BRICS

Basic Research in Computer Science

Preliminary Proceedings of the Workshop on

Geometry and Topology in Concurrency Theory

GETCO ’02

Toulouse, France, October 30–31, 2002

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

BRICS Notes Series NS-02-5

ISSN 0909-3206 October 2002

BRICS NS-02-5 Cousot et al. (eds.): GETCO ’02 Preliminary Proceedings

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/02/5/`

*GE*ometry and *T*opology in *CO*ncurrency theory and distributed systems theory

Preliminary Proceedings of GETCO'2002
Satellite Workshop of Disc'2002
Toulouse, France, October 30–31, 2002

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

The main mathematical disciplines that have been applied 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 complexes 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.

The first workshop on the subject "Geometric and Topological Methods in Concurrency Theory" was held in Aalborg, Denmark, in June 1999. GETCO 2000 was held at Penn State University as a satellite to CONCUR 2000 in August 2000, and GETCO 2001 took place at Aalborg University as a satellite to CONCUR 2001 in August 2001.

The workshop has been financially supported by the Basic Research Institute in Computer Science (Aarhus, Denmark) and also by GDR ARP and GDR ALP of the French CNRS. I would like to thank them for this support. Special thanks are due to Uffe Engberg from BRICS, who kindly and swiftly handled the printing of this preproceedings volume.

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. Last but not least, I wish to thank the organizers of DISC 2002, Mamoun Filali, Philippe Mauran, Gérard Padiou and Philippe Quéinnec, for their cooperation regarding this workshop.

Eric Goubault, the 11'th of October 2002.

Programme

Wednesday 30'th of October.

14:00 - 14:05	Opening
14:05 - 14:55	*Tutorial:* Intrduction to Classical Algebraic Topology by Lisbeth Fajstrup and Martin Raussen
15:05 - 15:55	*Tutorial:* Geometry of Fault-Tolerant Distributed Systems I by Maurice Herlihy
15:55 - 16:10	COFFEE BREAK
16:10 - 17:00	*Tutorial:* Geometry of Fault-Tolerant Distributed Systems II by Maurice Herlihy
17:10 - 18:00	*Tutorial:* Directed Topology and Concurrency: Some Motivations by Eric Goubault

Thursday 31'st of October.

08:00 - 08:40	Using the Topological Characterization of Synchronous Models by G. Adagio
08:50 - 09:30	A Note on Set Agreement with Omission Failures by R.Guerraoui, P. Kouznetsov and B. Pochon
09:40 - 10:20	TBA by S. Rajsbaum
10:20-10:40	BREAK
10:40-11:20	Rewriting Systems and Hochschild-Mitchell Homology by P. Malbos
11:30-12:10	Geometric Aspects of Multiagent Systems by T. Porter
12:10 - 13:20	LUNCH
13:20 - 14:00	The Geometry of timed PV programs by U. Fahrenberg
14:10 - 14:50	A Case for Po-Manifolds: in chase after a good topological model for concurrency by S. Sokolowski
15:00 - 15:40	A convenient Category for the Homotopy Theory of Concurrency by P. Gaucher
15:40 - 16:00	AFTERNOON TEA

Table of Contents

Electronic Notes in Theoretical Computer Science 52 No. 2 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume52.html 12 pages

Using the Topological Characterization of Synchronous Models[⋆]

Giovanni Adagio[1]

Distributed Programming Laboratory
School of Computer Science and Communication Systems
Swiss Federal Institute of Technology in Lausanne (EPFL)

Institute of Mathematics
School of Basic Sciences
Swiss Federal Institute of Technology in Lausanne (EPFL)

Abstract

This paper contributes to the characterization of synchronous models of distributed computing using topological techniques. We consider a generic synchronous model with send-omission failures and use a topological structure corresponding to a bounded number of rounds of the model. We observe some nice properties of the structure and derive from these properties necessary and sufficient conditions to solve consensus in this model.

1 Introduction

Motivations

Several distributed computing models have proliferated in the last decades. Results that have been proven in these models are difficult to compare, essentially because relationships between these models are not clear. Quite recently, some preliminary steps have been taken towards providing a mathematical framework to unify these models: basically, the observation that connectivity is at the heart of many distributed computing lower bounds has led to some topological (or graph-based) characterizations of distributed models [4,7]. To our knowledge, however, the only complete characterization has been given so far for the asynchronous model with process crash failures [7]: the idea is to

[⋆] This work is partially supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation (under grant number 5005-67322).

[1] G. Adagio is a short name for *Gruppo di Algoritmi Distributi e AlGebra a l'InstitutO Politecnico di Losanna.* The group gathers, in alphabetical order, S. Blanc, R. Guerraoui, K. Hess, P. Kouznetsov, P.-E. Parent, B. Pochon, and O. Sauvageot.

represent a set of global states of the system as a mathematical structure called a *simplicial complex* [9]. Interestingly, the iterative model based on the *immediate snapshot memory* [2] has a very regular structure, corresponding to a subdivision of a simplicial complex, and is used to completely characterize the asynchronous model [7]. Very few problems are solvable in the asynchronous model however, and it is very tempting to seek characterizing models with some synchrony assumptions. Some topological constructs characterizing particular executions of a synchronous model have indeed been proposed [6], and were used to derive nice and succinct proofs of various lower bounds results. These were however only partial characterizations (i.e., considering particular executions only) and concerned a completely synchronous system. [2] An open and rather challenging question is how to completely characterize a generic model, parametrized with some synchrony assumptions, in a comprehensible and easy-to-use way.

Contributions

Our approach is based on the iterative round-by-round failure detector model [5], where we consider send-omissions as the only source of failures.

In this preliminary attempt, we illustrate our characterization with a notion of graph sequences, and we (1) give a proof of the lower bound of $f+1$ rounds for consensus in the f-resilient omission model and (2) derive from our characterization an algorithm that matches the lower bound. We believe that the main contribution of this note is the way we derive the algorithm: we use two observations from our characterization about connected components containing omission-free executions. The first observation is a sufficient condition for a connected component to contain a omission-free execution, and the second observation relates, for a connected component with a certain number of rounds, the number of faulty processes with the existence of an omission-free execution in that component.

Our graph-based characterization has a major limitation, however: it is based on the indistinguishability of two global states for one process (it is, in a sense, customized for consensus), and its extension (e.g., to set agreement) is not trivial. We show that such a characterization allows us to reason about executions of an omission model with f possible faulty processes.

Roadmap

Section 2 introduces our system model. Section 3 presents the characterization. Section 4 presents an application of our characterization to consensus by showing that $f+1$ rounds are necessary and sufficient to solve consensus in the f-resilient model.

[2] A semi-synchronous model is also considered in [6], but the difference with a synchronous model is rather small.

2 Model

We consider a distributed system of n processes $\Pi = \{p_1, \ldots, p_n\}$. Processes communicate by message-passing, and each pair of processes is connected by a reliable channel. Processes may however fail by send-omissions. We assume that, in any execution of the system, at most f processes may lose messages, and we call such processes *faulty* processes. We call such a model the f-resilient omission model [5]. An execution of the f-resilient omission model is *omission-free*, if and only if all messages are sent and received in every round.

We represent executions in our model as in the *communication graphs* approach of [4,8]. We assume that processes execute a *full-information* protocol [7]: in each round, every process sends its entire local state to every other process. One round of any such execution can be described by a directed graph, the vertexes of which are labeled by the process ids and their local state in that round [4,8]. There is a directed edge in this graph from process p_i to process p_j, whenever p_j receives the message from p_i. A lack of an edge between processes means that the message is lost. We always assume that a process receives its own message, and we omit the corresponding edge. Therefore, a multi-round execution of such a system is represented by a sequence of graphs, one graph per round.

We call an *r-round execution* an execution in which all processes execute a full-information protocol for r rounds and do not execute any step for any round $r' > r$. An *r-sequence* is the sequence of graphs corresponding to an r-round execution.

3 Topological characterization

Before presenting our characterization, we first recall some basic concepts borrowed from algebraic topology (formally defined, for instance, in [9]). These concepts have been recently used in distributed computing, for instance, in [3,4,6,7].

3.1 Background

We represent a global state of our system of n processes by a $(n-1)$-dimensional simplex $S^{n-1} = \{s_1, \ldots, s_n\}$ of n vertexes, where each vertex $s_i = \langle p_i, v_i \rangle$ corresponds to a process p_i and its local state v_i [7]. A non-empty simplex T is a *face* of a simplex S if and only if all vertexes of T are vertexes of S. A *simplicial complex* C is a set of simplexes, closed under containment, such that any face of any simplex of C is also part of C.

In our model, we consider an initial configuration (i.e., an initial global state) where processes have generic input values. The state of the system at the end of an execution in which processes started with generic values is represented by a simplex.

3.2 Moves

A *move* from an execution e consists in adding or removing a single arrow to e. For an execution e and a move s, we denote by $s(e)$ the execution resulting from applying s to e.

An *elementary* move s from an r-round execution e to an r-round execution e' is a move such that $e' = s(e)$, and there exists at least one process p_i such that p_i's local state after round r in e or e' is identical when starting from the same initial state in both executions.

A process p_i can *change its mind* at the beginning of a r-round execution e if and only if there exists a process p_j such that p_j's local state after r rounds does not depend on p_i's initial state.

We define a notion of path between two executions as in [4] (called a similarity chain in [4]). For a given execution e, a *k-path* P is a sequence $\{s_i\}_{i=1}^{k}$ of elementary moves, such that s_1 is an elementary move for e, s_2 is an elementary move for $s_1(e)$, etc. We denote by $length(P)$ the length of the sequence and we simply say a *path* when the length of the sequence is not relevant.

For any execution e, we denote by $P(e)$ the execution resulting from successively applying the elementary moves of P to e (that is, applying s_1 to e, s_2 to $s_1(e)$, etc.). For two executions e and e', we say that e' is *reachable* from e, and we write $e \sim e'$, if and only if there exists a path P such that $P(e) = e'$.

3.3 Characterization and Connectivity

The characterization takes into account all possible executions of a full-information protocol running in the f-resilient omission model, and corresponds to a generalization of [1]. The generalization is threefold: (i) we consider an arbitrary number of failures f ([1] considers only one failure), (ii) we generalize the failures to send-omissions (which allows for some synchrony assumptions [5]) and (iii) we have a notion of *degree* of similarity between two global states (the degree of similarity corresponds to the dimension of the intersection of the two corresponding simplexes) [6].

We established a correspondence between a set of r-sequences of graphs and a simplicial complex, by identifying an r-sequence of graphs in our model with a simplex of dimension $n-1$. In our characterization, we thus consider the simplicial complex that represents all possible r-executions of our system.

For any given r-execution e and elementary move s, consider the execution $e' = s(e)$. Executions e and e' both correspond to simplexes in a simplicial complex. By definition of s, there exists at least one process p_j which does not distinguish between e and e', so we can *glue* the simplexes corresponding to e and e' on the vertex corresponding to p_j (or on the simplex determined by processes that do not distinguish the two global states).

Figure 1 gives an example of how simplexes are glued together. A tetrahedron represents an execution of a 4-process protocol. The different ways of

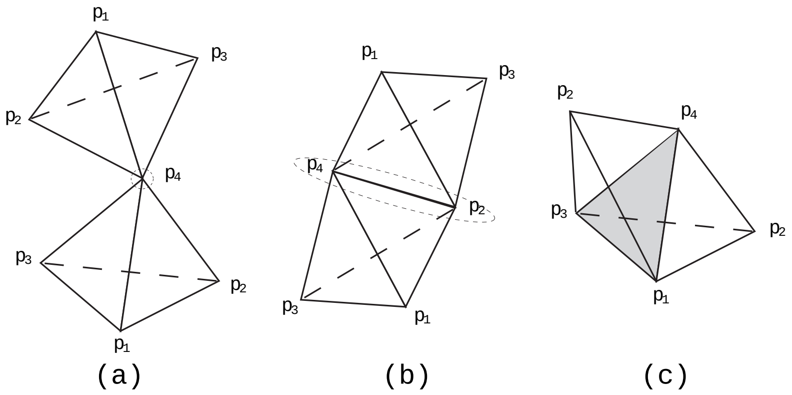


Fig. 1. Gluing simplexes

gluing two simplexes are given by the following three situations. In case **(a)**, process p_4 is the only process that cannot distinguish between the two executions. In case **(b)**, processes p_4 and p_2 cannot distinguish between the two executions. In case **(c)**, processes p_4, p_3 and p_1 cannot distinguish between the two executions.

Two simplexes S_i and S_j are said to be *adjacent* in a simplicial complex if their corresponding executions e_i and e_j differ by at least one elementary move. Two adjacent simplexes S_i and S_j always have a face, corresponding to $S_i \cap S_j$, in common.

A k-path for an execution e implies a sequence of $k+1$ simplexes $S_0, \ldots, S_k$ in a simplicial complex, of dimension $(n-1)$ each, such that S_0 corresponds to execution e, for every $i \in [1, k-1]$, S_i and S_{i+1} are adjacent.

Two simplexes of a simplicial complex are *congruent* if and only if there exists a path in the simplicial complex that connect both simplexes. Congruency is a reflexive, symmetric and transitive relation, and we can therefore partition a simplicial complex into congruency classes, which form the *connected components* of the complex. In the rest of the paper, we interchangeably use the notion of execution, simplex, or sequence of graphs.

3.3.1 On the sufficiency for omission-freedom

The following lemma gives a sufficient condition for a connected component to contain the omission-free execution. Roughly speaking, it says that if a component includes a path in which every process can change its mind, then the component also includes the omission-free execution.

Lemma 3.1 *If a connected component contains executions e_i $(1 \leq i \leq n)$, such that process p_i can change its mind in e_i, then this component contains the omission-free execution e_0.*

Proof. Let C be any connected component satisfying the condition of the lemma, i.e., there exists an execution $e \in C$ and a path $P = \{s_i\}$ of elementary moves, defined on e, that passes through executions $e_1, \ldots, e_n$ where, respectively, processes $p_1, \ldots, p_n$ change their mind. We want to show that there exists a sequence of elementary moves that connects e with the omission-free execution e_0.

Consider an elementary move s, defined on an execution e, that modifies (adds or removes an arrow in) the r-th round of e. Let e' be an execution that is identical to e in rounds $r' \geq r$. Obviously, s applied to e' is also elementary, as the same process cannot distinguish e' and the result of applying s to e'.

Now it is easy to see that if we drop from P all elementary moves which remove arrows in the first round of executions, then we still obtain a path P' of elementary moves. The corresponding execution e'_1 differs from e_1 in the first round only. Thus p_1 can change its mind at the end of the first round of e'_1, and some process does not see it at the end of the execution (otherwise p_1 could not change its mind in e_1). So we can let p_1 receive all messages in the first round of e'_1, i.e., adding arrows from all processes to p_1 in the first round is an elementary move. We then continue applying the moves of P' until we reach e'_2 and so on.

As a result, we obtain a path that connects e to an execution in which every message is received in the first round. Moreover, the sequence passes through executions $e^1_1, \dots, e^1_n$ where, respectively, processes $p_1, \dots, p_n$ can change their states at the end of the first round.

Inductively applying the argument to the second round etc., we finally obtain a path that connects e to the omission-free execution. Note that we modify the elementary moves only by adding arrows, i.e., we do not introduce more failures. As a result, we cannot violate the limit of at most f faulty processes in every execution of our model. □

3.3.2 On the necessity of omissions

We observe another property of connected components. Roughly speaking, this property gives the least number of faulty processes in some execution of a connected component which contains at the same time (i) the omission-free execution and, (ii) an execution in which the initial local state of several processes is never received by some process.

Lemma 3.2 *Any path P that connects the r-round omission-free execution e_0 with an execution e^i in which i processes can simultaneously change their minds ($i < n$), passes by an execution in which at least $r + i - 1$ processes are faulty.*

Proof. We proceed by induction on r. The case $r = 1$ is trivial: a process can change its mind in a 1-round execution only if it is faulty. Now assume that the claim holds for r-round executions, for any i. Consider a path P that satisfies the condition of the lemma, namely, P connects the $(r + 1)$-round omission-free execution e_0 with an execution e^i in which i processes can simultaneously change their minds.

(i) Assume that a set X of i processes can change their minds in e^i. It is not difficult to see that we can remove all the links departing from the set X

in the first round in an elementary way: for any process $p \in X$, there is a process q that cannot see the change of p's initial value. Thus, q cannot see that a link from p is removed. We define by $\tilde{P}$ an extension of path P by adding the moves in which all the links from X are removed. The last execution of $\tilde{P}$ is $\tilde{e}^i$ in which no message from X is ever received.

(ii) It is important to notice that the set of processes that are faulty in all executions between e^i and $\tilde{e}^i$ are also faulty in e^i. Indeed all processes of X are already faulty in e^i and no more faulty processes can be obtained by removing links from them.

(iii) Consider the last move of $\tilde{P}$ in which the last link from X to some process $p \in \Pi \backslash X$ is removed. We denote the resulting execution by e^i_{r+1} and the last r rounds of it by e^i_r.

(iv) Since the removal of the link to p in the first round is an elementary move, then p can change its initial value in the beginning of the second round. As a result, p can change its mind in e^i_r. Since all processes from X can change their minds in e^i_{r+1}, they can change their minds also in e^i_r. As a result, path P restricted to the last r rounds leads to an execution e^i_r in which $i+1$ processes from the set $X \cup \{p\}$ can change their minds.

(v) By the induction hypothesis, $\tilde{P}$ passes through an execution e, in which at least $r+i$ processes are faulty. The executions obtained by the moves $\tilde{P} \backslash P$ do not contain more faulty processes than e^i. Thus, P also passes through an execution in which at least $r+i = (r+1)+i-1$ processes are faulty.

□

4 Application to Consensus

Informally, in the consensus problem, each process proposes a value, and all processes must then agree (or decide) on a single value among the proposed ones. More precisely, we require that (*validity*) every decided value is a proposed value, (*agreement*) no two process decide differently, and (*termination*) every process eventually decides.[3] In particular, if all processes propose the same value, then by the validity requirement this value must be decided. As an immediate consequence we obtain the following proposition.

Proposition 4.1 *Consensus cannot be solved if and only if there exists a connected component C containing, for every $i \in [1, n]$, an r-round execution e_i, such that process p_i can change its mind in e_i.*

Proof. "If" direction ($\Leftarrow$):
Consider a connected component C that satisfies the condition of the proposition, namely, C contains, for every $i \in [1, n]$, execution e_i in which process

[3] Note that a process is not allowed to halt in our model, and thus, every process must eventually decide.

p_i can change its mind. Consider the initial state of the system in which all processes propose the value 0. By validity, all processes decide 0, in any execution, and in particular in e_1. Consider executions $\{e_i\}$ for increasing values of i, starting from e_1, and ending at e_n. Considering execution e_i, one can change the initial value of process p_i to a different value, say 1. After considering execution e_n, all processes now propose 1. By validity, they all decide 1 in e_n. There are two cases to consider: (i) there exists two adjacent executions e', where processes decide 0, and e'', where processes decide 1, or (ii) there exists an execution e_j after which processes decide on 0 if p_j proposes 0 and decide 1 if p_j proposes 1. For case (i) and because of adjacency, there exists at least one process p_j which has the same local state at the end of executions e' and e'' and which decide 0 in e' and 1 in e'' – a contradiction. For case (ii), by definition there exists a process $p_{j'}$ whose local state at the end of execution e_i does not contain p_i's local state, thus $p_{j'}$ decides the same value in e_i independently of p_i's value – a contradiction.

"Only if" direction ($\Rightarrow$):
We prove the contrapositive of the claim, namely that if there is no connected component which contains executions e_i ($1 \leq i \leq n$), such that process p_i can change its mind in e_i, then consensus is solvable.

For any connected component C, we denote by K the set of processes such that for any execution $e \in C$, processes in K cannot change their minds. For any connected component C satisfying the condition of the proposition, $|K| \geq 1$ and every process is aware of the initial values of processes in K, in any execution $e \in C$. We can thus define a deterministic decision function δ_C of the initial values of processes in K. Function δ_C outputs the decision value associated with the component C. By construction, δ_C outputs just a single decision value within C, and thus ensures agreement. Validity follows from the fact that δ_C only takes as arguments initial values of processes. Termination follows from the fact the each process executes a bounded number of rounds before deciding (each execution in C is composed of r rounds). □

4.1 Application to Consensus: Necessity

In this section, we observe another specific property of a connected component. Basically, in any connected component containing the omission-free r-round ($1 \leq r \leq f$) execution e_0, there exists a path from e_0 to an r-execution where a process can change its mind. This allows us to deduce a lower bound of $f + 1$ rounds for solving consensus in the f-resilient omission model.

