \operatorname{int} U = X$. For all $U, V \in \mathcal{U}$, let $a_{UV} : U \cap V \hookrightarrow U$, $b_{UV} : U \cap V \hookrightarrow V$, $c_U : U \hookrightarrow X$ be the inclusions, and consider the diagram in the category of double categories*

$$\bigsqcup_{U,V\in\mathcal{U}} \rho_2(U \cap V) \overset{a^*}{\underset{b^*}{\rightrightarrows}} \bigsqcup_{U\in\mathcal{U}} \rho_2 U \xrightarrow{c^*} \rho_2 X$$

where a^*, b^*, c^* are the disjoint unions of the induced mappings a^*_{UV}, b^*_{UV}, c^*_U. Then c^* is the coequaliser of a^* and b^*.

As to future directions of research, there are two obvious things to do. One is to transfer the notion of components [9,13] to the dihomotopy double category, which would then make it usable for actual computations.

The other is to generalise the dihomotopy double category to *local* po-spaces. This is not a trivial task, as local po-spaces admit non-constant loops, while po-spaces don't, a fact we have made strategic use of in the above considerations.

Acknowledgement

I would like to express my gratitude towards Ronnie Brown for many helpful comments, as well as Lisbeth Fajstrup, Emmanuel Haucourt, Eric Goubault, and Martin Raussen.

References

[1] Al-Agl, F. A., R. Brown and R. Steiner, *Multiple categories: the equivalence of a globular and a cubical approach*, Adv. Math. **170** (2002), pp. 71–118, also http://arxiv.org/abs/math/0007009.

[2] Brown, R., "Elements of Modern Topology," McGraw Hill, 1968.

[3] Brown, R., K. A. Hardie, K. H. Kamps and T. Porter, *A homotopy double groupoid of a Hausdorff space*, Theory and Applications of Categories **10** (2002), pp. 71–93, http://www.tac.mta.ca/tac/volumes/10/2/10-02abs.html.

[4] Brown, R. and P. J. Higgins, *On the algebra of cubes*, Journal of Pure and Applied Algebra **21** (1981), pp. 233–260.

[5] Brown, R. and P. J. Higgins, *Colimit theorems for relative homotopy groups*, Journal of Pure and Applied Algebra **22** (1981), pp. 11–41.

[6] Brown, R., K. H. Kamps and T. Porter, *A van Kampen theorem for the homotopy double groupoid of a Hausdorff space*, Preprint (2004), http://www.bangor.ac.uk/~mas010/VKT7.pdf.

[7] Brown, R. and G. H. Mosa, *Double categories, 2-categories, thin structures and connections*, Theory and Applications of Categories **5** (1999), pp. 163–175, http://www.tac.mta.ca/tac/volumes/1999/n7/5-07abs.html.

[8] Fajstrup, L., E. Goubault and M. Raussen, *Algebraic topology and concurrency*, Report R-99-2008, Department of Mathematical Sciences, Aalborg University (1999), http://www.math.auc.dk/research/reports/R-99-2008.ps. Conditionally accepted for publication in *Theoretical Computer Science*.

[9] Fajstrup, L., M. Raussen, E. Goubault and E. Haucourt, *Components of the fundamental category*, Applied Categorical Structures **12** (2004), pp. 81–108.

[10] Goubault, E., *Geometry and concurrency: A User's Guide*, Electronic Notes in Theoretical Computer Science **39** (2000).

[11] Goubault, E. and M. Raussen, *Dihomotopy as a tool in state space analysis*, in: *Proc. LATIN 2002*, Lecture Notes in Computer Science **2286** (2002), pp. 16–37.

[12] Hardie, K. A., K. H. Kamps and R. W. Kieboom, *A homotopy 2-groupoid of a Hausdorff space*, Applied Categorical Structures **8** (2000), pp. 209–234.

[13] Haucourt, E., *A framework for component categories*, in: *Preliminary Proceedings GETCO'04*, BRICS Notes Series (2004), in this volume.

[14] Nachbin, L., "Topology and Order," D. Van Nostrand, 1965.

[15] Raussen, M., *State spaces and dipaths up to dihomotopy*, Homotopy Homology Appl. **5** (2003), pp. 257–280.

Recent BRICS Notes Series Publications

NS-04-2 **Patrick Cousot, Lisbeth Fajstrup, Eric Goubault, Maurice Herlihy, Martin Raußen, and Vladimiro Sassone, editors.** ***Preliminary Proceedings of the Workshop on Geometry and Topology in Concurrency and Distributed Computing, GETCO '04,*** **(Amsterdam, The Netherlands, October 4, 2004), September 2004. vi+80.**

NS-04-1 **Luca Aceto, Willem Jan Fokkink, and Irek Ulidowski, editors.** ***Preliminary Proceedings of the Workshop on Structural Operational Semantics, SOS '04,*** **(London, United Kingdom, August 30, 2004), August 2004. vi+56.**

NS-03-4 **Michael I. Schwartzbach, editor.** ***PLAN-X 2004 Informal Proceedings,*** **(Venice, Italy, 13 January, 2004), December 2003. ii+95.**

NS-03-3 **Luca Aceto, Zoltán Ésik, Willem Jan Fokkink, and Anna Ingólfsdóttir, editors.** ***Slide Reprints from the Workshop on Process Algebra: Open Problems and Future Directions, PA '03,*** **(Bologna, Italy, 21–25 July, 2003), November 2003. vi+138.**

NS-03-2 **Luca Aceto.** ***Some of My Favourite Results in Classic Process Algebra*****. September 2003. 21 pp. Appears in the** ***Bulletin of the EATCS*****, volume 81, pp. 89–108, October 2003.**

NS-03-1 **Patrick Cousot, Lisbeth Fajstrup, Eric Goubault, Maurice Herlihy, Kurtz Alexander, Martin Raußen, and Vladimiro Sassone, editors.** ***Preliminary Proceedings of the Workshop on Geometry and Topology in Concurrency Theory, GETCO '03,*** **(Marseille, France, September 6, 2003), August 2003. vi+54.**

NS-02-8 **Peter D. Mosses, editor.** ***Proceedings of the Fourth International Workshop on Action Semantics, AS 2002,*** **(Copenhagen, Denmark, July 21, 2002), December 2002. vi+133 pp.**

NS-02-7 **Anders Møller.** ***Document Structure Description 2.0*****. December 2002. 29 pp.**

NS-02-6 **Aske Simon Christensen and Anders Møller.** ***JWIG User Manual*****. October 2002. 35 pp.**

NS-02-5 **Patrick Cousot, Lisbeth Fajstrup, Eric Goubault, Maurice Herlihy, Martin Raußen, and Vladimiro Sassone, editors.** ***Preliminary Proceedings of the Workshop on Geometry and Topology in Concurrency Theory, GETCO '02,*** **(Toulouse, France, October 30–31, 2002), October 2002. vi+97.**