Theorem 4.2 *If a connected component C contains the omission-free r-round ($1 \leq r \leq f$) execution e_0, there exists an execution $e \in C$, such that a process p can change its mind in e.*

Proof. First note that it suffices to prove the proposition for $r = f$, as this proof immediately leads to the case where $r < f$. By symmetry, as C contains

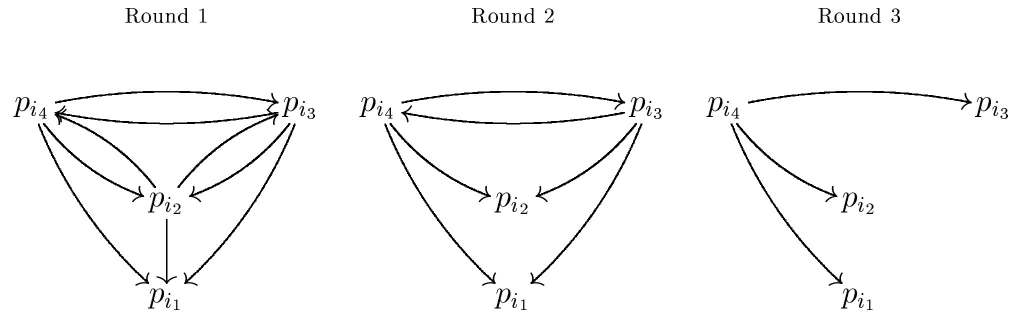


Fig. 2. Execution e with $n = 4$ and $f = 3$

the failure-free execution e_0, the theorem implies that any process can change its mind in some execution in C.

Consider now the f-round execution e where, in each round k, exactly k processes fail by omission. (An execution e is given in Fig. 2 for the case where $n = 4$ and $f = 3$.)

Round 1: Exactly one process p_{i_1} fails by omitting all its messages.

Round 2: Exactly two processes p_{i_1}, p_{i_2} fail by omitting all their messages.

. . .

Round f: Exactly f processes $p_{i_1}, \ldots, p_{i_f}$ fail by omitting all their messages.

In execution e, as p_{i_1} fails by omitting all its messages, it can clearly change its mind. We show how one can construct a path P from the omission-free execution e_0 to execution e. In round f, it is easy to see that we can remove any arrow of any execution so that some process cannot notice it. As a result, we can construct a sequence of elementary moves that connect e_0 to an execution e_1 in which no arrows depart from processes $p_{i_1}, \ldots, p_{i_f}$.

Suppose by induction that, for any $k+1$ processes $p_{i_1}, \ldots, p_{i_{k+1}}$, there exists an execution e_{f-k} ($e_{f-k} \sim e_0$) in which $p_{i_1}, \ldots, p_{i_{k+1}}$ fail by omitting all their messages in round $k' \geq k+1$. We exhibit a sequence of elementary moves to reach execution e_{f-k+1}, where k processes $p_{i_1}, \ldots, p_{i_k}$ fail by omitting all their messages in round k.

Denote by e^0_{f-k+1} the execution identical to e_{f-k} except that, in round k, no message is exchanged among processes $p_{i_1}, \ldots, p_{i_k}$. Indeed, all these processes are silent starting from the next round. Thus, $e_{f-k} \sim e^0_{f-k+1}$. We also denote by $m_1, \ldots, m_j, \ldots, m_l$ the messages sent in round k, from processes $p_{i_1}, \ldots, p_{i_k}$ to the remaining processes. We consider the sequence of elementary moves that successively remove messages $m_1, \ldots, m_j, \ldots, m_l$, and we denote by $e^1_{f-k+1}, \ldots, e^l_{f-k+1}$ the corresponding executions (in e^j_{f-k+1}, messages $m_1, \ldots, m_j$ are lost). We show, by induction on j, that $e_0 \sim e^j_{f-k+1}$.

Initially, we have seen that execution $e_{f-k} \sim e^0_{f-k+1}$. By the induction hypothesis, there is an execution $e^{j-1}_{f-k+1} \sim e^0_{f-k+1}$. Consider execution e^j_{f-k+1}

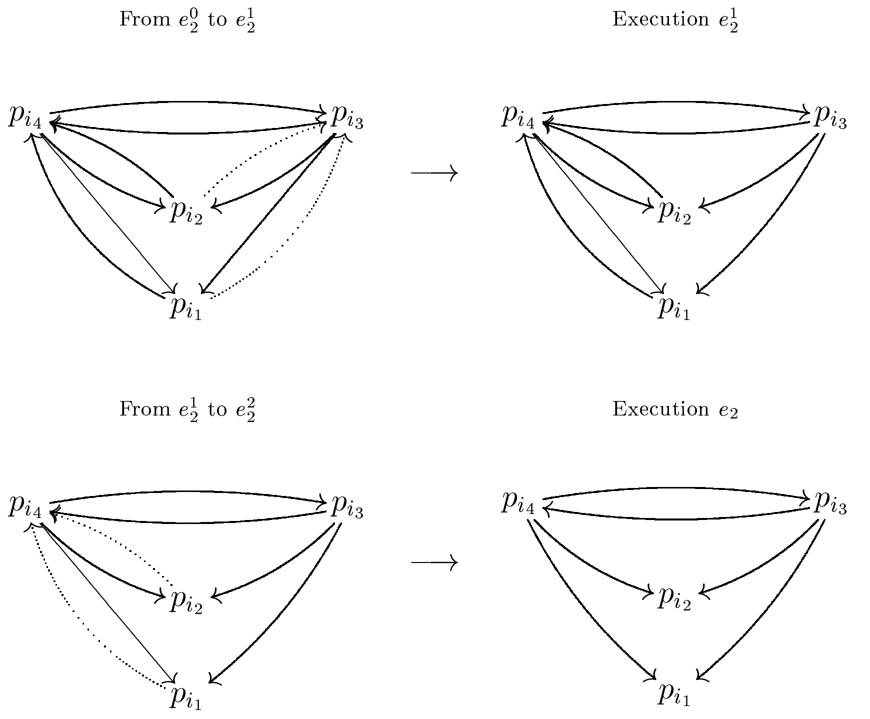


Fig. 3. Induction step for round 2 with $n = 4$ and $f = 3$

where messages $m_1, \dots, m_j$ are lost. Denote by $p_{i_{k+1}}$ the process that receives m_j in e^{j-1}_{f-k+1}. As $p_{i_{k+1}}$ does not send any message starting from round $k+1$, by construction, then e^{j}_{f-k+1} is indistinguishable from e^{j-1}_{f-k+1}. By the induction hypothesis, $e^{j}_{f-k+1} \sim e^{j-1}_{f-k+1} \sim e^{0}_{f-k+1}$, which implies in turn that $e^{j}_{f-k+1} \sim e^{0}_{f-k+1}$. For $j = l$, we have $e_{f-k} \sim e_{f-k+1}$. We conclude by considering execution $e_f = e$.

The example presented in Fig. 3 considers round 2 of executions e_1 and e_2 for the case where $n = 4$ and $f = 3$. Figure 3 illustrates the inductive step to connect e_1 to e_2. The two graphs at the top illustrate the connection from e_2^0 to e_2^1, by considering that process p_{i_3} fail by omitting all its messages in round 3. The two graphs at the bottom illustrate the connection from e_2^1 to e_2^2, by considering that process p_{i_4} fail by omitting all its messages in round 3. The resulting execution corresponds to e_2.

□

From Proposition 4.1 we immediately obtain the following corollary.

Corollary 4.3 *In an f-resilient omission model, no protocol can solve consensus in f rounds.*

4.2 Application to Consensus: Sufficiency

In this section, we show how one can reason about the solvability of consensus by using the properties on connected components previously observed. We establish a contradiction with the base assumption on our f-resilient omission model, which allows us to conclude that consensus is solvable in the f-resilient model in exactly $f + 1$ rounds.

Precisely, Lemma 3.1 gives a sufficient condition for a connected component to contain the omission-free execution, whereas Lemma 3.2 gives a characteristic of the connected component containing the omission-free execution. We deduce an upper bound on the number of rounds to solve consensus by using the two lemmas in a complementary way.

Theorem 4.4 *There is a protocol that solves consensus in an f-resilient omission model processes in $f + 1$ rounds.*

Proof. Assume no protocol solves consensus in $f + 1$ rounds. By Proposition 4.1, there exists a component that, for every process p_i, contains an execution in which p_i can change its mind. By Lemma 3.1, the component also contains the omission-free execution. Thus there is a path P that links the omission-free execution with an execution in which one process can change its initial value. By Lemma 3.2, P passes through an execution in which $f+1$ processes are faulty. This contradicts with the assumption that at most f processes can omit messages in any execution of our model. □

5 Acknowledgments

Eli Gafni, Maurice Herlihy and Sergio Rajsbaum introduced us to the application of topology to distributed computing.

References

[1] O. Biran, S. Moran, and S. Zaks. A combinatorial characterization of the distributed 1-solvable tasks. *Journal of Algorithms*, 11(3):420–440, September 1990.

[2] E. Borowsky and E. Gafni. Immediate atomic snapshots and fast renaming. In *Proceedings of the* 12^{th} *ACM Symposium on Principles of Distributed Computing*, pages 41–51, 1993.

[3] E. Borowsky and E. Gafni. A simple algorithmically reasoned characterization of wait-free computations (extended abstract). In *Proceedings of the* 16^{th} *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 189–198, 1997.

[4] 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.

[5] E. Gafni. Round-by-round fault detector–unifying synchrony and asynchrony. In *Proceedings of the* 17^{th} *ACM Symposium on Principles of Distributed Computing*, 1998.

[6] 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.

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

[8] Y. Moses and M. R. Tuttle. Programming simultaneous actions using common knowledge. *Algorithmica*, 3(1):121–169, 1988.

[9] J. R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, Reading MA, 1984.

Electronic Notes in Theoretical Computer Science 52 No. 2 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume52.html 11 pages

A Note on Set Agreement with Omission Failures[*]

Rachid Guerraoui, Petr Kouznetsov, Bastian Pochon

Distributed Programming Laboratory,
Swiss Federal Institute of Technology in Lausanne (EPFL)

Abstract

This paper considers the k-set agreement problem in a synchronous distributed system model with send-omission failures in which at most f processes can fail by send-omission. We show that, in a system of $n+1$ processes ($n+1 > f$), no algorithm can solve k-set agreement in $\lfloor \frac{f}{k} \rfloor$ rounds. Our lower bound proof uses topological techniques to characterize subsets of executions of our model. The characterization has a surprisingly regular structure which leads to a simple and succinct proof. We also show that the lower bound is tight by exhibiting a new algorithm that solves k-set agreement in $\lfloor \frac{f}{k} \rfloor + 1$ rounds.

1 Introduction

Context.

A generalization of the consensus problem [7], k-set agreement [4], consists of processes deciding on some final values based on their initial proposed values in such a way that: (1) the set of decided values contains at most k distinct values, (2) every decided value is a proposed value, and (3) every correct process eventually decides. The problem cannot be solved in a crash stop asynchronous model if the number f of processes that can crash is at least k [2,10]. This is a generalization of the FLP impossibility result [7] stating that consensus is not solvable if at least one process can crash: in this case, $k = 1$ and $f = 1$. It can be shown that in a synchronous model of $n + 1$ processes, where up to f processes can crash, k-set agreement requires exactly $\lfloor f/k \rfloor + 1$ rounds if $\lfloor \frac{f}{k} \rfloor k \leq n - k$, and exactly $\lfloor f/k \rfloor$ rounds if $\lfloor \frac{f}{k} \rfloor k > n - k$: this is a simple generalization of [5], where only the case $f \leq n - k$ was considered.

[*] This work is partially supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

Model.

In this paper, we consider a synchronous message-passing model with send-omission failures (we will simply say an omission model). In this model, processes proceed in a round-by-round manner: in each round, every process sends a message to all, receives messages from other processes, and updates its local state. The only failures allowed in our model are send-omission failures (we will simply say omissions): a process can send a message that is never received.

Contributions.

We show that in the omission model of $n+1$ processes, where at most f processes can fail by omission ($f \leq n$), k-set agreement cannot be solved in $\lfloor f/k \rfloor$ rounds. To prove this lower bound, we use the convenient notion of *pseudosphere* from [9] to describe the topological structure corresponding to a one-round execution of our model. In comparison with the proofs given in [5,9] on k-set agreement lower bounds in a synchronous model with crash failures, our result is much easier to derive. This is due to the observation that the protocol complex corresponding to a bounded number of rounds of our omission model has a very regular structure: it is a complex homeomorphic to a union of *n-dimensional* pseudospheres. As a result, the connectivity of the complex giving the lower bound for k-set agreement can be easily computed. We also present a new algorithm that solves the problem in $\lfloor f/k \rfloor + 1$ rounds. Thus, for any $f < n+1$, k-set agreement requires exactly $\lfloor f/k \rfloor + 1$ rounds of a synchronous model of $n+1$ processes with at most f processes that can fail by omission.

Roadmap.

Section 2 discusses the link between our result and known lower bounds on k-set agreement. Section 3 presents our model. Section 4 recalls some basic topological results used in this paper. Section 5 proves the lower bound. Section 6 proves that the lower bound is tight by giving an algorithm that matches it.

2 Related work

An execution of a synchronous model of $n+1$ processes with up to f crash failures can be viewed as an execution of our omission model: a crash failure is modeled in our case as a special case of an omission where, having failed by omission in a given round, a process fails by omitting all its messages in the subsequent rounds. In a synchronous model of $n+1$ processes, with up to f crash failures where $\lfloor \frac{f}{k} \rfloor k \leq n-k$, k-set agreement cannot be solved in $\lfloor f/k \rfloor$ rounds [5,9]. Hence, no algorithm can solve k-set agreement in our omission model where up to f processes can fail by omission (we will call these processes *unreliable* in order to disambiguate with the notion of faulty process in the crash-prone model) and $\lfloor \frac{f}{k} \rfloor \leq n-k$, using less than $\lfloor \frac{f}{k} \rfloor + 1$ rounds: this would otherwise contradict the lower bound of [5].

However, in the case where $\lfloor f/k \rfloor k > n-k$, the lower bound of $\lfloor f/k \rfloor + 1$ rounds does not hold anymore for the synchronous model with crash failures: one can easily derive an algorithm that solves the problem in exactly $\lfloor f/k \rfloor$ rounds. We show in this paper that the lower bound for the omission model holds for all $f < n+1$ (not only for $\lfloor \frac{f}{k} \rfloor k \leq n-k$). Thus, for the case where $n-k < \lfloor f/k \rfloor k < n+1$, our result does not follow from [5].

With respect to k-set agreement, a round-based asynchronous model of $n+1$ processes with the strong failure detector $\mathcal{S}$ of [3] is equivalent to our omission model with $f = n$ in following sense: whenever k-set agreement is solvable in one model, it is also solvable in the other model. Thus, the lower bound of $n+1$ rounds for consensus holds for this model too. Note that the lower bound for consensus in this model was obtained in [6] independently of our general proof for k-set agreement. On the other hand, in a synchronous model with at most n crash failures, consensus can be solved in n rounds. In this sense, our tight lower bound captures an interesting difference between the synchronous model with omission failures and the synchronous model with crash failures.

An alternative proof of our lower bound for the omission model was obtained in [8] by reduction of the first $\lfloor f/k \rfloor$ rounds of the model to the asynchronous *round-by-round failure detector* atomic snapshot shared memory model with at most k crash failures. The latter model is known to be too weak to solve k-set agreement [2] which implies that $\lfloor f/k \rfloor$ rounds of the omission model are not enough. The lower bound proof is based on two fundamental results in distributed computing: the impossibility of k-set agreement in the asynchronous model [2] and the atomic snapshot shared memory construction [1]. Neither of these is easy to derive. The proof we give in this paper is self-contained and simple: it is based on an interesting regularity of the omission model. Moreover, we show here that the lower bound is tight by presenting an optimal k-set agreement algorithm.

3 Model

The system we consider is a set of $n+1$ processes $\Pi = \{p_0, ..., p_n\} (n > 0)$. The processes evolve in synchronized rounds. In each round r, every process p_i executes the following steps: p_i sends a message to all other processes, receives a set of messages $M_{i,r}$ from other processes, and then updates its state.

We assume that all protocols we consider are *full-information* protocols where, in each round, every process sends its local state to all processes. The only failures allowed are (send) *omission* failures: messages sent by a process to a subset of other processes can be lost. It is known that no deterministic algorithm can solve k-set agreement in a model with omissions where, in *every* round, *some* k processes can fail by omission [2,12]. We assume here that at most f processes can fail by omission ($f < n+1$). As we pointed out in the introduction, we call these processes unreliable. By definition, in our model, every process is correct. Thus, when the k-set agreement problem is invoked in the omission model, every process, even an unreliable one, should eventually decide on some value according to the problem specification (recalled in the introduction).

4 Background

This section recalls some notions and results from basic algebraic topology (presented, for example in [11]) and some remarkable definitions and results from [9] that we use in this paper.

4.1 Simplexes and complexes

It is convenient to model a global state of a system of $n+1$ processes as an *n-dimensional simplex* $S^n = (s_0, ..., s_n)$, where $s_i = \langle p_i, v_i \rangle$ defines local state v_i of process p_i [10]. We say that the vertexes $s_0, ..., s_n$ *span* the simplex S^n. We say that a simplex T is a *face* of a simplex S if all vertexes of T are vertexes of S. A set of global states is modeled as a set of simplexes, closed under containment, called a *complex.*

4.2 Protocols

A *protocol* $\mathcal{P}$ is a subset of executions of our model. For any initial state represented as an n-simplex S, a *protocol complex* $\mathcal{P}(S)$ defines the set of final states reachable from them through the executions in $\mathcal{P}$. In other words, a set of vertexes $\langle p_{i_0}, v_{i_0} \rangle, ..., \langle p_{i_n}, v_{i_n} \rangle$ span a simplex in $\mathcal{P}(S)$ if and only if (1) S defines the initial state of $p_{i_0}, ..., p_{i_n}$, and (2) there is an execution in $\mathcal{P}$ in which $p_{i_0}, ..., p_{i_n}$ finish the protocol with states $v_{i_0}, ..., v_{i_n}$. For a set $\{S_i\}$ of possible initial states, $\mathcal{P}(\cup_i S_i)$ is defined as $\cup_i \mathcal{P}(S_i)$. If S^m is a face of S^n, then we define $\mathcal{P}(S^m)$ to be a subcomplex of $\mathcal{P}(S^n)$ corresponding to the executions in $\mathcal{P}$ in which only processes of S^m take steps and processes of $S^n \backslash S^m$ failed by omitting all their messages. For $m < n-f$, $\mathcal{P}(S^m) = \emptyset$, since in our model, there is no execution in which more than f processes fail by omissions.

For any two complexes $\mathcal{K}$ and $\mathcal{L}$, $\mathcal{P}(\mathcal{K} \cap \mathcal{L}) = \mathcal{P}(\mathcal{K}) \cap \mathcal{P}(\mathcal{L})$: any state of $\mathcal{P}(\mathcal{K} \cap \mathcal{L})$ belongs to both $\mathcal{P}(\mathcal{K})$ and $\mathcal{P}(\mathcal{L})$, any state from $\mathcal{P}(\mathcal{K}) \cap \mathcal{P}(\mathcal{L})$ defines the final states of processes originated from $\mathcal{K} \cap \mathcal{L}$ and, thus, belongs to $\mathcal{P}(\mathcal{K} \cap \mathcal{L})$.

We denote by $\mathcal{I}$ a complex corresponding to a set of possible initial configurations. Informally, a protocol $\mathcal{P}$ solves k-set agreement for $\mathcal{I}$ if there exists a map δ that carries each vertex of $\mathcal{P}(\mathcal{I})$ to a decision value in such a way that, for any $S^m = (\langle p_{i_0}, v_{i_0} \rangle, ..., \langle p_{i_m}, v_{i_m} \rangle) \in \mathcal{I}$ ($m \geq n-f$), we have $\delta(\mathcal{P}(S^m)) \subseteq \{v_{i_0}, ..., v_{i_m}\}$ and $|\delta(\mathcal{P}(S^m))| \leq k$. (The formal definition of a *solvable task* is given in [10].)

Thus, in order to show that k-set agreement is not solvable in r rounds, it is sufficient to find an r-round protocol $\mathcal{P}$ that cannot solve the problem for some $\mathcal{I}$. Such a protocol can be interpreted as a set of worst-case executions in which no decision can be taken.

4.3 Connectivity

Informally, a complex is said to be k-connected if it has no holes in dimension k or less. More precisely:

Definition 4.1 A complex $\mathcal{K}$ is *k-connected* if every continuous map of the k-sphere to $\mathcal{K}$ can be extended to a continuous map of the $(k+1)$-disk. By convention, a complex is (-1)-connected if it is non-empty, and every complex is k-connected for $k < -1$.

We will also use the following corollary to the Mayer-Vietoris sequence [11] that helps define the connectivity of the result of $\mathcal{P}$ applied to a union of complexes:

Theorem 4.2 *If $\mathcal{K}$ and $\mathcal{L}$ are k-connected complexes, and $\mathcal{K} \cap \mathcal{L}$ is $(k-1)$-connected, then $\mathcal{K} \cup \mathcal{L}$ is k-connected.*

4.4 Pseudospheres

To prove our lower bound, we use the notion of *pseudosphere* introduced in [9] as a convenient abstraction to describe the topological structure corresponding to a bounded number of rounds of our model. To make the paper self-contained, we recall the definition of [9] here:

Definition 4.3 Let $S^m = (s_0, ..., s_m)$ be a simplex and $U_0, ..., U_m$ be a sequence of finite sets. The *pseudosphere* $\psi(S^m; U_0, ..., U_m)$ is a complex defined as follows. Each vertex of $\psi(S^m; U_0, ..., U_m)$ is a pair $\langle s_i, u_i \rangle$, where s_i is a vertex of S^m and $u_i \in U_i$. Vertexes $\langle s_{i_0}, u_{i_0} \rangle, ..., \langle s_{i_l}, u_{i_l} \rangle$ define a simplex of $\psi(S^m; U_0, ..., U_m)$ if and only if all s_{i_j} $(0 \leq j \leq l)$ are distinct. If for all $0 \leq i \leq m$, $U_i = U$, the pseudosphere is written $\psi(S^m; U)$.

The following properties of pseudospheres follow from their definition:

(i) If $U_0, ..., U_m$ are singleton sets, then $\psi(S^m; U_0, ..., U_m) \cong S^m$.

(ii) $\psi(S^m; U_0, ..., U_m) \cap \psi(S^m; V_0, ..., V_m) \cong \psi(S^m; U_0 \cap V_0, ..., U_m \cap V_m)$.

(iii) If $U_i = \emptyset$, then $\psi(S^m; U_0, ..., U_m) \cong \psi(S^{m-1}; U_0, ..., \widehat{U_i}, ..., U_m)$, where circumflex means that U_i is omitted in the sequence $U_0, ..., U_m$.

4.5 Impossibility and connectivity

The following theorem, borrowed from [9], is based on Sperner's lemma [11]: it relates the connectivity of a protocol complex derived from a pseudosphere, with the impossibility of k-set agreement:

Theorem 4.4 *Let $\mathcal{P}$ be a protocol. If for every n-dimensional pseudosphere $\psi(p_0,...,p_n;V)$, where V is non-empty, $\mathcal{P}(\psi(p_0,...,p_n;V))$ is $(k-1)$-connected, and there are more than k possible input values, then $\mathcal{P}$ cannot solve k-set agreement.*

5 Lower bound

In this section we prove our lower bound by presenting a counter-example: a protocol $\mathcal{P}$, such that the corresponding complex satisfies the precondition of Theorem 4.4: for any pseudosphere $\psi(p_0,...,p_n;V)$, where V is non-empty, $\mathcal{P}(\psi(p_0,...,p_n;V))$ is $(k-1)$-connected. More precisely, we consider a set of executions in which, in every round, at most k processes are allowed to fail by omission. The corresponding protocol complex can be viewed as a union of n-dimensional pseudospheres which makes the reasoning about its connectivity very simple.

5.1 Connectivity theorem

The following generalization of Theorem 9 and Theorem 11 of [9] helps define the connectivity of a union of pseudospheres. The proof which basically reuses the arguments from [9] is given here to make the paper self-contained.

Theorem 5.1 *Let $\mathcal{P}$ be a protocol, S^m a simplex, and c a constant integer. Let for every face S^l of S^m, the protocol complex $\mathcal{P}(S^l)$ be $(l-c-1)$-connected. Then for every sequence of finite sets $\{A_{0_j}\}_{j=0}^m,...,\{A_{l_j}\}_{j=0}^m$, such that for any $j\in[0,m]$, $\bigcap\limits_{i=0}^{l} A_{i_j}\neq\emptyset$, the protocol complex*

$$\mathcal{P}\left(\bigcup_{i=0}^{l}\psi(S^m;A_{i_0},...,A_{i_m})\right)\ \textit{is } (m-c-1)\textit{-connected.} \qquad \text{(Eq. 1)}$$

Proof. Since for any sequence $V_0,...,V_l$ of singleton sets, $\psi(S^l;V_0,...,V_l)\cong S^l$, we notice that $\mathcal{P}(\psi(S^l;V_0,...,V_l))\cong\mathcal{P}(S^l)$ is $(l-c-1)$-connected.

(i) First, we prove that, for any m and any non-empty sets $U_0,...,U_m$, the protocol complex $\mathcal{P}(\psi(S^m;U_0,...,U_m))$ is $(m-c-1)$-connected. We introduce here the partial order on the sequences $U_0,...,U_m$: $(V_0,...,V_m)\prec(U_0,...,U_m)$ if and only if each $V_i\subseteq U_i$ and for some j, $V_j\subset U_j$. We proceed by induction on m. For $m=c$ and any sequence $U_0,...,U_m$, the protocol complex $\mathcal{P}(\psi(S^m;U_0,...,U_m))$ is non-empty and, by definition, (-1)-connected.

Now assume that the claim holds for all simplexes of dimension less than m ($m>c$). We proceed by induction on the partially-ordered sequences of sets $U_0,...,U_m$. For the case where $(U_0,...,U_m)$ are singletons, the claim follows from the theorem condition. Assume that the claim holds for all sequences smaller than $U_0, ..., U_m$ and there is an index i, such that $U_i = v \cup V_i$, where V_i is non-empty ($v \notin V_i$). $\mathcal{P}(\psi(S^m; U_0, ..., U_m))$ is the union of $\mathcal{K} = \mathcal{P}(\psi(S^m; U_0, ..., V_i, ..., U_m))$ and $\mathcal{L} = \mathcal{P}(\psi(S^m; U_0, ..., \{v\}, ..., U_m))$ which are both $(m-c-1)$-connected by the induction hypothesis. The intersection is:

$$\begin{aligned}\mathcal{K} \cap \mathcal{L} &= \mathcal{P}(\psi(S^m; U_0, ..., V_i \cap \{v\}, ..., U_m)) = \\ &= \mathcal{P}(\psi(S^m; U_0, ..., \emptyset, ..., U_m)) \cong \\ &\cong \mathcal{P}(\psi(S^{m-1}; U_0, ..., \widehat{\emptyset}, ..., U_m)).\end{aligned}$$

The argument of $\mathcal{P}$ in the last expression represents an $(m-1)$-dimensional pseudosphere which is $(m-c-2)$-connected by the induction hypothesis. By Theorem 4.2, $\mathcal{K} \cup \mathcal{L} = \mathcal{P}(\psi(S^m; U_0, ..., U_m))$ is $(m-c-1)$-connected.

(ii) Now we prove our theorem by induction on l. We show that for any $l \geq 0$ and any sequence of sets $\{A_{i_j}\}$ satisfying the condition of the theorem, Equation 1 is guaranteed. The case $l = 0$ follows directly from (i). Now assume that, for some $l > 0$,

$$\mathcal{K} = \mathcal{P}\left(\bigcup_{i=0}^{l-1} \psi(S^m; A_{i_0}, ..., A_{i_m})\right) \text{ is } (m-c-1)\text{-connected.} \qquad \text{(Eq. 2)}$$

By (i), $\mathcal{L} = \mathcal{P}(\psi(S^m; A_{l_0}, ..., A_{l_m}))$ is $(m-c-1)$-connected. The intersection is

$$\begin{aligned}\mathcal{K} \cap \mathcal{L} &= \mathcal{P}\left((\bigcup_{i=0}^{l-1} \psi(S^m; A_{i_0}, ..., A_{i_m})) \cap \psi(S^m; A_{l_0}, ..., A_{l_m})\right) = \\ &= \mathcal{P}\left(\bigcup_{i=0}^{l-1} \psi(S^m; A_{i_0} \cap A_{l_0}, ..., A_{i_m} \cap A_{l_m})\right).\end{aligned}$$

By the initial assumption (Equation 2), for any j, $\bigcap_{i=0}^{l-1} (A_{i_j} \cap A_{l_j}) = \bigcap_{i=0}^{l} A_{i_j} \neq \emptyset$. Thus by the induction hypothesis,

$$\mathcal{K} \cap \mathcal{L} = \mathcal{P}\left(\bigcup_{i=0}^{l-1} \psi(S^m; A_{i_0} \cap A_{l_0}, ..., A_{i_m} \cap A_{l_m})\right) \text{ is } (m-c-1)\text{-connected.}$$

By Theorem 4.2, $\mathcal{K} \cup \mathcal{L}$ is $(m-c-1)$-connected.

□

Considering an identity protocol gives

Corollary 5.2 $\bigcup_{i=0}^{l} \psi(S^m; A_{i_0}, ..., A_{i_m})$ *is* $(m-1)$*-connected.*

5.2 One round

Now we define the protocol complex $\mathcal{R}^1(S^l)$ corresponding to one round of execution of our model, starting from an initial configuration S^l, in which up to k processes can fail by omission.[1]

Lemma 5.3 *Let $S^l = (p_{i_0}, ..., p_{i_l})$ be a simplex. If $l \geq n - k$, then*

$$\mathcal{R}^1(S^l) \cong \bigcup_{|K| \leq k} \psi(S^l; 2^{K-\{p_{i_0}\}}, ..., 2^{K-\{p_{i_l}\}}). \qquad \text{(Eq. 3)}$$

If $l < n - k$, then $\mathcal{R}^1(S^l)$ is empty.

Proof. Consider first the case $l \geq n - k$. Each vertex of $\mathcal{R}^1(S^l)$ has the form $\langle p_i, M_i \rangle$, where $p_i \in S^l$ and M_i is the set of messages received by p_i in the first round. Consider a particular set of executions in which exactly a subset $K \subset \Pi$ failed by omission in the first round. Each process p_i receives all messages from $\Pi \backslash K$ and a subset of messages from $K - \{p_i\}$ (p_i always knows its own message). Thus we can map in a one-to-one manner each vertex $\langle p_i, M_i \rangle$ of our protocol complex to a vertex labeled with a value from $2^K - \{p_i\}$. All combinations of the form $\langle p_i, u_i \rangle$, where $p_i \in S^l$ and $u_i \in 2^{K-\{p_i\}}$, give us a pseudosphere $\psi(S^l; 2^{K-\{p_{i_0}\}}, ..., 2^{K-\{p_{i_l}\}})$. The union over all sets K, such that $|K| \leq k$ gives the characterization of Equation 3.

The case $l < n - k$ is trivial: by the initial assumption, at most k processes can fail by omission. Thus no execution in which less then $n + 1 - k$ processes participate exists in the protocol complex. □

Example. Figure 1 depicts a protocol complex $\mathcal{R}^1(S^n)$, where $n = 2$, $f = 1$ and $k = 1$, corresponding to one round of the omission model of 3 processes of which at most one can fail by omission. Each vertex of the protocol complex corresponding to a reachable *local* state of a process is defined by the process id and the set of messages received by this process in the first round. Since at least two processes never fail by omission, each process receives at least two messages in each round. Moreover, in every simplex of the protocol complex corresponding to a reachable *global* state of the system, all sets of received messages include at least two common elements and every process is aware of its own message. Geometrically, the complex of Figure 1 consists of four pyramids starring from the vertexes $p : \{p, q, r\}$, $q : \{p, q, r\}$, $r : \{p, q, r\}$ with the base quadrangles corresponding to all possible executions where pairs of processes (q, r), (p, r) and (p, q) can miss the message of, respectively, p, q and r. These pyramids are homeomorphic to pseudospheres of the type $\psi(\{p, q, r\}; 2^{K-\{p\}}, 2^{K-\{q\}}, 2^{K-\{r\}})$, where K is, respectively, $\{p\}$, $\{q\}$ and $\{r\}$.

By Lemma 5.3 and Corollary 5.2, $\mathcal{R}^1(S^l)$ is $(l-1)$-connected for all $l \geq n - k$. Since for all $l < n - k$, $\mathcal{R}^1(S^l)$ is (-2)-connected, we have:

[1] Naturally, we consider the case where $k \leq f$. Otherwise the protocol complex is trivial.

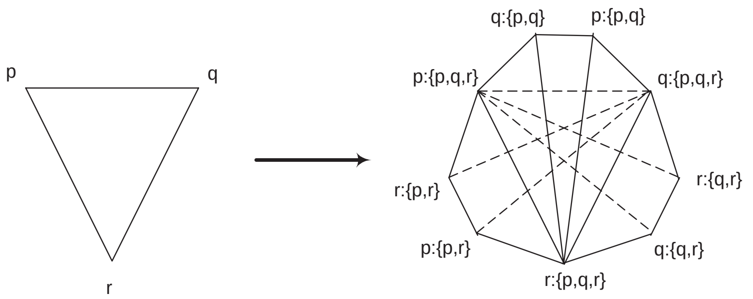


Fig. 1. One-round protocol complex for three processes and one unreliable process.

Lemma 5.4 *For all l, $\mathcal{R}^1(S^l)$ is $(l-(n-k)-1)$-connected.*

5.3 Multiple rounds

Now we are ready to derive our main result.

Theorem 5.5 *If $rk \le f$, then no algorithm can solve k-set agreement in r rounds.*

Proof. We apply Theorem 4.4 by showing that, for any non-empty set V and $rk \le f$, $\mathcal{R}^r(\psi(S^n; V))$ is $(k-1)$-connected. First, we prove that, for any m, $\mathcal{R}^r(S^m)$ is $(m-(n-k)-1)$-connected. Then we apply Theorem 5.1 showing that $\mathcal{R}^r(\psi(S^n; V))$ is $(k-1)$-connected.

We proceed by induction. The initial step ($r=1$) trivially follows from Lemma 5.4. Now assume that, for all m, $\mathcal{R}^{r-1}(S^m)$ is $(m-(n-k)-1)$-connected under the condition $rk \le f$. Thus,

$$\mathcal{R}^r(S^m) = \mathcal{R}^{r-1}(\mathcal{R}^1(S^m)) \cong \mathcal{R}^{r-1}\left(\bigcup_{|K|\le k} \psi(S^m; 2^{K-\{p_{i_0}\}}, ..., 2^{K-\{p_{i_m}\}})\right).$$

Since, for any $j \in [0,n]$, $\bigcap_{|K|\le k} 2^{K-\{p_{i_j}\}} = \{\emptyset\} \neq \emptyset$, by Theorem 5.1, $\mathcal{R}^r(S^m)$ is $(m-(n-k)-1)$-connected. □

6 Algorithm

Figure 2 presents an algorithm that matches our lower bound of Theorem 5.5. The algorithm solves k-set agreement in our model and guarantees that, in every execution, every process decides in round $\lfloor f/k \rfloor + 1$. The algorithm can be viewed as a generalization of the consensus algorithm of [3] defined for the asynchronous model augmented with the strong failure detector $\mathcal{S}$.[2] The idea

[2] The generalization is twofold: (1) we extend the algorithm from consensus to k-set agreement and (2) we make it f-resilient ($f \le n$) instead of n-resilient (for a system of $n+1$ processes) as in [3].

of our algorithm is the following:

(i) Each process p_i sets its decision estimate est_i to its initial proposal v_i.

(ii) In each round r from 1 to $\lfloor \frac{f}{k} \rfloor + 1$, every process p_i, such that $(r-1)k \leq i \leq rk - 1$, sends its current decision estimate to all.

(iii) Each process p_i receives the set $M_{i,r}$ of messages from other processes. If at least one estimate is received, then it is adopted by p_i.

(iv) Each process p_i decides its est_i after running $\lfloor \frac{f}{k} \rfloor + 1$ rounds.

```
$est_i := v_i$
for $r = 1..\lfloor \frac{f}{k} \rfloor + 1$ do
    if $i \in [(r-1)k, rk-1]$ then
        send $(i, est_i)$ to all processes
    receive $M_{i,r}$
    if $(\exists j)(\exists u)((j,u) \in M_{i,r})$ then
        $est_i := u$
decide $est_i$
```

Fig. 2. An algorithm for k-set agreement: process p_i.

Theorem 6.1 *The algorithm of Figure 2 solves k-set agreement in an omission model with $f < n+1$ unreliable processes.*

Proof. Every process decides after $\lfloor \frac{f}{k} \rfloor + 1$ rounds of computation. By the algorithm, the decided value is a proposed value of some process. Now we need to show that, in any execution, the set of decided values does not include more than k distinct values.

In any execution, there are in total $(\lfloor \frac{f}{k} \rfloor + 1)k$ distinct processes that broadcast their estimates. Since, $(\lfloor \frac{f}{k} \rfloor + 1)k > f$ and there are at most f unreliable processes in the system, there exist a round $r' \in [1, \lfloor \frac{f}{k} \rfloor + 1]$ and $p_j \in \Pi$, such that $\forall p_i \in \Pi : (j,u) \in M_{i,r'}$. By the algorithm, in each round, at most k processes broadcast their estimates. Thus, at most k distinct estimates can stay in the system after round r' and k-set agreement is guaranteed. □

7 Acknowledgements

Maurice Herlihy got us excited about the use of algebraic topology in distributed computing a year ago. Eli Gafni and Sergio Rajsbaum helped us control our excitement. Our colleagues from the Institute of Mathematics of EPFL, Sylvestre Blanc, Kathryn Hess, Paul-Eugene Parent and Orin Sauvageot, were instrumental in our understanding of simplexes and complexes.

References

[1] Y. Afek, H. Attiya, D. Dolev, E. Gafni, M. Merrit, and N. Shavit. Atomic snapshots of shared memory. *Journal of the ACM*, 40(4):873–890, September 1993.

[2] 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*, pages 91–100, May 1993.

[3] T. D. Chandra and S. Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267, March 1996.

[4] S. Chaudhuri. Agreement is harder than consensus: set consensus in totally asynchronous systems. In *Proceedings of the 9th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 311–324, August 1990.

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

[6] C. Delporte-Gallet and H. Fauconnier. Latency measures and lower bounds for consensus with failure detectors. In *Proceedings of the 9th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2002)*, Andros, 2002.

[7] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(3):374–382, April 1985.

[8] E. Gafni. Round-by-round fault detectors : Unifying synchrony and asynchrony. In *Proceedings of the 17th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 143–152, June 1998.

[9] M. Herlihy, S. Rajsbaum, and M. R. Tuttle. Unifying synchronous and asynchronous message-passing models. In *Proceedings of the 17th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 133–142, June 1998.

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

[11] J. R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, Menlo Park, Calif., 1984.

[12] N. Santoro and P. Widmayer. Time is not a healer. In *Proceedings of the 6th Annual Symp. Theor. Aspects of Computer Science*, volume 349 of LNCS, pages 304–313, Padernorn, Germany, February 1989. Springer Verlag.

Rewriting Systems and Hochschild-Mitchell Homology

Philippe Malbos

malbos@math.univ-montp2.fr
Laboratoire Géométrie-Topologie-Algèbre, UMR 5030
Université Montpellier II,
F-34095 Montpellier Cedex, France

Abstract

This paper is a part of a project focusing on the development of homological and simplicial methods in rewriting. The purpose of this contribution is to generalize Kobayashi's theorem for monoids to "monoids with several objects". Following Squier's theorem, Kobayashi constructs a resolution for monoids together with a presentation by complete rewriting systems. We construct a free resolution of a commutative ring k, viewed as a $k\mathbf{C}$-bimodule, where $\mathbf{C}$ is a small category. This resolution, associated to the Knuth-Bendix completion algorithm, reflects the combinatorial properties of $\mathbf{C}$. In particular, categories admitting finite complete presentations by graphs and relations have a finite type Hochschild-Mitchell homology.

MSC 2000. 18G10 - 68Q42 - 18A40 - 18G60 - 16S15 - 18D10.

Keywords. Projective resolutions - Rewriting systems - Kan extension - Hochschild-Mitchell homology - Finite presentation - Monoidal categories.

Preprint submitted to Elsevier Preprint *2 October 2002*

1 Introduction

Rewriting is a method to look at equational theories from a computational point of view. This method is related to representation theory of finite dimensional algebras. Oriented equations, called rewriting rules, equip equational theories with decision procedures [13]. Many problems of decision arising in equational theories, for instance, the fundamental word problem or the combinatorial enumeration problems, require specific presentations by rewriting systems. Both confluent and strongly normalizing presentations, called complete presentations, are methods to solve these problems in an algorithmic way. D. Knuth and P. Bendix [14] proved that there exists for every rewriting system an equivalent complete system which may be infinite. In general, equational theories do not have a presentation by a finite Complete Rewriting System (fCRS), even if the equational theory is finite.

Our purpose is two-fold. First, we would like to enlarge a categorical framework for rewriting on associative and unitary equational theories. The second goal is to construct homological finiteness conditions for presentations of such theories by complete rewriting systems. The aim of this paper is to study the case of small categories presented by graphs and relations that can be viewed as algebras in the monoidal category of graphs.

1.1 Rewriting on Algebras in Monoidal Categories

Rewriting occurs in many contexts, rewriting acts on words in algebraic structures (monoids, multisets, groups, algebras), on paths in oriented structures (small categories, trees, graphs) or on terms in terms rewriting systems. All these processes of rewriting lie in a common framework given by the notion of algebras in monoidal categories on which concepts of rewriting such as confluence or presentations by fCRS can be generalized. The word problem has a natural extension to algebras in monoidal categories defined by generators and relations. This framework allows us to generalize Squier's criterion on the rewriting on words in a monoid to the rewriting on words in an algebra into a monoidal category. We hope that there is a way to extend it to term rewriting systems. This categorical framework also unifies a variety of algebraic structures involved in computer science and includes various kinds of monads such as algebraic theories [12] and linear operads [5].

1.2 Homological Criterion

The second purpose is to construct projective resolutions to reflect the combinatorial properties of the rewriting systems. How can one detect the obstruction to the presentation by a fCRS ? In rewriting theory, only a few criteria giving necessary conditions to the existence of such presentations are known

[20]. Some of them are homological finiteness conditions. after the works of C.C. Squier [22], Y. Kobayashi [15], J.R.J Groves [9] and D.J. Anick [1], [2], it is known that properties such as confluence for the presentation of monoids by rewriting systems can be characterized in terms of resolutions of the involved monoid.

Using the Fox's differential calculus, Squier [22], has shown that if a monoid is defined by a fCRS then it is of type FP_3 over $\mathbb{Z}$. In particular, its third homology group is of finite type. In addition, he constructed a monoid which has a decidable word problem but cannot be of type FP_3. Consequently, a finititely presented monoid with a decidable word problem do not admit in general a fCRS presentation. Numerous generalizations of this result have been made. Kobayashi [15] proved, under the same hypotheses, that such monoids are of type FP_∞ over any commutative ring. He constructed recursively a free resolution of modules over the algebra of the monoid in which the chains are paths in the graph of the irreducibles. A general method, based on antichains, for constructing such resolutions was given by Anick, and applied both to associative augmented algebras over a field and to path algebras [1], [2].

The idea we want to develop in this paper is to construct a free resolution for small categories, using Kobayashi construction [15]. Our method is a way to extend the criterion of Squier and Kobayashi to algebras in monoidal categories. The resolution is constructed using the additive Kan extension of Anicks's antichains of the irreducibles. Let us note that Kan extensions can be thought as a way of extending the domain of a functor from "one object" to "several objects". Thus, we prove that if a small category has a presentation by a fCRS then its Hochschild-Mitchell homology is of finite type.

1.3 Conventions and notations

In the following, k is a commutative ring. A small k-category $\mathcal{C}$ is a small category provided with a structure of k-module on its hom sets and where the composition is k-bilinear. $\mathcal{C}^e$ will denote the enveloping k-category $\mathcal{C}^o \otimes_k \mathcal{C}$ and $\mathbf{Mod}(k)^{\mathcal{C}}$ the category of k-functors $M : \mathcal{C} \longrightarrow \mathbf{Mod}(k)$ called $\mathcal{C}$-modules. This category is isomorphic to the category of additive functors $\mathbf{Ab}^{\mathcal{C}}$. A $\mathcal{C}$-bimodule is a k-functor $M : \mathcal{C}^e \longrightarrow \mathbf{Mod}(k)$. If $\mathbf{C}$ is a small category, we will denote by $|\mathbf{C}|$ its set of object, by $\mathbf{C}_d$ the discrete category associated and by $\mathbf{C}^e = \mathbf{C}^o \times \mathbf{C}$ the enveloping category. The k-category $k\mathbf{C}$ is the category generated by $\mathbf{C}$ such that $|k\mathbf{C}| = |\mathbf{C}|$ and $k\mathbf{C}(p,q)$ is the k-module freely generated by $\mathbf{C}(p,q)$, $p, q \in |\mathbf{C}|$. For a morphism w in $\mathbf{C}$, we will denote $\sigma(w)$ the source of w and $\tau(w)$ its target.

2 Rewriting Systems for Small Categories

Small additive categories presented by graphs and relations and path algebras both give a way to compute projective resolutions of modules over rings [3], [2]. In algebra, the main interest of small additive categories and their quotients is that they are natural generalizations of rings [19]. There are applications of path algebras to the study of free algebras, group algebras, but the main interest lies in the representation of finite dimensional algebras through Morita theory. Another reason to study path algebras is their use in many contexts in computer science: process algebras, data structures like Giavitto's Group-Based Field [8] or reduction graphs in the diagrammatic presentation of the axiomatic rewriting systems [16], [18].

2.1 Notation and Rewriting Systems

For monoidal categories and algebras in monoidal categories we refer to [11] and [5]. Let Q be a directed graph, Q_0 and Q_1 will denote respectively the set of vertices and the set of arrows. Let us consider the slice category $\mathbb{Q} = \mathbf{Set}/Q_0 \times Q_0$, where objects are morphisms $X \to Q_0 \times Q_0$ in **Set** and where morphisms are commutative triangles of the obvious kind. For $Q : Q_1 \to Q_0 \times Q_0$ and $Q' : Q'_1 \to Q_0 \times Q_0$ in $|\mathbb{Q}|$, the product $Q \times_{Q_0} Q'$ is defined as the composition: $\pi \circ p^* : Q_1 \times_{Q_0} Q'_1 \to Q_0 \times Q_0$ where π is the first projection and p^* is given by the following pullback diagram:

$$\begin{tikzcd}
Q_1 \times_{Q_0} Q'_1 \arrow[r, "p^*"] \arrow[d] & Q_0 \times Q_0 \times Q_0 \arrow[d, "1\times\delta\times 1"] \\
Q_1 \times Q'_1 \arrow[r, "Q\times Q'"] & Q_0 \times Q_0 \times Q_0 \times Q_0.
\end{tikzcd}$$

The product $\times_{Q_0}$ endows the category $\mathbb{Q}$ with a structure of non-symmetric monoidal category where the unit is given by the diagonal $\delta : Q_0 \to Q_0 \times Q_0$. Let $\mathbf{Alg}(\mathbb{Q})$ be the category of algebras in $(\mathbb{Q}, \times_{Q_0}, \delta)$. The forgetful functor $U : \mathbf{Alg}(\mathbb{Q}) \longrightarrow \mathbb{Q}$ has a left adjoint L. For every directed graph $Q \in |\mathbb{Q}|$, $(L(Q), \circ, 1)$ is the free category generated by Q. $L(Q)$ is the morphism $L(Q_1) \to Q_0 \times Q_0$ where $L(Q_1) = \coprod_{n\in\mathbb{N}} Q_n$ and Q_n is the n-ith iterate : $Q_n = Q_{n-1} \times_{Q_0} Q_1$, $n \geq 2$. Multiplication $L(Q_1) \times_{Q_0} L(Q_1) \overset{\circ}{\to} L(Q_1)$ is given by pullback and the unit $Q_0 \overset{1}{\to} L(Q_1)$ is obvious. Let $(T = UF, \mu_Q, \eta_Q)$ be the monad on $\mathbb{Q}$ given by the adjunction above. For $Q \in |\mathbb{Q}|$ the free T-algebra $(T(Q), \mu_Q)$ is called the *path algebra* of Q, where μ_Q is the concatenation of paths. We will denote by λ the empty path. The slice category $\mathbb{Q}$ is cartesian with product given by pullback over $Q_0 \times Q_0$; thus, the category $\mathbb{Q}_T$ of free T-algebras on Q is also cartesian with product induced by $\times_{Q_0 \times Q_0}$.

A *rewriting system* $< Q \mid R >$ on $\mathbb{Q}$ consists of an object $Q \in |\mathbb{Q}|$ together with a subset $R \subset T(Q) \times_{Q_0 \times Q_0} T(Q)$. We denote by $\overset{R}{\to}$ the transitive reflexive closure of R, and by $\overset{R}{=}$ the congruence generated by R, i.e, if $w \overset{R}{=} w'$ then $\mu_Q(v, \mu_Q(w, v)) \overset{R}{=} \mu_Q(v, \mu_Q(w', v))$, for every $v, w \in T(Q)$. The small category presented by $< Q \mid R >$ is the quotient of the free T-algebra $(T(Q), \mu_Q)$ by the congruence $\overset{R}{=}$. A rewriting system $< Q \mid R >$ associated to a small category $\mathbf{C}$ is said to be *complete* (CRS) if the following two conditions hold:

i) R is *strongly normalizing*, that is, the relation $\overset{R}{\to}$ is well founded,

ii) R is *confluent*, that is, each congruence class of $\overset{R}{=}$, defining $\mathbf{C}$ as a quotient of $L(Q)$, contains exactly one element which represent the *normal form* of the elements of the class.

In order to fix notation, we recall some facts about rewriting system. A *critical pair* is a pair $(\alpha, \beta) \in \overset{R}{\to} \times \overset{R}{\to}$ describing one of the following situation:

i) inclusion ambiguities: $uwv \overset{\alpha}{\to} w'$ and $w \overset{\beta}{\to} w''$, $w \neq \lambda$,

ii) overlap ambiguities: $uw \overset{\alpha}{\to} w'$ and $wv \overset{\beta}{\to} w''$, $w \neq \lambda$.

A *resolution* of a critical pair (α, β) is a pair (α', β') such that $\tau(\alpha) = \sigma(\alpha')$ and $\tau(\beta) = \sigma(\beta')$ and $\tau(\alpha') = \tau(\beta')$. According to Knuth-Bendix theorem, a strongly normalizing rewriting system is confluent if and only if there exist a resolution for each its critical pair. Thus, if $< Q \mid R >$ is a complete, each path w in $L(Q)$ can be rewritten into a unique irreducible path, denoted by $\widehat{w}$, representing the same element in $\mathbf{C}$. Moreover, a CRS $< Q \mid R >$ is said to be a *finite complete rewriting system* (fCRS) if Q is a finite graph and R is finite.

2.2 Anick's Chains

Let $\mathbf{C}$ be a category presented by $< Q \mid R >$. We denote by $\leq$ the order on the paths defined by $u \leq w$ if $u = 1_{\sigma(w)}$ or $u = v_1 v_2 ... v_s$ and $w = v_1 v_2 ... v_n$ for some $1 \leq s \leq n$ and $v_i \in L(Q)$. The set $I = \{w \in L(Q) \mid w \text{ is irreducible}\}$ is an *order ideal of paths* pointed by source, i.e., $u \in I$ and $w \leq v$ implies $w \in I$. The set of *antichains* for I is defined by:

$$A_I = \{v \in L(Q) \mid v \notin I \text{ and } u \lneqq v \text{ imply } u \in I\}.$$

Let us note that the notion of antichains in the sense of [1] could be defined with another order ideal in order to adapt it to other processes of completion.

We denote by:

$$C_n = \{(v_1, ..., v_n) \in I^{\times_{Q_0} n} \mid v_1 \in Q_1,\ v_i v_{i+1} \in A_I\},$$

the set of n-chains for the rewriting system $< Q|R >$. We set by convention that $C_0 = Q_0$ and $C_1 = Q_1 \cap I$. For each $n \geq 0$ and $p, p' \in Q_0$, let $C_n(p, p')$ be the free abelian group generated by $C_n \cap \mathbf{C}(p, p')$. Thus C_n can be viewed as an additive functor from the discrete enveloping $\mathbb{Z}\mathbf{C}^e_d$ to $\mathbf{Ab}$.

We recall ([19], sec.6) that, if $\mathcal{C}$ is a small additive category, the functor $\otimes_{\mathcal{C}} : \mathbf{Ab}^{\mathcal{C}} \otimes_{\mathbb{Z}} \mathbf{Ab}^{\mathcal{C}^o} \longrightarrow \mathbf{Ab}$ is defined by $F \otimes_{\mathcal{C}} G = \bigoplus\limits_{p \in |\mathcal{C}|} F(p) \otimes_{\mathbb{Z}} G(p)/M$ where M is the abelian group generated by $< F(x)(u) \otimes v - u \otimes G(x)(v) \mid x \in \mathcal{C}(p, q), u \in F(p), v \in G(q) >$.

The module of *Anick's n-chains* of $< Q|R >$ is the left $\mathbb{Z}$-linear Kan extension of C_n along the inclusion functor $i : \mathbb{Z}\mathbf{C}^e_d \longrightarrow \mathbb{Z}\mathbf{C}^e$, i.e, the bifunctor:

$$\mathbb{Z}\mathbf{C}[C_n] = C_n \otimes_{\mathbb{Z}\mathbf{C}^e_d} \mathbb{Z}\mathbf{C}^e, \tag{1}$$

such that the following diagram commutes:

$$\begin{tikzcd}
\mathbb{Z}\mathbf{C}^e_d \arrow[rr, "i"] \arrow[dr, "C_n"'] & & \mathbb{Z}\mathbf{C}^e \arrow[dl, "\mathbb{Z}\mathbf{C}[C_n]"] \\
& \mathbf{Ab}. &
\end{tikzcd}$$

3 Finitely Generated Resolutions for Small Categories

3.1 Categories of type FP_n

Let $\mathcal{C}$ be a small k-category, we recall that $M \in \mathbf{Ab}^{\mathcal{C}}$ is *finitely generated* if it is the quotient of a finite coproduct of representables, or equivalently if the natural transformation $\bigoplus\limits_{i \in I} \mathcal{C}(p_i, \_) \longrightarrow M$ which sends 1_{p_i} to $|p_i| \in M(p_i)$ is an epimorphism in $\mathbf{Ab}^{\mathcal{C}}$. A resolution $P_\bullet$ in $\mathbf{Ab}^{\mathcal{C}}$ is said to be of *finite type* if each P_i is finitely generated. A $\mathcal{C}$-module $M \in \mathbf{Ab}^{\mathcal{C}}$ is said to be *of type FP_n*, $n \geq 0$, if there is a partial projective resolution of finite type $P_n \to \ldots \to P_0 \to M \to 0$ in $\mathbf{Ab}^{\mathcal{C}}$. By adapting the proof of [7], VIII prop. 4.3, we have:

Lemma 3.1. *Let M be a $\mathcal{C}$-module, and $n \geq 1$. The following conditions are equivalent:*

i) M is of type FP_n,

ii) M is finitely generated, and for every partial projective resolution of finite type $P_k \xrightarrow{\delta_k} P_{k-1} \to \ldots \to P_0 \to M \to 0$, $k < n$, the kernel $\ker \delta_k$ is finitely generated as $\mathcal{C}$-module.

Thus, a $\mathcal{C}$-module M is said to be *of type FP_∞* if it is of type FP_n for all integers $n \geq 0$.

A small category $\mathbf{C}$ is said to be of *type (right-left-)FP_n over k*, if $k\mathbf{C}$ is of type FP_n as $k\mathbf{C}$-bimodule. If $\mathbf{C}$ is of type FP_n over $\mathbb{Z}$ then, by tensoring by $k\mathbf{C}^e$, $\mathbf{C}$ is of type FP_n over any commutative ring k. We will say that $\mathbf{C}$ is of type FP_n if it is of type FP_n over $\mathbb{Z}$.

By the projection $\mathbf{C} \longrightarrow \mathbf{1}$, every small category $\mathbf{C}$ is of type FP_0. Let $\mathbf{C}$ be small category and M be a $\mathbf{C}$-bimodule. A *derivation* $d : \mathbf{C} \longrightarrow M$ is a family of applications: $d_{p,q} : \mathbf{C}(p,q) \longrightarrow M(p,q)$, $p,q \in |\mathbf{C}|$, satisfying:

$$d(xy) = d(x)y + yd(x), \qquad p \xrightarrow{y} q \xrightarrow{x} r \in \mathbf{C}.$$

More generally, for every path $(\alpha_1, \alpha_2, ..., \alpha_n)$ in $\mathbf{C}$ we have:

$$d(\alpha_1 ... \alpha_n) = \sum_{i=1}^{n} \alpha_1 ... \alpha_{i-1} d(\alpha_i) \alpha_{i+1} ... \alpha_n.$$

Let $\mu : \mathbb{Z}\mathbf{C} \otimes_{\mathbb{Z}Q_0} \mathbb{Z}\mathbf{C} \longrightarrow \mathbb{Z}\mathbf{C}$ be the augmentation: $\mu(w \otimes w') = ww'$. The *augmentation ideal* of $\mathbf{C}$ is the $\mathbf{C}$-bimodule $I_{\mathbf{C}}$ defined by the exact sequence:

$$0 \longrightarrow I_{\mathbf{C}} \longrightarrow \mathbb{Z}\mathbf{C} \otimes_{\mathbb{Z}Q_0} \mathbb{Z}\mathbf{C} \xrightarrow{\mu} \mathbb{Z}\mathbf{C} \longrightarrow 0. \tag{2}$$

The ideal $I_{\mathbf{C}}$ as $\mathbf{C}$-bimodule is generated by the set $\{1 \otimes w - w \otimes 1, \quad w \in \mathbf{C}\}$. Let $Q : Q_1 \to Q_0 \times Q_0$ be a directed graph, $\mathbf{C}$ be the category presented by the rewriting system $< Q|R >$ and $I_{<Q|R>}$ the augmentation ideal of $\mathbf{C}$. Then there exist an unique derivation $d : \mathbf{C} \longrightarrow I_{<Q|R>}$ such that: $d(f) = 1 \otimes f - f \otimes 1$, for every $f \in Q_1$. By the universal property of the derivation, we get that $I_{<Q|R>}$ is a free $L(Q)$-bimodule with basis $(d(f))_{f \in Q_1}$. Thus $\mathbf{C}$ is finitely generated if and only if $\mathbf{C}$ is FP_1. In the same way, we prove that $\mathbf{C}$ is FP_2 if and only if $\mathbf{C}$ is finitely presented. For $n \geq 2$, the FP_n condition constitutes a strengthening of the finite presentation reflecting combinatorial properties of $\mathbf{C}$.

3.2 Small Categories fCRS are FP_∞

Let $\mathbf{C}$ be a small category presented by a rewriting system $< Q|R >$, where $Q : Q_1 \times Q_1 \to Q_0$. The right-$\mathbf{C}^e$-module $\mathbb{Z}\mathbf{C}[C_n] = C_n \otimes_{\mathbb{Z}\mathbf{C}^e_d} \mathbb{Z}\mathbf{C}^e$ is equivalent to the $\mathbf{C}$-bimodule $\mathbb{Z}\mathbf{C} \otimes_{\mathbb{Z}Q_0} C_n \otimes_{\mathbb{Z}Q_0} \mathbb{Z}\mathbf{C}$ via $c_n \otimes (w^o \otimes w') = w^o \otimes c_n \otimes w'$. A generator $[c_n]$ in $\mathbb{Z}\mathbf{C}[C_n]$ will be denoted by $1_{\sigma(c_n)} \otimes c_n \otimes 1_{\tau(c_n)}$, and a n-chain

$$\xrightarrow{v_n}\xrightarrow{v_{n-1}}\rightarrow ... \rightarrow\xrightarrow{v_1},$$

will be denoted by $(v_1, ..., v_n)$.

For $n \geq 1$, let X_n be a set of paths of length n in $\mathbf{C}$. The relations R induce on $\mathbf{C}^o \times_{Q_0} X_n \times_{Q_0} \mathbf{C}$ a partial order defined by: $(\widehat{u}^o, v_1, ..., v_n, \widehat{w}) > (\widehat{u'}^o, v'_1, ..., v'_n, \widehat{w'})$ if $\widehat{u}^o v_1 ... v_n \widehat{w} \xrightarrow{R} \widehat{u'}^o v'_1 ... v'_n \widehat{w'}$. Let $\mathbb{Z}\mathbf{C}[X_n]$ be the $\mathbf{C}$-bimodule generated by X_n, and $X = \sum_{i=1}^{l} \alpha_i \widehat{u_i}^o \otimes x_i \otimes \widehat{w_i} \in \mathbb{Z}\mathbf{C}[X_n]$, $\alpha_i \in \mathbb{Z}$. The element $\widehat{u_1}^o \otimes x_1 \otimes \widehat{w_1}$ is said to be the *hight term* of X, and denoted by

$HT(X)$, if $l = 1$ or $\widehat{u_1}^o \otimes x_1 \otimes \widehat{w_1} > \widehat{u_i}^o \otimes x_i \otimes \widehat{w_i}$ for all $i \leq l$. R induces a partial order on $\coprod\limits_n \mathbb{Z}\mathbf{C}[X_n]$ defined by: for $X, Y \in \coprod\limits_n \mathbb{Z}\mathbf{C}[X_n]$, $X > Y$ if for $X = \sum_{i=1}^{l} \alpha_i \widehat{u_i}^o \otimes x_i \otimes \widehat{w_i}$, $\alpha_i \in \mathbb{Z}$, we have $\widehat{u_i}^o \otimes x_i \otimes \widehat{w_i} > HT(Y)$ for every $i \in 1, ..., l$. An exact sequence of $\mathbf{C}$-bimodules $(\mathbb{Z}\mathbf{C}[X_n], \delta_n)$ is said to be *nœtherian* if for every n, $\delta_n(\widehat{w}^o \otimes x_n \otimes \widehat{w'}) < \widehat{w}^o \otimes x_n \otimes \widehat{w'}$ for the order induced by R. Moreover,if there is a left contracting homotopy i_n such that $i_n(\widehat{w}^o \otimes x_n \otimes \widehat{w'}) < \widehat{w}^o \otimes x_n \otimes \widehat{w'}$ we say that $(\mathbb{Z}\mathbf{C}[X_n], \delta_n)$ is *nœtherian acyclic.*

Theorem 3.2. *Let $< Q|R >$ be a complete rewriting system for a small category $\mathbf{C}$. Then there exists a nœtherian acyclic resolution of $\mathbb{Z}\mathbf{C}$ by $\mathbf{C}$-bimodules:*

$$...Z\mathbf{C}[C_n] \overset{\delta_n}{\to} Z\mathbf{C}[C_{n-1}] \to ... \to \mathbb{Z}\mathbf{C}[C_1] \overset{\delta_1}{\to} \mathbb{Z}\mathbf{C}[Q_0] \overset{\mu}{\to} \mathbb{Z}\mathbf{C} \to 0,$$

where μ is the augmentation, $\mu([p]) = 1_p$, $p \in |\mathbf{C}|$, and

$$\delta_n([v_1, ..., v_n]) = 1 \otimes [v_1, ..., v_{n-1}] \otimes v_n - X, \quad (v_1, ..., v_n) \in C_n,$$

where $HT(X) < 1 \otimes [v_1, ..., v_{n-1}] \otimes v_n$, for the order induced by R.

Proof. As in [15], δ_n and the left $\mathbf{C}$-linear contracting homotopy i_n are constructed inductively.

First, the augmentation ideal $ker(\mu)$ is the $\mathbf{C}$-bimodule generated by the set $< d(v) \mid v \in C_1 >$. Let $\eta : \mathbb{Z}\mathbf{C} \longrightarrow \mathbb{Z}\mathbf{C}[Q_0]$ be the left-$\mathbf{C}$-linear morphism defined by $\eta(\widehat{w}) = \widehat{w} \otimes 1_{\tau(w)} \otimes 1$. We define the morphism of $\mathbf{C}$-bimodules δ_1 by

$$\delta_1([v]) = 1 \otimes [1_{\sigma(v)}] \otimes v - \eta\mu(1 \otimes [1_{\sigma(v)}] \otimes v).$$

The left-contracting homotopy $i_0 : \mathbb{Z}\mathbf{C}[Q_0] \longrightarrow \mathbb{Z}\mathbf{C}[Q_1]$ is defined inductively by:

$$\begin{cases} i_0(1 \otimes \lambda) = 0, \\ i_0(1 \otimes \widehat{w}w_i) = i_0(1 \otimes \widehat{w})w_i + \widehat{w}^o i_0(1 \otimes w_i), \quad \text{if } \widehat{w} \in |\mathbf{C}|,\ w_i \in I \text{ and } \widehat{w}w_i \in I. \end{cases}$$

It is easy to check that $\eta\mu + \delta_1 i_0 = 1_{\mathbb{Z}\mathbf{C}[Q_0]}$, $\delta_1(\widehat{w}^o \otimes v \otimes \widehat{w'}) < \widehat{w}^o \otimes v \otimes \widehat{w'}$ and $i_0(1 \otimes \widehat{w}) < \widehat{w}$.

In order to construct δ_2 from δ_1 and from i_0 as for δ_1, that is:

$$\delta_2(1 \otimes (v_1, v_2) \otimes \widehat{w}) = 1 \otimes v_1 \otimes v_2\widehat{w} - i_0\delta_1(1 \otimes v_1 \otimes v_2\widehat{w}), \quad (v_1, v_2) \in C_2,$$

we have to resolve the ambiguities in the construction of $i_2 : \mathbb{Z}\mathbf{C}[C_1] \longrightarrow \mathbb{Z}\mathbf{C}[C_2]$. Indeed, there is an ambiguity for $i_1(1 \otimes v_1 \otimes \widehat{w})$ since $v_1\widehat{w}$ can be either reducible or irreducible. If $v_1\widehat{w}$ is irreducible then we set: $i_1(1 \otimes v_1 \otimes \widehat{w}) = 0$, and thus $(i_0\delta_1 + \delta_2 i_1)(1 \otimes v_1 \otimes \widehat{w}) = i_0\delta_1(1 \otimes 1 \otimes v_1\widehat{w}) - i_0(\eta\mu(1 \otimes 1 \otimes v_1)\widehat{w}) = 1 \otimes v_1 \otimes \widehat{w}$.

Whereas if $v_1\widehat{w}$ is reducible, then there exist $v_2 \in I$ such that $v_1\widehat{w} = v_1v_2\widehat{w'}$ and $v_1v_2 \in A_I$. We can define i_1 inductively by

$$i_1(1 \otimes v_1 \otimes \widehat{w}) = 1 \otimes (v_1, v_2) \otimes \widehat{w'} + i_1(i_0\delta_1(1 \otimes v_1 \otimes v_2)\widehat{w'}).$$

Indeed $i_0\delta_1(1 \otimes v_1 \otimes v_2)\widehat{w'} = i_0(1 \otimes 1 \otimes v_1v_2)\widehat{w'} - i_0(v_1 \otimes 1 \otimes v_2)\widehat{w'}$, moreover $i_0(1 \otimes 1 \otimes v_1v_2)\widehat{w'} < 1 \otimes v_1 \otimes \widehat{w}$ and $i_0(v_1 \otimes 1 \otimes v_2)\widehat{w'} < 1 \otimes v_1 \otimes \widehat{w}$, thus $i_0\delta_1(1 \otimes v_1 \otimes v_2)\widehat{w} < 1 \otimes v_1 \otimes \widehat{w}$ and $i_1(i_0\delta_1(1 \otimes v_1 \otimes v_2)\widehat{w})$ is already defined.

The next steps of the resolution are constructed in the same way. Suppose now that $(\mathbb{Z}\mathbf{C}[C_k], \delta_k, i_k)_{0\leq k\leq n}$ is an acyclic and nœtherian resolution of $\mathbb{Z}\mathbf{C}$, satisfying, for $0 \leq k \leq n$:

$$(A_k) \begin{cases} i_{k-1}\delta_k(1 \otimes (v_1, ..., v_k) \otimes \widehat{w}) < 1 \otimes (v_1, ..., v_k) \otimes \widehat{w}, & \text{if } v_k\widehat{w} \text{ is reducible,} \\ i_{k-1}\delta_k(1 \otimes (v_1, ..., v_k) \otimes \widehat{w}) = 1 \otimes (v_1, ..., v_k) \otimes \widehat{w}, & \text{otherwise.} \end{cases}$$

Let δ_{n+1} be defined by:

$$\delta_{n+1}([v_1, ..., v_{n+1}]) = 1 \otimes v_1...v_n \otimes v_{n+1} - i_{n-1}\delta_n(1 \otimes v_1...v_n \otimes v_{n+1}).$$

By construction we have $\delta_n\delta_{n+1} = 0$. In order to construct a left contracting homotopy $i_n(1 \otimes (v_1, ..., v_n) \otimes \widehat{w})$ as we did for $n = 2$, we distinguish two cases depending of the irreducibility of $v_n\widehat{w}$.

If $v_n\widehat{w}$ is irreducible we set $i_n(1 \otimes (v_1, ..., v_n) \otimes \widehat{w}) = 0$. If $v_n\widehat{w}$ is reducible, then there exists a unique irreducible v_{n+1} such that $v_n\widehat{w} = v_nv_{n+1}w'$ and $(v_1, ..., v_n, v_{n+1}) \in C_{n+1}$. As v_nv_{n+1} is reducible, according to (A_n), we have $i_{n-1}\delta_n(1 \otimes (v_1, ..., v_n) \otimes \widehat{v_{n+1}}) < 1 \otimes (v_1, ..., v_n) \otimes \widehat{v_{n+1}}$ and consequently, by induction, $i_n(i_{n-1}\delta_n(1 \otimes (v_1, ..., v_n) \otimes \widehat{v_{n+1}})w')$ is defined. Thus we put

$$i_n(1\otimes(v_1, ..., v_n)\otimes\widehat{w}) = 1\otimes(v_1, ..., v_n, v_{n+1})\otimes w' + i_n(i_{n-1}\delta_n(1\otimes(v_1, ..., v_n)\otimes\widehat{v_{n+1}})w').$$

We consider also this two cases to check that $i_n\delta_{n+1} + i_{n-1}\delta_n = 1_{\mathbb{Z}\mathbf{C}[C_n]}$ and to show (A_{n+1}). □

If the rewriting system $< Q|R >$ is finite, then for every $n \geq 0$ the sets C_n are finite. Thus we have:

Theorem 3.3. *If a small category* $\mathbf{C}$ *has a presentation by a finite complete rewriting system* $< Q \mid R >$ *then* $\mathbf{C}$ *is of type* FP_∞ *over any commutative ring* k.

4 Hochschild-Mitchell Homology for Complete Categories

Let $\mathbf{C}$ be a small category, and M be a $\mathbf{C}$-bimodule. The cohomology $H^n(\mathbf{C}, M)$ and homology $H_n(\mathbf{C}, M)$ can be described using the functors *Ext* and *Tor* on the category $\mathbf{Ab}^{\mathbf{C}^e}$ which is abelian with enough projectives. The functor $Tor_n^{\mathbf{C}^e}$ is derived from the functor $\otimes_{\mathbf{C}^e}$: for every $\mathbf{C}$-bimodules F and G, $Tor_n^{\mathbf{C}^e}(F, G) = H_n(F \otimes_{\mathbf{C}^e} P_\bullet)$ where $P_\bullet$ is a projective resolution of G in $\mathbf{Ab}^{\mathbf{C}^e}$. In the same spirit, the functor $Ext^n_{\mathbf{C}^e}$ is derived from the functor $Hom_{\mathbf{C}^e}$: for every $\mathbf{C}$-bimodules F and G, $Ext^n_{\mathbf{C}^e}(F, G) = H^n(Hom_{\mathbf{C}^e}(P_\bullet, G))$ where $P_\bullet$ is a projective resolution of F in $\mathbf{Ab}^{\mathbf{C}^e}$. The k-category $k\mathbf{C}$ can be considered as a $k\mathbf{C}$-bimodule, the n-th Hochschild-Mitchell homology group of $\mathbf{C}$ ([19], sec. 12) is defined by:

$$H_n(\mathbf{C}, M) = Tor_n^{\mathbf{C}^e}(\mathbb{Z}\mathbf{C}, M).$$

See [4] for its relations with other homological theories.

If the presentation $< Q|R >$ is finite, for each $n \geq 0$ the set of n-chains is finite, and the complex obtained by tensoring by the resolution of Theorem 3.2 over $k\mathbf{C}^e$ is a finitely generated resolution of k. Thus we have:

Corollary 4.1. *If a small category* $\mathbf{C}$ *has a presentation by a fCRS* $< Q \mid R >$ *then the Hochschild-Mitchell homology groups* $H_n(\mathbf{C}, k)$ *are finitely generated.*

Several consequences can be deduced from the property FP_∞. In particular, let us consider $\Gamma(Q|R)$ the graph of irreducible of $< Q|R >$: the set of vertices is the set I of irreducibles, and the edges are pairs (v_1, v_2) such that $v_1 v_2 \in A_I$. Let us denote by $m(Q|R)$ the maximal length of a path in $\Gamma(Q|R)$ beginning in $I \cap Q_1$. Then, if $\mathbf{C}$ admits a fCRS presentation, its Hochschild-Mitchell dimension $\dim_k \mathbf{C} = Sup\{n \mid Ext^n_{k\mathbf{C}^e}(k\mathbf{C}, \_) \neq 0\}$ is finite and bounded by $m(Q|R)$.

5 Conclusion and Further Work

Following the framework defined in [5] for the (co)homology of algebras in monoidal categories we want to construct the same kind of resolution as in Theorem 3.2 for algebras $\mathcal{G}$ in a monoidal category $\mathbb{V}$, presented by rewriting systems. Moreover, $\mathbf{Alg}(\mathbb{V})$ can be viewed as the category of models of the algebraic theory of monoids $\mathbb{M}$ in the category $\mathbb{V}$, i.e, the category of functors $\mathbb{Q}^{\mathbb{M}}$ preserving finite products: we can also consider models of an arbitrary algebraic theory $\mathbb{A}$ in $\mathbb{V}$. In this setting, we have to work out a construction for the contracting homotopy involved in (3.2). We will then be able to construct a simplicial object in the category of coefficients of $\mathcal{G}$ reflecting the combinatorial properties of the involved rewriting system and this will lead to geometrical interpretations. We can also adapt the techniques of Anick's antichains to other procedures of completion than Knuth-Bendix's one. On the other hand, the constructive resolution used in the present paper gives a way to compute the (co)homology of a small category. For a finitely presented category, we have to implement the algorithm which perform such computation.

In addition, we believe that our construction can be adapted to monads over various kind of categories in order to obtain this condition for algebraic theories and term rewriting systems, which can be respectively modeled by finitary monads over the category **Set** and **Pre**, the category of preordered sets [12], [17]. M. Jiblaze and T. Pirashvili [21] define the (co)homology for algebraic theories $\mathbb{A}$ in terms of the Hochschild-Mitchell (co)homology, thus we hope that our construction gives a way to extend the finiteness condition *à la* Squier to terms rewriting systems.

References

[1] **D.J. Anick**. *On the homology of associative algebras*, Trans. Amer. Math. Soc., 296, 641-659, (1986).

[2] **D.J. Anick** and **E.L. Green**. *On the homology of quotients of path algebras*, Comm. in Alg., 15, 309-341, (1987).

[3] **M.J. Bardzell**. *Non-Commutative Groebner Bases and Hochschild Cohomology*, Contemp. Math., 286, Symbolic Computation: Solving Equations is Algebra, Geometry, and Engineering, 227-240, (2001).

[4] **H.-J. Baues** and **G. Wirsching**. *Cohomology of small categories*, J. of Pure and Appl. Alg., vol 38, 187-211, (1985).

[5] **H.-J. Baues**, **M. Jibladze** and **A. Tonks**. *Cohomology of Monoids in Monoidal Categories*, in "Operads: Proceedings of renaissance conferences", Contemp. Math. 202, 137-165, (1997).

[6] **R. Bieri**. *Homological Dimention of Disrete Groups*, Quenn Mary College Mathematics Notes, London, (1976).

[7] **K.S. Brown**. *Cohomology of Groups*, Graduate Texts in Mathematics, 87, Springer, (1982).

[8] **J.-L. Giavitto**, **O. Michel** and **J. Sansonnet**. *Group-based fields*, In Parallel Symbolic Languages and Systems, 1068, 209-215, (1996).

[9] **J.R.J. Groves**. *Rewriting Systems and Homology of Groups*, 3rd Int. Conf. Theory Groups Rel. Top., Canberra/Aust. 1989, Lect. Notes Math. 1456, 114-141 (1990).

[10] **J.R.J. Groves** and **G.C. Smith**. *Soluble Groups with a Finite Rewriting System*, Proceedings of the Edinburgh Mathematical Society, 36, 283-288, (1993).

[11] **G.M. Kelly**. *Basic concepts of enriched category theory*, London Math. Soc. Lecture Note Series 64, Cambridge Univ. Press, (1982).

[12] **G.M. Kelly** and **A.J. Power**. *Adjunctions whose counits are coequalizers and presentations of finitary monads*, J. of Pure and Appl. Alg., 89, 153-179, (1993).

[13] **J.W. Klop**. *Term Rewriting Systems*, In S. Abramsky, D. Gabbay, T. Maibaum, eds. Handbook of Logic in Computer Science, 2, 2-117, (1992).

[14] **D. Knuth** and **P. Bendix**. *Simple word problems in universal algebra*, in J. Leech ed., Computational problems in abstract algebra, Pergamon, Oxford

[15] **Y. Kobayashi**. *Complete rewriting systems and homology of monoid algebras*, J. of Pure and Appl. Alg., 65, 263-275, (1990).

[16] **J.-J. Lévy**. *Réductions correctes et optimales dans le λ-calcul*, Thèse de Doctorat d'Etat, Université Paris VII, (1978).

[17] **C. Lüth**. *Categorical Term Rewriting: Monads and Modularity*, PhD Thesis, University of Edinburgh, (1998).

[18] **P.-A. Melliès**. *Description abstraite des systèmes de réécritures*, Thèse de l'Université Paris VII, (1996).

[19] **B. Mitchell**. *Rings with Several Objects*, Adv. in Math. , 8, 1-161, (1972).

[20] **M. Nivat**. *Congruences Parfaites et Quasi-Parfaites*, Seminaire Dubreuil, 7, (1971)

[21] **M.A. Jibladze** and **T.I. Pirashvili**. *Cohomology of algebraics theories*, J. of Alg., 137, (2), 253-296, (1991).

[22] **C.C. Squier**. *Word problems and a homological finiteness condition for monoids*, J. of Pure and Appl. Alg., 49, 201-217, (1987).

Electronic Notes in Theoretical Computer Science 52 No. 2 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume52.html 20 pages

Geometric Aspects of Multiagent Systems

Timothy Porter

School of Informatics,
University of Wales Bangor,
Bangor,
Gwynedd, LL57 1UT,
Wales, U.K.
*Email:*tporter@bangor.ac.uk

Abstract

Recent advances in Multiagent Systems (MAS) and Epistemic Logic within Distributed Systems Theory, have used various combinatorial structures that model both the geometry of the systems and the Kripke model structure of models for the logic. Examining one of the simpler versions of these models, interpreted systems, and the related Kripke semantics of the logic $S5_n$ (an epistemic logic with n-agents), the similarities with the geometric / homotopy theoretic structure of groupoid atlases is striking. These latter objects arise in problems within algebraic K-theory, an area of algebra linked to the study of decomposition and normal form theorems in linear algebra. They have a natural well structured notion of path and constructions of path objects, etc., that yield a rich homotopy theory.

In this paper, we examine what an geometric analysis of the model may tell us of the MAS. Also the analogous notion of path will be analysed for interpreted systems and $S5_n$-Kripke models, and is compared to the notion of 'run' as used with MASs. Further progress may need adaptions to handle $S4_n$ rather than $S5_n$ and to use directed homotopy rather than standard 'reversible' homotopy.

Geometric Aspects of Multiagent Systems

Timothy Porter

1 Introduction

In many studies of distributed systems, a multiagent model is used. An agent is a processor, sensor or finite state machine, interconnected by a communication network with other 'agents'. Typically each agent has a local state that is a function of its initial state, the messages received from other agents, observations of the external environment and possible internal actions. It has become customary when using formal models of distributed systems to use modal epistemic logics as one of the tools for studying the knowledge of such systems. The basic such logic for handling a system with n-agents is one known as $S5_n$. Unless the system is very simple the actual logic will be an extension of that basic one, that is, it may have more axioms. For instance, the way the various agents are connected influences the logic in subtle ways. Suppose that agent 1 sends all its information immediately to agents 2 and 3, then if we denote by $K_i\phi$, the statement that agent i 'knows' proposition ϕ, we clearly expect within the logic of that system that $K_1\phi \Rightarrow K_2\phi \wedge K_3\phi$.

The logic $S5_n$ is obtained from ordinary propositional logic by adding 'knowledge operators', K_i as above. (In the literature the notation $K_i\phi$ is often replaced by $\Box_i\phi$.) It models a community of ideal knowledge agents who have the properties of veridical knowledge (everything they know is true), positive introspection (they know what they know) and negative introspection (they know what they do not know). These properties are reflected in the axiom system for the logic. The axioms include all propositional tautologies, plus the schemata of axioms: $K_i(\phi \Rightarrow \psi) \Rightarrow (K_i\phi \Rightarrow K_i\psi)$, $K_i\phi \Rightarrow \phi$, $K_i\phi \Rightarrow K_iK_i\phi$, and $\neg K_i\neg\phi \Rightarrow K_i\neg K_i\neg\phi$, where $i \in A$, the set of 'agents'. (We will see an alternative presentation of the logic later on.) Two comments worth making are (i) several of these axioms and in particular the last one - negative introspection - are considered computationally infeasible and (ii) ideas such as common knowledge (represented by an additional operator, C) can be introduced to give a richer extended language. Here however we will be restrict attention largely to models for $S5_n$ and extensions that may reflect the geometry of the system being modelled. How is this 'epistemic analysis used in practice? We mention three examples. One is given in [19] (§1.9) as due to Halpern and Zuck. It shows the way in which epistemic operators give compact and exact specifications of protocols that are verifiably 'safe'. Another worth mentioning is the analysis of AI data / knowledge searches, such as the Muddy Children problem (cf. Lomuscio and Ryan, [17]). Finally the study of knowledge based programming, [12], in which languages one may require statements such as : if i knows ϕ, set $x = 0$, by formalising what 'knows' means in this context requires analyses of this type. The book, [19], and several of the papers cited here contains numerous further examples.

The classical models for multimodal logics, and for $S5_n$ and its extensions in particular, are combinatorial models known as Kripke frames and, for $S5_n$, Kripke equivalence frames. These consist of a set W, called the *set of possible worlds*, and n-equivalence relations $\sim_i$, one for each agent. The interpretation of $\sim_i$ is that if w_1, w_2 are two possible worlds and $w_1 \sim_i w_2$, then agent i cannot tell these two worlds apart. In a series of papers and books (see in particular [6]) Fagin, Halpern, Moses and Vardi, in various combinations, have put forward a simpler combinatorial model known as an *interpreted system.* These have the same formal expressive power as Kripke frames, but are nearer the intuition of interacting agents than is the more abstract Kripke model.

In each case the underlying frame / set of global states, has a very similar combinatorial structure to that underlying a structure, *global actions* (or *groupoid atlases*), introduced by A. Bak, [1,2]. These arose from an analysis of algebraic problems related to the solution of systems of linear equations over arbitrary rings. (The mathematical area is *algebraic K-theory* and lies at the interface between algebraic topology and algebra / algebraic geometry). Any action of a group on a set leads to a set of orbits. These are the equivalence classes for an equivalence by a 'reachability' or 'accessibility' relation generated by the group action. (Translating and weakening to a monoid action, one has a variant of the reachability of states in a finite automaton.) In a global action, the set X is divided up into a family of patches, each of which has a group attached, which acts on that patch (see below for the more detailed definition). If the patches all coincide the resulting 'single domain global action' is essentially a set with a collection of (possibly independent/ possibly interacting) group actions. As group actions yield groupoids by a well known construction, and the resulting equivalence relations are also groupoids, a useful generalisation of global actions is that of groupoid atlas introduced by Bak, Brown, Minian and the author, [3]. These therefore present a context in which both the algebraic ideas and the logical models of $S5_n$ can fit. Moreover both global actions and groupoid atlases have a rich homotopy theory. This homotopy theory is based on a notion of path that, suitably modified, bears an uncanny resemblance to that of the 'runs' considered in multiagent systems, but seems to be better structured and, in fact, more computationally realistic.

The point of this paper is to examine these models in some more detail and to start the analysis of the necessary modifications to the global action/groupoid atlas homotopy theory that will allow its application to the problem of the geometric analysis of multiagent systems: how does the geometry of a multiagent system influence its inherent logic and thus its computational ability? The author's hope is that such an analysis will aid in three specific problem areas: firstly, any analysis of systems such as these hits the combinatorial explosion problem, the effective state space is too large for efficient search to be implemented. By reducing the search space via homotopical methods, it is expected that some progress in this can be achieved. Next, some distributed systems can be modularised thereby aiding verification that their description and behaviour matches their specification. This modularisable attribute should be identifiable by a combination of algebraic and geometric tools. A related question here is as to whether or not it is better to group a set of agents together as one 'super-agent' and under what condition can this be done without changing the behaviour of the system for the worse.[1] This, of course, presupposes a mechanism for comparison of MASs with different numbers of agents, a point to which we will return. The final hope is that a

[1] Our models of agents tend to work as if they are given 'atomic' entities, however if they interact or if they themselves consist of 'subagents', (processors), a different grouping into 'full agents' may be beneficial to analysis, optimisation and verification.

geometric homotopical overview may aid in the description and handling of *knowledge evolution* without MAS.

Acknowledgements

The research in this paper would not have been possible without conversations with Tony Bak, George Janelidze, and various participants at the 2000 category theory conference at the Villa Olmo, Como, and therefore without the assistance of ARC grant Number 859 and INTAS grants (INTAS 93-436 and 97-31961). It has also benefitted from conversations and discussions with Eric Goubault, Jean Goubault-Larrecq and other members of the Geometric Methods in Theoretical Computer Science group.

2 Preliminaries.

(As references for basic modal logic, try Kracht, [13], Meyer and van der Hoek, [19] and Blackburn, de Ryke and Venema, [4].) In the following, at least to start with, there will be n-agents and $A = \{1, 2, \ldots, n\}$ will denote the set of such 'agents'.

$S5$ and $S5_n$.

To introduce these logics fairly formally we suppose given a set of variables and form a language $\mathcal{L}_\omega(n)$ given by

$$\phi ::= p_\lambda \mid \perp \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid M_i\phi$$

where the p_λ are the propositional variable ordered by the finite ordinals λ, and M_i is a modality for each agent $i = 1, \ldots, n$.

In contrast to some treatments, we are using operators, M_i, corresponding to "possibility" , rather than "knowledge" operators, i.e. we interpret $M_i\phi$ as "agent i considers ϕ is possible". The relation with $K_i\phi$ ("agent i knows ϕ") is $M_i = \neg K_i \neg \phi$, "agent i does not know that ϕ is false". For computational purposes these may be expected to yield different methods, since $\neg$ is not well behaved computationally, however for this paper we will not be considering computational/ implementational problems, so the M v. K debate need not concern us greatly.

A logic in $\mathcal{L}_\omega(n)$ is any set Λ of $\mathcal{L}_\omega(n)$-formulae such that

- Λ includes all $\mathcal{L}_\omega(n)$-formulae that are instances of tautologies,

and

- Λ is closed under the inference rule

  if ϕ, $\phi \to \psi \in \Lambda$ then $\psi \in \Lambda$

  i.e. detachment or *modus ponens*

The logic is *uniform* if it is closed under the rule of uniform substitution of $\mathcal{L}_\omega(n)$-formulae for propositional variables and is *normal* if it contains the schemata

$(K)\quad M_i(\psi \vee \chi) \to M_i(\psi) \vee M_i(\chi)$
$(N)\quad \neg M_i(\bot)$

and monotonicity (for each i):

$$\text{if } \psi \to \chi \in \Lambda \text{ then } M_i\psi \to M_i\chi \in \Lambda.$$

As is well known, $S5_n$ is defined to be the smallest normal logic in $\mathcal{L}_\omega(n)$ containing

$(T)_i\quad \phi \to M_i\phi$

$(4)_i\quad M_iM_i\phi \to M_i\phi$

and

$(B)_i\quad \phi \to K_iM_i\phi,$

(so is $K4.BT$ or $S4.B$ in the notation used in Kracht, [13], p.72).

Of course $\phi \to \psi$ is shorthand for $\neg\phi \vee \psi$

As we are in the classical rather than the intuitionistic case, it is easy to rewrite this in terms of K_i instead of M_i. The logic $S5_1$ is usually called $S5$.

The related logic $S4_n$, mentioned earlier, does not require the schemata $(B)_i$.

The usual semantics of $S5_n$ is given by Kripke equivalence frames and models .

Kripke equivalence frames

An *equivalence frame* (or simply *frame*) $F = (W, \sim_1, \ldots, \sim_n)$ consists of a set W with, for each $i \in A$, an equivalence relation $\sim_i$ on W. Elements of W are called *worlds* and are denoted w, w', etc. We will write $[w]_i$ for the equivalence class of the element $w \in W$ for the i^{th} equivalence relation, $\sim_i$. A Kripke frame is very like a labelled transition system, but it has equivalence relations rather than partial orders as its basic relational structure. The logic gives a semi-static view of the system. To get a dynamic aspect one needs to look at knowledge evolution, cf. for example, Lomuscio and Ryan, [17].

An *equivalence (Kripke) model* (or simply *model*) $M = (F, \pi)$ is a frame F together with a relation

$$R_\pi \subseteq P \times W,$$

where $P = \{p_\lambda \ : \ \lambda \in \mathbb{N}\}$. This relation yields an *interpretation*

$$\pi_R : W \to \mathcal{P}(P),$$

which interprets as : $\pi_R(w)$ is the set of basic propositions "true" at w, or dually a *valuation*

$$_L\pi : P \to \mathcal{P}(W)$$

giving : $_L\pi(p)$ is the set of worlds at which p is "true". Of course π_R and $_L\pi$ contain the same information and will be merged in notation to π when no confusion will result.

A weak map or weak morphism of frames $f : F \to F' = (W', \sim'_1, \ldots, \sim'_n)$ is a function $f : W \to W'$ such that for each i,

$$\text{if } w \sim_i w', \text{ then } f(w) \sim'_i f(w').$$

The map f will give a map of models $f : (F, \pi) \to (F', \pi')$ if

$$\begin{tikzcd} W \arrow[rr, "f"] \arrow[dr, "\pi"'] & & W' \arrow[dl, "\pi'"] \\ & \mathcal{P}(P) & \end{tikzcd}$$

commutes.

Weak maps are too weak to react well with the logic so a stronger notion of bounded morphism (or p-morphism) is also used.

A weak morphism $f : F \to F'$ of frames is *bounded* if for each i, $1 \leq i \leq n$ and $u \in W$, $v' \in W'$,

$$f(u) \sim_i' v' \text{ if and only if there is a } v \in W \text{ with } f(v) = v' \text{ and } u \sim_i v.$$

Remark:

A discussion of some of the properties of the resulting categories of frames and weak maps (or of frames and bounded maps) can be found in [21]. In this paper we will not be considering bounded morphisms nor models in any great detail due to restrictions on space.

Global states for interpreted systems

Interpreted systems were first proposed by Fagin, Halpern, Moses and Vardi, [6] to model distributed systems. They give simple combinatorial models for some of the formal properties of multiagent systems. As before one has a set, $A = \{1, 2, \ldots, n\}$, of agents, and now one assumes each agent i can be in any state of a set L_i of local states. In addition one assumes given a set L_e of possible states of the 'environment'. More formally:

A *set of global states* (SGS) for an interpreted system is a subset S of the product $L_e \times L_1 \times \ldots \times L_n$ with each L_e, L_i non-empty. If $S = L_e \times L_1 \times \ldots \times L_n$, then the SGS is called a *hypercube*, cf. [14].

The idea behind allowing the possibility of considering a subset and not just the whole product is that some points in $\prod \underline{L} = L_e \times \prod_{i=1}^n L_i$ may not be 'feasible', because of explicit or implicit constraints present in the multiagent system (MAS). The explicit way these constraints might arise is usually not considered central for the general considerations of the multimodal logic approach to MASs, yet it seems clear that it represents the interconnection of the network of agents and, if the local states are the states of a finite state automaton, questions of reachability may also arise. This will be where the 'topology' of the MAS is most clearly influencing the combinatorial topology of the model. As a simple example, suppose we have agent 1 acts solely as a sensor for agent 2, so anything agent 1 knows, agent 2 automatically knows, $K_1\phi \Rightarrow K_2\phi$. The effect of this can be illustrated where L_1 has two local states, x_1 and x_2. In x_1, p is true; in x_2, $\neg p$ is true. Suppose L_2 has 5 local states, $y_1, \ldots, y_5$, and p is true only in y_1 and y_2, $\neg p$ being true in the remainder. Then $S = \{(x_1, y_1), (x_1, y_2), (x_2, y_3), (x_2, y_4), (x_2, y_5)\}$ is as large a SGS (or more precisely interpreted system as the valuation plays a role) as one can get within this setting. The situation mentioned earlier, $K_1\phi \Rightarrow K_2\phi \wedge K_3\phi$, will lead to a similar 3-dimensional example. The link between the structure of the SGS, the logic inherent in the interrelations between agents and the computational power of the system is subtle, see [16] for a set of examples. Other restrictions may also play a role. Agents may share resources, e.g. in a context where they need to access a distributed database and one agent may block another from an otherwise feasible transition.[2]

Any SGS yields a Kripke frame. If we write $\underline{L} = (L_e, L_1, \ldots, L_n)$ and $(S, \underline{L})$ for an SGS based on $\underline{L}$, then set $\mathcal{F}(S, \underline{L})$ to be the frame with S as its set of possible worlds with $\sim_i$ defined by:

$$\underline{\ell} \sim_i \underline{\ell}' \quad \Leftrightarrow \quad \ell_i = \ell'_i,$$

i.e. $\underline{\ell}$ and $\underline{\ell}'$ correspond to the same local state for agent i. For simplicity we will assume that L_e is a singleton set.

There are notions of weak map and bounded map of SGSs and an adjoint equivalence between the categories of frames and those of SGSs modulo a notion of essential equivalence. If $F = (W, \sim_1, \ldots, \sim_n)$ is an equivalence frame, then for each agent i, let $W_i = W/\sim_i$, be the set of equivalence classes of elements of W for the relation, $\sim_i$, and set $\underline{W} = (W_1, \ldots, W_n)$. There is a 'diagonal' function

$$\Delta : W \rightarrow \prod \underline{W},$$

given by

$$\Delta(w) = ([w]_1, \ldots, [w]_n)$$

and $(\Delta W, \underline{W})$ is an SGS. Setting $\mathcal{G}(F) = (\Delta W, \underline{W})$ gives the functor, left adjoint to $\mathcal{F}$, that is used in [21] to prove the equivalence mentioned above.

Mathematical Interlude: Global Actions and Groupoid Atlases.

A very similar structure to a Kripke equivalence frame is that of a global action, see [1,2]. Their generalisation in [3] to groupoid atlases gives a context where both Kripke equivalence frames and global actions coexist and it is a situation with a well defined and quite well behaved homotopy theory, therefore it yields a potential tool for the geometric analysis of MASs.

The prime example of a global action is a set X with a family of groups acting on it. In particular if G is a group (in the usual mathematical sense) then given a family of subgroups $\{H_i : i \in I\}$ of G, we can consider the actions

[2] For simplicity, it is assumed that each local agent is a reversible transition system. Thus if a transition $s \rightarrow s'$ can occur in L_i, then $s' \rightarrow^* s$ as well, i.e. we can get back from s' to s by some sequence of transitions, even if this requires reinitialising L_i, but any given transition in L_i may not be feasible at some state of S, being blocked by the actions of other agents, whence the complication of the system.

of each H_i on the set of elements of G by left multiplication. The important point to note is that the different subgroups H_i may be related, e.g. we may have $H_i \subseteq H_j$, which implies structural relationships between the equivalence relations generated by the actions of H_i and H_j. In more detail, we have for each $i \in I$, an equivalence relation $\sim_i$ on the set of elements of G defined by

$$x \sim_i y \text{ if and only if there is some } h_i \in H_i \text{ with } x = h_i y.$$

If $H_i \subseteq H_j$, then $x \sim_i y$ implies $x \sim_j y$, which is exactly the sort of relationship that results from 'knowledge passing' within a MAS, (cf. [15]). In a global action or groupoid atlas, this relationship is explicitly specified from the start. The example above is a single domain global action as there is one set on which all the groups act. The general form assumes only that the groups act on subsets of the 'domain'. This adds additional flexibility and adaptability to the concept. (In addition to the notes, [3], the original definition and discussion of global actions can be found in [1,2].)

A *global action* $\mathfrak{A}$ consists of a set $X_{\mathfrak{A}}$, together with a family $\{(G_{\mathfrak{A}})_\alpha \curvearrowright (X_{\mathfrak{A}})_\alpha \mid \alpha \in \Phi_{\mathfrak{A}}\}$ of group actions on subsets $(X_{\mathfrak{A}})\alpha \subseteq X_{\mathfrak{A}}$. The various *local groups* $(G_{\mathfrak{A}})_\alpha$ and the corresponding subsets (*local patches*), $(X_{\mathfrak{A}})_\alpha$, are indexed by the index set, $\Phi_{\mathfrak{A}}$, called the *coordinate system* of $\mathfrak{A}$. This set $\Phi_{\mathfrak{A}}$ is equipped with a reflexive relation, written $\leq$, and it is required that

- if $\alpha \leq \beta$ in $\Phi_{\mathfrak{A}}$, then $(G_{\mathfrak{A}})_\alpha$ leaves $(X_{\mathfrak{A}})_\alpha \cap (X_{\mathfrak{A}})_\beta$ invariant (so $(X_{\mathfrak{A}})_\alpha \cap (X_{\mathfrak{A}})_\beta$ is a union of equivalence classes for the $(G_{\mathfrak{A}})_\alpha$-action), and
- there is given for each pair $\alpha \leq \beta$, a group homomorphism

$$(G_{\mathfrak{A}})_{\alpha\leq\beta} : (G_{\mathfrak{A}})_\alpha \to (G_{\mathfrak{A}})_\beta$$

such that if $\sigma \in (G_{\mathfrak{A}})_\alpha$ and $x \in (X_{\mathfrak{A}})_\alpha \cap (X_{\mathfrak{A}})_\beta$, then

$$\sigma x = (G_{\mathfrak{A}})_{\alpha\leq\beta}(\sigma)x.$$

This second axiom says that if α and β are *explicitly* related and their domains intersect then the two actions are related on that intersection. Again this is the sort of structural compatibility that arises in MASs, except that, as so far considered, interpreted systems, etc. do not allow for the 'multi-patch' setting.

Any global action yields on each $(X_{\mathfrak{A}})_\alpha$ an equivalence relation due to the $(G_{\mathfrak{A}})_\alpha$-action. The equivalence classes (local orbits or local components) for these 'local equivalence relations' form a structure that is sometimes useful, regardless of what group actions are used, i.e. we need the local equivalence relations rather than the local groups that were used to derive them. As both equivalence relations and group actions yield groupoids (small categories in which all the morphisms are isomorphisms), it is convenient to adapt the notion of global actions to give a generalisation which handles the local equivalence relations as well. This generalisation is called a groupoid atlas in [3]. Fuller details of that transition are given in that source.

A *groupoid atlas* $\mathfrak{A}$ on a set $X_{\mathfrak{A}}$ consists of a family of groupoids $(G_{\mathfrak{A}})_\alpha$ defined with object sets $(X_{\mathfrak{A}})_\alpha$ which are subsets of $X_{\mathfrak{A}}$. These *local groupoids* are indexed by an index set $\Phi_{\mathfrak{A}}$, called the *coordinate system* of $\mathfrak{A}$, which is equipped with a reflexive relation, written $\leq$. This data is required to satisfy:
(i) if $\alpha \leq \beta$ in $\Phi_{\mathfrak{A}}$, then $(X_{\mathfrak{A}})_\alpha \cap (X_{\mathfrak{A}})_\beta$ is a union of components of $(G_{\mathfrak{A}})_\alpha$, i.e. if $x \in (X_{\mathfrak{A}})_\alpha \cap (X_{\mathfrak{A}})_\beta$, and $g \in (G_{\mathfrak{A}})_\alpha$, $g : x \to y$, then $y \in (X_{\mathfrak{A}})_\alpha \cap (X_{\mathfrak{A}})_\beta$; and
(ii) if $\alpha \leq \beta$ in $\Phi_{\mathfrak{A}}$, then there is given a groupoid morphism,

$$(G_{\mathfrak{A}})_\alpha\Big|_{(X_{\mathfrak{A}})_\alpha \cap (X_{\mathfrak{A}})_\beta} \to (G_{\mathfrak{A}})_\beta\Big|_{(X_{\mathfrak{A}})_\alpha \cap (X_{\mathfrak{A}})_\beta},$$

defined between the restrictions of the local groupoids to the intersections, and which is the identity on objects.

Example 1. (From Kripke frames to Groupoid Atlases.)
Let X be a set and $\sim_i$, $i = 1, 2, \ldots, n$, n equivalence relations on X. Then $F = (X, \sim_1, \ldots, \sim_n)$ is a Kripke frame, but also, if we specify the local groupoids

$$G_i := \text{ Objects } X, \text{ arrows } x_1 \to_i x_2 \text{ if and only if } x_1 \sim_i x_2,$$

and Φ to be discrete, i.e. " $\leq$ " = " $=$ ", we have a simple groupoid atlas, $\mathfrak{A}(F)$, (cf. example 2 §2 of [3]). In fact later we will introduce a second method for turning a frame into a groupoid atlas.

Example 2. The Line
The simplest non-trivial groupoid is $\mathcal{I}$. This is the groupoid corresponding to the Kripke frame $W = \{0, 1\}$, $\sim$ = the indiscrete / universal equivalence relation so $1 \sim 0$. (If the number of equivalence relations / agents is needed to be kept constant, then set $\sim_i \ = \ \sim$ for $i = 1, \ldots, n$. This is sometimes useful, but should not concern us too much for the moment.

The line, $\mathfrak{L}$, is obtained by placing infinitely many copies of $\mathcal{I}$ end to end, so

$|\mathfrak{L}| :=$ the set, $\mathbb{Z}$, of integers
$\Phi := \mathbb{Z} \cup \{\Leftrightarrow\infty\}$, where $\Leftrightarrow\infty \leq \Leftrightarrow\infty$, $\Leftrightarrow\infty < n$ for all $n \in \mathbb{Z}$ and $n \leq n$,

but that gives all related pairs.

What about models?
The above construction (Example 1 and later on its variant) gives us a way to think of Kripke frames and SGSs as groupoid atlases, but they do not directly consider the interpretations / valuations that are needed if Kripke models and interpreted systems are to be studied via that combinatorial gadgetry. Given a frame $F = (W, \sim_1, \ldots, \sim_n)$ and an interpretation

$$\pi : W \to \mathcal{P}(P),$$

we can get a bounded morphism of frames

$$\pi_* : F \to S_P^\Lambda$$

for $\Lambda = S5_n$ or an extension. This frame S_P^Λ is the *canonical frame* for the logic Λ with the given set of variables P. Its 'possible worlds' are the Λ-maximal sets of $\mathcal{L}_\omega(n)$-formulae, and π_* assigns to a world w the set of ϕ such that $(F, w) \models_\pi \phi$, i.e. the set of ϕ valid at the world w given π as interpretation, (cf. [13] p.63). Thus, if the homotopy theory of frames / SGSs informs us about their 'geometry', the homotopy theory of frames *over* S_P^Λ should inform us of the corresponding 'geometry' of the Λ-models in each context. Because of this and our relative ignorance of 'homotopy over' in *this context*, we will put models aside for this paper and concentrate on frames and SGSs.

3 Morphisms, Runs, Curves and Paths.

In the previous section, we have seen that groupoid atlases form a class of structures that encompasses Kripke equivalence frames as well as more general objects such as global actions. This by itself need not be useful. The general notion of a categorical model of a situation demands that serious attention be paid to the morphisms. We have seen that Kripke frames and interpreted systems have both weak morphisms and bounded morphisms available for use, the latter preserving more of the internal logic. We therefore need to consider morphisms of groupoid atlases. The payoff will be if the known function space structure on certain classes of morphisms between groupoid atlases (cf. [1,2]) can allow a similar structure to be made available for models of MASs.

A function $f : |\mathfrak{A}| \to |\mathfrak{B}|$ between the underlying object sets of two groupoid atlases is said to support the structure of a weak morphism if it preserves *local frames*, (the term comes from the original work on global actions and is not connected with the model theoretic meaning). Here a local frame in $\mathfrak{A}$ is a set $\{x_0, \ldots, x_p\}$ of objects in some connected component of some $(G_\mathfrak{A})_\alpha$, i.e. $\alpha \in \Phi$ and there are arrows $g_i : x_0 \to x_i$ in $(G_\mathfrak{A})_\alpha$ for $i = 1, 2, \ldots, p$. The function f preserves local frames if for $\{x_0, \ldots, x_p\}$ is a local frame in $\mathfrak{A}$ then $\{f(x_0), \ldots, f(x_p)\}$ is a local frame in $\mathfrak{B}$.

Any weak morphism of Kripke frames will give a weak morphism of the corresponding groupoid atlases, but not conversely since if $\mathfrak{A} = (W, \sim_1, \ldots, \sim_n)$ and $\mathfrak{B} = (W', \sim'_1, \ldots, \sim'_n)$, the notion of weak morphism of groupoid atlases allows $f : W \to W'$ to ignore which agents are involved, i.e. $\{w_0, \ldots, w_p\}$ is a local frame in $\mathfrak{A}$ if there is some agent i such that $w_0 \sim_i w_k$, $k = 1, \ldots, p$, so agent i considers these worlds equivalent; if f preserves this local frame, then there is some agent j such that $f(w_0), \ldots, f(w_p)$ are considered equivalent by that agent. Note however that agent j need not be the same agent as agent i, nor necessarily have the same position in the lists of agents if the sets of agents in the two cases are represented by disjoint lists. In fact the number of agents in the context of the two Kripke frames did not actually need to be the same for a weak morphism to exist between them. This added flexibility would seem to be essential when discussing modularisation, as mentioned above, but also for interacting MASs and the resulting interaction between the corresponding epistemic logics, however note these are *weak* morphisms so the link with the logic here is fairly weak. There is a notion of strong morphism of groupoid atlases, but this will not be strong enough either for the logic. In fact Bak's notion of stong morphisms in this particular context reduces to that of weak morphisms. The difference is that in a weak morphism the 'reasons', i.e. the elements g_i above, that a set of objects forms a local frame is not considered a part of the data of the morphism, with strong morphisms this data *is* recorded. In the groupoid atlases derived from Kripke frames, there is only one 'reason' possible. If it exists, it is unique! Thus the difference between the two types of morphism can be safely ignored for the moment. We hope to return to morphisms of groupoid atlases that correspond to bounded morphisms of Kripke frames and SGSs in a future paper.

Runs (cf. [19], p. 59).

A *run* in a Kripke model $M = (F, \pi)$ associated with a distributed system / MAS is simply a finite or infinite sequence of states $s^{(1)}, s^{(2)}, \ldots$, with $s^{(i)} \in S$, the set of possible worlds of F, here being thought of as being an SGS for an interpreted $(S, \underline{L})$.

A weakness in this definition is that no apparent restriction is put on adjacent states in a run. This thus ignores essential structure in the SGS, and any link between runs and morphisms is not immediately clear. Because of this, we will take the view that as formulated, this notion of 'run' is not quite adequate for the analysis of these systems. It needs refining, bringing it nearer to the mathematical notion, not just for æsthetic reasons but also because it does not do the job for which it was 'designed'! It does work well in some situations however. If we, for the moment, write $x \to x'$ to mean $x \sim_i x'$ for some i and then extend to the corresponding category (reflexive, transitive closure) to give $x \to^* y$, then (cf. again [19], p. 60), for hypercubes with more than one agent, any two states are related via $\to^*$.

Lemma 3.1
If M is a hypercube SGS associated to a distributed system with more than one agent, then given any states s, t in S, $s \to^* t$.

Proof

If $\underline{s} = (s_1, \ldots, s_n)$ and $t = (t_1, \ldots, t_n)$ then

$$(s_1, \ldots, s_n) \to (t_1, s_2, \ldots, s_n) \to (t_1, \ldots, t_n).$$

The first arrow comes from $\sim_i$, and $i \neq 1$, the second from $\sim_1$. ■

Of course, this argument may fail if $(S, \underline{L})$ is not a hypercube as simple examples show.

Proposition 3.2
If M is a Kripke frame or SGS, considered, as above, as a groupoid atlas, then any weak morphism

$$f : L \to M$$

for which $f(n) = f(0)$ for all $n < 0$, determines a run $s^{(i)} = f(i)$ in S ■

Often the form of the set of global states is not specified that precisely. Sometimes local transition functions are used so that the L_i are transformed into "local transition systems" with the actions involved being coupled with each other (cf. for instance, the $\mathcal{VSK}$ systems of Wooldridge and Lomuscio [18]). The feasible runs would seem in any case to be those for which $s^{(i+1)}$ is reachable from $s^{(i)}$ so there is some set of transitions in the various agents that leads from $s^{(i)}$ to $s^{(i+1)}$, or precisely:

a run $(s^{(k)})$ is *feasible* if for each $k = 1, 2, \ldots,$ $s^{(k)} \to^* s^{(k+1)}$.

Of course, hidden within this notion is a certain potential for concurrency. We do not specify in $(s^{(k)})$ how $s^{(k)}$ becomes $s^{(k+1)}$ except that by some set of local transitions within the state spaces of the different agents, components of $s^{(k)}$ have changed to become those of $s^{(k+1)}$ and at all times *the resulting intermediate list of local states is a valid one*, i.e. is a list of global states within S.

Looking at finer granularity, assume that $s^{(k)}$ and $s^{(k+1)}$ are not linked directly, i.e. $s^{(k)} \to^* s^{(k+1)}$ but it is not the case that $s^{(k)} \to s^{(k+1)}$.

If $\underline{s}$, $\underline{t}$ are two states in $(S, \underline{L})$, we will write

$$HC(\underline{s}, \underline{t}) = \{\underline{x} \in \prod \underline{L} : \text{ for each } i, 1 \leq i \leq n, x_i = s_i \text{ or } t_i\}$$

and say this is the *hypercube interval* between $\underline{s}$ and $\underline{t}$.

Proposition 3.3
Suppose $(s^{(k)})$ is a run in $\mathfrak{M} = (S, \underline{L})$. If $HC(s^{(k)}, s^{(k+1)}) \subset S$ for each k, then there is a morphism

$$f : \mathfrak{L} \to \mathfrak{M}$$

of groupoid atlases satisfying $f(n) = s^{(1)}$, $n \leq 2$

$$f(2k) = s^{(k)}, \quad k = 1, 2, \ldots$$

■

In other words, if at each stage, the hypercube interval between adjacent states of a run is contained in S, we can replace $(s^{(k)})$ by a curve. Within each hypercube interval, there are many possible concurrent paths between adjacent states of the run. We therefore have not only that a 'curve' can be given to represent the run, but the different representing curves are in some sense 'homotopic', i.e. equivalent via deformations (or interleaving equivalence). Of course, the condition is far from being necessary. If each $s^{(k)} \to^* s^{(k+1)}$, we could find a curve, but would not be able to specify it as closely. The intermediate 'odd' points of the curve can be given, up to interleaving equivalence, as in the earlier lemma.

The precise definition of a curve in a groupoid atlas is given as follows:

If $\mathfrak{A}$ is a groupoid atlas, a *(strong) curve* in $\mathfrak{A}$ is a (strong) morphism of groupoid atlases

$$f : \mathfrak{L} \to \mathfrak{A},$$

so for each n, one gets a $\beta \in \Phi_{\mathfrak{A}}$ and $f(i_m) : f(n) \to f(n+1)$ in $(\mathcal{G}_{\mathfrak{A}})_\beta$, where we have written $f(i_m)$ for $f_{\mathcal{G}}(i_m)$, where $i_m : m \to m+1$ in $(X_{\mathfrak{L}})_m$. N.B. the β and g_β are part of the specification of the strong curve. The corresponding weak notion of curve only asks for the existence of β and g_β, but does not specify them

A *(free) path* in $\mathfrak{A}$ will be a curve that stabilises to a constant value on both its left and right ends, i.e. it is an $f : \mathfrak{L} \to \mathfrak{A}$ such that there are integers $N^- \leq N^+$ with the property that

for all $n \leq N^-$, $f(n) = f(N^-)$,
for all $n \geq N^+$, $f(n) = f(N^+)$,

Of more use for modelling runs is the notion of a *based path* (i.e. when a basepoint / initial state is specified, but no final point is mentioned). Given a basepoint $a_0 \in \mathfrak{A}$, a based path in $(\mathfrak{A}, a_0)$ is a 'free' path that stabilises to a_0 on the left, i.e. $f(N^-) = a_0$. We can similarly define a based curve by requiring merely left stabilisation at a_0. Runs correspond to such curves in which $N^- = 1$.

4 Objects of Curves, and Paths

Within the interpreted systems approach to MASs, a set R of runs is often considered as a model (see, for example, [11] or [12]). The equivalence frame structure given to R may involve the local history of each processor / agent or merely the various 'points' visited at the same time; see the discussion in [19] p.39. The groupoid atlas viewpoint provides a local frame structure on R that is canonical, but, of course, that will need evaluating for its relevance to the problems of MASs.

Let $\mathfrak{A}$ be a groupoid atlas with coordinate system $\Phi_{\mathfrak{A}}$, underlying set $X_{\mathfrak{A}}$, etc, as before. We will write Curves($\mathfrak{A}$) for the set of curves in $\mathfrak{A}$.

If $f : \mathfrak{L} \to \mathfrak{A}$ is a curve in $\mathfrak{A}$, a function $\beta : |\mathfrak{L}| \to \Phi_{\mathfrak{A}}$ *frames* f if β is a function such that

(i) for $m \in |\mathfrak{L}|$, $f(m) \in (X_{\mathfrak{A}})_{\beta(m)}$;
(ii) for $m \in |\mathfrak{L}|$, there is a b in $\Phi_{\mathfrak{A}}$ with $b \geq \beta(m)$, $b \geq \beta(m+1)$ and a $f(i_m) : f(m) \to f(m+1)$ in $(\mathcal{G}_{\mathfrak{A}})_b$.

Remarks:

(a) The intuition is that the local set containing f in Curves($\mathfrak{A}$) will consist of curves passing through the same local sets $(X_{\mathfrak{A}})_\alpha$ in the same sequence.

The idea of a framing of f is that β picks out the local sets $(X_{\mathfrak{A}})_{\beta(m)}$ that receive $f(m)$. Condition (ii) then ensures that these choices are compatible with the requirement that f be a curve.

(b) We have used several times the groupoid atlas associated to a Kripke frame or SGS. The set Φ in that case was just the set of agents with the discrete order. This use of the discrete order is too simplistic in general as it hides the relationships between the agents. Mathematically this simple model breaks down first on considering framings, since the condition (ii) implies $b \geq \beta(m)$ and $b \geq \beta(m+1)$ so $\beta(m) = \beta(m+1)$ if the order is discrete, but then if β is to frame f, f must never have left a single equivalence class of the Kripke frame which was not the intention! To avoid this silly restriction, we can replace the set of agents by the finite non-empty subsets of that set.

Kripke frames to Groupoid Atlases revisited.

Suppose $F = (X, \sim_1, \dots, \sim_n)$ is a Kripke frame. Define a new groupoid atlas $\mathfrak{A}'(F)$ by :

$|X_{\mathfrak{A}'(F)}| = X$, the underlying set of F;

$\Phi_{\mathfrak{A}'(F)}$ = the set of non-empty subsets of A ordered by $\supseteq$, i.e. $\alpha \leq \beta$ if $\alpha \supseteq \beta$;

$|X_{\mathfrak{A}'(F)}|_\alpha = X$, for all $\alpha \in \Phi_{\mathfrak{A}'(F)}$

and

$\sim_\alpha = \bigcap\{\sim_i : i \in \alpha\}$, i.e. the equivalence relation

$$x \sim_\alpha y \Leftrightarrow \bigwedge_{i \in \alpha}(x \sim_i y)$$

Remark.

We can think of $\mathfrak{A}'(F)$ as a 'subdivision' of $\mathfrak{A}(F)$, rather like the barycentric subdivision of a simplicial complex, a construction to which it is very closely related. To any global action or groupoid atlas, one can assign two simplicial complexes; see Appendix. These encode valuable geometric information about the system and relate to the interaction of the different equivalence classes. (Fuller details can be found in [3].) Our subdivision above makes no significant change to the homotopy information encoded in the corresponding complexes.

This subdivision is just what is needed to encode runs in 'framings'. Logically, it seems to correspond to the enrichment of our language with 'group common knowledge' operators K_α, $\alpha \subseteq A$, or dually 'group possibility' operators M_α, where

$$K_\alpha\phi = \bigwedge_{i \in \alpha} K_i\phi, \text{ etc.}$$

Here it should be possible to adapt the 'subdivision' to reflect more closely the geometry of the distributed system. For instance, not all finite sets of agents might be included as there might be no direct link between certain of them. The clique complexes used in analyses of scheduling problems in distributed systems and in the theory of traces may be relevant here.

Now let $\mathfrak{A}$ be a general groupoid atlas and let $\mathfrak{A}^{\mathfrak{L}}$ be the following data for a groupoid atlas:

$|X_{\mathfrak{A}^{\mathfrak{L}}}| = \textsc{Curves}(\mathfrak{A})$;

$\Phi_{\mathfrak{A}^{\mathfrak{L}}} = \{\beta : |\mathfrak{L}| \to \Phi_{\mathfrak{A}} \mid \beta$ frames some curve f in $\mathfrak{A}\}$

For $\beta \in \Phi_{\mathfrak{A}^{\mathfrak{L}}}$,

$(X_{\mathfrak{A}^{\mathfrak{L}}})_\beta = \{f \in \textsc{Curves}(\mathfrak{A}) \mid \beta$ frames $f\}$

$(\mathcal{G}_{\mathfrak{A}^{\mathfrak{L}}})_\beta = \{(\sigma_m) \mid \text{source}(\sigma_m) \in (X_{\mathfrak{A}^{\mathfrak{L}}})_\beta, \sigma_m \in (\mathcal{G}_{\mathfrak{A}^{\mathfrak{L}}})_{\beta(m)}\}$

Note that it is easy to see that target(σ_m) is also in $(X_{\mathfrak{A}^{\mathfrak{L}}})_\beta$ in this situation (see lemma in section 4 of [3]).

Finally define

$$\beta \leq \beta' \Leftrightarrow \beta(m) \leq \beta'(m) \text{ for all } m \in |\mathfrak{L}|.$$

Proposition 4.1
With the above notation, $\mathfrak{A}^{\mathfrak{L}}$ is a groupoid atlas. If $\mathfrak{A}$ is a global action, then so is $\mathfrak{A}^{\mathfrak{L}}$. ■

It is natural to ask if $\mathfrak{A} = \mathfrak{A}'(F)$ for F a Kripke frame, is $\mathfrak{A}^{\mathfrak{L}}$ associated to some Kripke frame. In general the answer would seem to be no as there will be more than one local 'patch', $(X_{\mathfrak{A}^{\mathfrak{L}}})_\beta$, in this case and the index set is that of *all* framings. In fact this structure is not obviously in the MAS literature. Each framing β of a curve f in $\mathfrak{A}'(F)$ defines a sequence $(\beta(m))$ of finite non-empty subsets of the set of agents, satisfying the Kripke frame version of condition (ii) namely that if $m \in |\mathfrak{L}|$, there is a b with $b \geq \beta(m)$, $b \geq \beta(m+1)$ and $f(m) \sim_b f(m+1)$. In other words $b \subseteq \beta(m) \cap \beta(m+1)$ and $f(m) \sim_b f(m+1)$. For a given set of runs, the framings *may* reflect a possiblility of some modularisation as they indicate which agents are idle during the run. This raises an interesting problem of using the framings to optimise use of resources.

Each of the local groupoids in $\mathfrak{A}'(F)^{\mathfrak{L}}$ is an equivalence relation on that local patch. Given f, $f' \in (X_{\mathfrak{A}'(F)})_\beta$, so β frames both f and f', they will be equivalent if

$$f(m) \sim_{\beta(m)} f'(m)$$

for each m. These linked pairs together with the fact that $f(m) \sim_b f(m+1)$ and $f'(m) \sim_{b'} f'(m+1)$, for some $b, b' \subseteq \beta(m) \cap \beta(m+1)$ give a pattern rather like a ladder of linked 'squares'. This is more or less a 'homotopy' between f and f' .

Remark

Perhaps a framing can best be thought of as the sequence of those subsets of A involved in a computation at each instant. Thus a particular agent may be idle throughout a run if no framing of that run/curve involves that agent. Sometimes more than one agent is involved in a transition at a particular time step, so if, at time m, the corresponding set of agents is $\beta(m)$, this interprets as saying that the two global states $f(m)$ and $f(m+1)$ are $\beta(m)$-equivalent, i.e. $f(m) \sim_i f(m+1)$ for *all* the agents i in $\beta(m)$. (In a SGS, which we can imagine as a hypercube for simplicity, then, for example, if $n = 5$, $f(m) = (s_1, s_2, s_3, s_4, s_5)$, and $f(m) = (s_1, s_2, s_3, s'_4, s'_5)$ are $\{1,2,3\}$-equivalent.)

The notion of homotopy between based curves should correspond to that of a path in $\mathfrak{A}^{\mathfrak{L}}$, where we *do* need 'path' not curve so that it stabilises to the given two curves at the two 'ends' of that path. We will not explore this here due to lack of space. The basics of a general treatment of homotopy for groupoid atlases can be found in [3] and in more detail in [20] (from the point of view of a cylinder based theory as against a cocylinder theory as would be natural from the viewpoint we have explored here). Another extremely useful source for this type of theory is [10].

Cartesian closedness?

Clearly the object of paths as defined corresponds to a mapping object with domain $\mathfrak{L}$. This raises the important but difficult question of the cartesian closedness of the category of groupoid atlases and more importantly of the part of it corresponding to the Kripke frames. Bak has shown [1,2] that global actions do allow a function space construction that is well behaved on a large class of examples. A closely related construction occurs with equilogical spaces as defined by Scott, [23]. These are T_0-spaces together with an equivalence relation. Kripke equivalence frames for a 'single agent system' give equilogical spaces and equilogical spaces form a cartesian closed category. No analogues of equilogical spaces for systems of n-agents seem to have been developed. Similarly no analogues are known where different models have different numbers of 'agents'. Yet from a logical point of view and for an an adequate logical language to handle multiagent systems, some setting in which a cartesian closed category structure is available is clearly desirable.

5 Conclusions, Critique and Future Directions

In this paper, I have tried to examine some of the methodological links between the theory of global actions / groupoid atlases and the general context of combinatorial models for studying multiagent systems. Within the space available, no firm conclusions can be reached as to the potential usefulness of these links, but the possibility of a better structured object of runs in a distributed system has been shown that extends the Kripke frames of runs considered in the MAS literature.

What has not been done? It is clear that a more detailed examination of homotopy is required, especially with respect to its interpretation in terms of computation. The problem of cartesian closedness has been noted, but, deliberately, set aside due to lack of space and firm knowledge. The whole question of the relationship between these constructions and bounded morphisms (and thus with the logic) has also been set aside. It is conjectured that bounded morphisms will form a (subclass of the) class of 'fibrations' in the homotopy theory of this context, since, for a bounded morphism $f : A \to B$ of frames, for each i, the condition corresponds to being a fibration of groupoids.

Finally, but crucially, the computational infeasibility of $S5_n$ suggests that a separate study using $S4_n$, probably in an intuitionistic form, cf. [8,9], will be worth doing. This will presumably need a version of directed homotopy, but which variant of the many available, cf. [7], will best suit is not yet clear. (The ideas in [10] are also very relevant here.) Perhaps then, some deeper evaluation of how the geometry of a distributed system influences its inherent logic and thus its computational ability will become possible.

References

[1] A. Bak, *Global Actions: The algebraic counterpart of a topological space*, Uspeki Mat. Nauk., English Translation: Russian Math. Surveys, 525, (1997), 955 –996.

[2] A. Bak, *Topological Methods in Algebra*, in S. Caenepeel and A. Verschoren, eds., *Rings, Hopf Algebras and Brauer Groups*, number 197 in Lect. Notes in Pure and Applied Math, 1998, M. Dekker, New York.

[3] A. Bak, R. Brown, G. Minian, and T. Porter, *Global Actions, Groupoid Atlases and Related Topics*, Draft lecture notes, Bangor, 2002.

[4] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, Cambridge Tracts in Theoretical Computer Science, Cambridge U.P., 2001.

[5] C. H. Dowker, Homology Groups of Relations, Annals of Maths. 56 (1952) 84 - 95.

[6] R. Fagin, J. Y. Halpern, Y. Moses and M.Y. Vardi, *Reasoning about Knowledge*, MIT Press, 1995.

[7] E. Goubault,

Some Geometric Perspectives in Concurrency Theory, Submitted (2001) (Available from http://www.di.ens.fr/ goubault/GOUBAULTpapers.html).

[8] J. Goubault-Larrecq and E. Goubault, Order theoretic, geometric and combinatorial models fof intuitionistic S4 proofs, in IMLA'99, 1999.

[9] J. Goubault-Larrecq and E. Goubault, On the geometry of Intuitionistic S4 proofs, submitted to Homotopy, Homology and its Applications.

[10] M. Grandis, An intrinsic homotopy theory for simplicial complexes, with applications to image analysis, Appl. Categ. Structures 10 (2002), 99-155.

[11] J. Y. Halpern and Y. Moses, Knowledge and Common knowledge in a Distributed Environment, J. A.C.M., 37(3) (1990) 549-587.

[12] J. Y. Halpern and Y. Moses, Using counterfactuals in Knowledge-based programming, in: Theoretical Aspects of Rationality and Knowledge, 1999, ed. Y. Gilboa.

[13] M. Kracht, *Tools and Techniques in Modal Logic*, Studies in Logic No. 142, Elsevier, 1999.

[14] A. Lomuscio and M. Ryan, On the relation between Interpreted Systems and Kripke Models, *Proceedings of the AI97 Workshop on Theoretical and Practical Foundation of Intelligent Agents and Agent-Oriented Systems*, Perth (WA), December. Springer lecture notes in AI vol. 1441.

[15] A. Lomuscio and M. Ryan, Ideal Agents Sharing (some!) knowledge, in Proc. ECAI-98. *13^{th} European Conference on Artificial Intelligence*, John Wiley & Sons, 5pp, 1998.

[16] A. Lomuscio and M. Ryan, A spectrum of modes of knowledge sharing between agents, Proceedings of the sixth international workshop on Agent Theories, Architectures, and Languages (ATAL-99), Springer LNAI © volume 1757.

[17] A. Lomuscio and M. D. Ryan, An algorithmic approach to knowledge evolution, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, Vol. 13, No. 2 (Special issue on Temporal Logic in Engineering), 23pp, 1999.

[18] A. Lomuscio and M. Wooldridge, Multi-Agent VSK Logic, Proceedings of JELIA2000, 7th European Workshop on Logic in Artificial Intelligence. Malaga. October. Springer LNAI © volume 1919.

[19] J.-J. Meyer and W. van der Hoek, *Epistemic Logic for AI and Computer Science*, Number 41 in Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1995.

[20] E. G. Minian, Generalized cofibration categories and global actions, K-Theory 20 (2000) 37 - 95.

[21] T. Porter, Interpreted systems and Kripke models for multiagents systems from a categorical perspective, preprint, UWB, School of Informatics, 2002.

[22] T. Porter, Categorical perspectives on multiagent systems, Research Report, UWB, School of Informatics, in preparation.

[23] D. S. Scott, A new category? Domains, Spaces and Equivalence Relations, Unpublished Manuscript, available http://www.cs.cmu.edu/Groups/LTC/.

Appendix : Simplicial complexes from Kripke frames

The two constructions mentioned in the main text are classical, dating in their initial forms to the embryonic algebraic topology of the 1920s and 30s. The local equivalence classes of a Kripke frame give a covering of the underlying set, X of the frame. It thus gives a relation from X to the set Y of equivalence classes. Abstracting, let $R \subset X \times Y$ be a relation. (In our case xRy is exactly $x \in y$, where y is an equivalence class for any of the equivalence relations). Using a formulation due to Dowker, [5], any such relation determines two simplicial complexes

(i) $K = K_R$:
- the set of vertices is the set, Y;
- p-simplex of K is a set $\{y_0, \cdots, y_p\} \subseteq Y$ such that there is some $x \in X$ with xRy_j for $j = 0, 1, \ \cdots, \ p$.

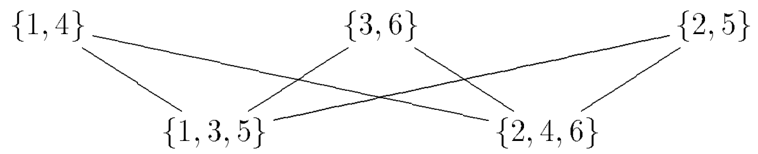
(ii) $L = L_R$:
- the set of vertices is the set X;
- a p-simplex of K is a set $\{x_0, \cdots, x_p\} \subseteq X$ such that there is some $y \in Y$ with x_iRy for $i = 0, 1, \ \cdots, \ p$.

These are, in some sense, dual constructions. In the topological context, K_R is often called the *Nerve* of the covering and L_R the *Vietoris complex.*

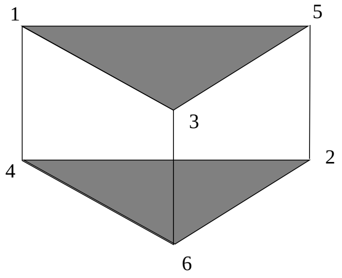
As a simple example, let $X = \{1, 2, \ldots, 6\}$,
$a \sim_1 b$ if $a \Leftrightarrow b$ is a multiple of 2;
$a \sim_2 b$ if $a \Leftrightarrow b$ is a multiple of 3.

This corresponds to a hypercube, $L_1 \times L_2$, with L_1 having 3 elements, L_2 having 2. Y has 5 elements. X has 6.

K_R is a bipartite graph:



L_R is a prism with two filled triangular faces:



They have both the homotopy type of a figure 8. (For instance in the prism, shrink the triangles to points and then shrink one vertical edge.)

The main result of Dowker's paper was that for an arbitrary relation R, the two complexes have the same homotopy type. The question of the influence of the homotopy type of these complexes on the complexity of searches in the state space of the original MAS seems to be a very interesting one.

Electronic Notes in Theoretical Computer Science 52 No. 2 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume52.html 15 pages

The Geometry of Timed PV Programs

Ulrich Fahrenberg

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

Abstract

We introduce a real-time extension of the PV language: A timed PV program consists of a number of timed automata which synchronize by locking and releasing common resources. We give a geometric semantics to such programs in terms of local po-spaces, and we lay out the foundations for making the established geometric techniques available for detecting deadlocks and unsafe configurations in timed PV programs.

1 Introduction

The PV formal language was introduced by Dijkstra in [3] and has since been applied in various areas of verification. Already in [3], a geometric understanding of PV programs is developed, in terms of *progress graphs*. In [6], these ideas are pursued further to develop an algorithm which is *geometric* in spirit, to detect deadlocks and unsafe states in simple PV programs without loops and branching.

The case of PV programs with loops is treated in [5] and [7], and in [5] it is noted that treating branching is easier than treating looping, hence the geometric techniques are applicable to the full calculus of untimed PV programs.

Dating back to [3], a PV process is commonly defined to be a regular expression on an alphabet $\{\mathtt{P}_a, \mathtt{V}_a \mid a \in \mathcal{O}\}$, subject to certain restrictions. Here $\mathcal{O}$ is a finite set of resources which can be locked (P) or released (V) by the processes, and a PV program then consists of a number of PV processes which synchronize by locking and releasing the common resources.

In this article we consider finite PV *automata* rather than PV processes, i.e. finite automata on the alphabet $\{\tau, \mathtt{P}_a, \mathtt{V}_a \mid a \in \mathcal{O}\}$. This makes the transition to geometric objects much more simple than in [7], and it also enables us to introduce time into the PV formalism, by passing from finite automata to *timed* automata.

After a review of the geometric realization technique for untimed PV programs from [5] and [7], rewritten to treat PV automata instead of PV processes, in section 2, we introduce our timed PV formalism in sections 3 and 4. In section 5 we define a geometric realization mapping from timed PV programs to local po-spaces, and in section 6 we elaborate on how this geometric realization technique could be applied to yield results similar to the ones of [5,6,7].

This article is based on the author's Master's thesis [4]; note however that notation and terminology have been changed slightly.

2 Untimed PV Programs

Throughout this article, we fix a finite set $\mathcal{O}$ of resources, which can be locked (P) and released (V) by the PV automata in question, and a semaphoricity mapping $s : \mathcal{O} \to \mathbb{N}_+$. We also let $\Sigma = \{\tau, \mathtt{P}_a, \mathtt{V}_a \mid a \in \mathcal{O}\}$.

2.1 PV Automata

A PV automaton is a finite automaton $P = (Q_P, q^0_P, E_P, f_P, L_P)$ on the alphabet Σ. Here Q_P is a finite set of locations, $q^0_P \in Q_P$ is the initial location, E_P is a finite set of edges, $f_P : E_P \to Q_P \times Q_P$ is the edge attaching mapping, and $L_P : E_P \to \Sigma$ is the edge labeling. For the mapping f_P, if $f_P(e) = (q_1, q_2)$, we will write $q_1 = f^0_P(e)$, $q_2 = f^1_P(e)$. If no confusion is possible, we will omit the indices P. Note that we allow multiple edges between any pair of locations.

In a PV automaton, we declare all locations to be *accepting*. Hence the language of a PV automaton (the set of its possible executions) is prefix-closed.

To justify our approach, we show that the formalism of [7] embeds nicely into it: [7] is concerned with (non-branching) PV *processes*, which are $+$-free regular expressions on the set $\{\mathtt{P}_a, \mathtt{V}_a \mid a \in \mathcal{O}\}$. These are given the semantics that the set of (possible) executions of a PV process is the set of *prefixes* of the regular language generated by the expression. Hence the transition from PV processes to PV automata is achieved by first translating the regular expression to a finite automaton generating the same language, and then declaring all locations of the automaton to be accepting.

We introduce a successor relation $\le \subseteq Q \times Q$ by letting $q_1 \le q_2$ if and only if $q_1 = q_2$ or there exists $e \in E$ such that $f(e) = (q_1, q_2)$. Without loss of generality we can assume our PV automata to be *connected* in the sense that there exists a path $q^0 \le \cdots \le q$ for any $q \in Q$.

PV automata are subject to a *well-behavedness* condition: During any of their executions, resources are only to be released if they have been previously locked, and once they have been locked, they cannot be locked again without being released first. Hence execution sequences like, e.g., $\mathtt{V}_a.\mathtt{P}_a$ and $\mathtt{P}_a.\mathtt{P}_a$ are to be disallowed.

Imposing the well-behavedness condition on a given PV automaton $P = (Q, q^0, E, f, L)$ is achieved by associating with it a *resource-use characteristics* mapping $r : \mathcal{O} \times Q \to \mathbb{Z}$ as follows:

(i) Let $r_a(q^0) = 0$ for all $a \in \mathcal{O}$.

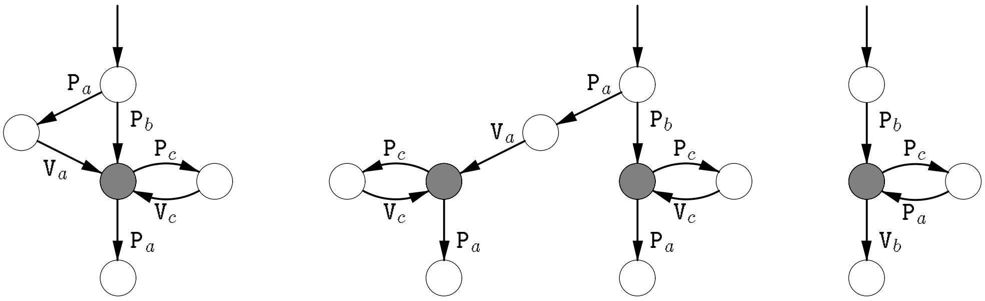


Fig. 1. Examples of resource-use conflicts. On the left, the shaded location has a conflict in r_b: Coming from the left, $r_b = 0$, coming from above, $r_b = 1$. The conflict is resolved by splitting up the offending location in the second automaton. The automaton on the right has an unresolvable resource-use conflict.

(ii) For all $e \in E$, $f(e) = (q_1, q_2)$, such that $r_a(q_1)$ has been defined and $r_a(q_2)$ has not for some $a \in \mathcal{O}$, let

$$r_a(q_2) = \begin{cases} r_a(q_1) + 1 & \text{if } L(e) = \mathtt{P}_a, \\ r_a(q_1) - 1 & \text{if } L(e) = \mathtt{V}_a, \\ r_a(q_1) & \text{else} \end{cases}$$

for all $a \in \mathcal{O}$.

Step (ii) is to be repeated until no more such e exist. As P is connected, the algorithm terminates with resource-use characteristics assigned to every location in Q. We assume that P has no *resource-use conflicts*, that is, the resource-use characteristics of a location is independent of which path one takes to it from q^0. This condition can easily be verified within the above algorithm, and some of the conflicts can be resolved by splitting up the offending locations, cf. figure 1.

With resource-use characteristics assigned to every location, the automaton is said to be well-behaved if and only if $r_a(q) \in \{0, 1\}$ for all $a \in \mathcal{O}$, $q \in Q$.

2.2 PV Programs

A PV program is a finite set $\mathcal{P} = \{P_1, \dots, P_n\}$ of well-behaved PV automata. The semantics of a PV program is, as in [7], given by associating a transition system with it:

A configuration of $\mathcal{P}$ is a function $\kappa : \mathcal{P} \to \bigcup_{i=1}^n Q_{P_i}$ mapping each automaton in $\mathcal{P}$ to one of its locations, i.e. such that $\kappa P_i \in Q_{P_i}$ for all $P_i \in \mathcal{P}$. The initial configuration is κ^0 given by $\kappa^0 P_i = q^0_{P_i}$ for all $P_i \in \mathcal{P}$. We transfer the successor relation $\leq$ to configurations of PV programs by letting $\kappa_1 \leq \kappa_2$ if and only if $\kappa_1 P_i \leq_{P_i} \kappa_2 P_i$ for all $P_i \in \mathcal{P}$, and we let $\leq^*$ denote the transitive closure of $\leq$.

The resource-use characteristics of a configuration κ is $r_a(\kappa) = \sum_{i=1}^n r_a(\kappa P_i)$, i.e. $r_a(\kappa)$ is the number of locks the processes together hold on resource a in the configuration κ.

A configuration κ is *allowed* iff $r_a(\kappa) \leq s(a)$. If $\kappa_1 \leq \kappa_2$ and both κ_1 and κ_2 are allowed, then $\kappa_1 \leq \kappa_2$ is an *allowed computation step.* An allowed computation step is denoted $\kappa_1 \mapsto \kappa_2$; the transitive closure of $\mapsto$ we indicate by $\mapsto^*$.

If $\mathrm{Con}^{\mathrm{a}}\mathcal{P}$ denotes the set of allowed configurations of the PV program $\mathcal{P}$, then the transition system $(\mathrm{Con}^{\mathrm{a}}\mathcal{P}, \kappa^0, \mapsto)$ gives the semantics of $\mathcal{P}$. Note that $\mapsto \, = \, \leq \cap \, (\mathrm{Con}^{\mathrm{a}} \times \mathrm{Con}^{\mathrm{a}})$.

2.3 Geometric Realization

The underlying digraph (Q, E, f) of a PV automaton $P = (Q, q^0, E, f, L)$ can be understood as a geometric object: The nodes in Q are discrete points, and the edges in E are directed unit intervals $\vec{I}$. This assigns a local po-space to the digraph (Q, E, f) called its *geometric realization* and denoted $\Phi_{\mathrm{g}} P$.

We carry over the resource-use characteristics r_a to the geometric realization as follows: Given $x \in \Phi_{\mathrm{g}} P$, let

$$r^{\mathrm{g}}_{\mathrm{a}}(x) = \begin{cases} r_a(x) & \text{if } x \in Q, \\ r_a(f^1(e)) & \text{if } x \in e \in E \text{ and } L(e) = \mathtt{P}_a, \\ r_a(f^0(e)) & \text{else.} \end{cases}$$

If x is a point on an edge, the above means that $r_a(x)$ is 1 as soon as r_a equals 1 in one of the endpoints of the edge, that is, any locations in which the automaton holds a lock on some resource are "fattened up" such that it also holds the lock on the edges incident with these locations.

The geometric realization of a PV *program* $\mathcal{P} = \{P_1, \dots, P_n\}$ is the product space $\Phi_{\mathrm{g}}\mathcal{P} = \prod_{i=1}^n \Phi_{\mathrm{g}} P_i$, which as a product of local po-spaces again is a local po-space. Resource-use characteristics is carried over to $\Phi_{\mathrm{g}}\mathcal{P}$ by letting $r^{\mathrm{g}}_{\mathrm{a}}(x) = \sum_{i=1}^n r^{\mathrm{g}}_{\mathrm{a}}(\pi_i \, x)$, where $\pi_i \, x = x_i$ is projection on the ith coordinate. Configurations of $\mathcal{P}$ are mapped into $\Phi_{\mathrm{g}}\mathcal{P}$ by declaring that $\Phi_{\mathrm{g}}\kappa = (\kappa P_1, \dots, \kappa P_n) \in \Phi_{\mathrm{g}}\mathcal{P}$.

A point $x \in \Phi_{\mathrm{g}}\mathcal{P}$ is said to be allowed if $r^{\mathrm{g}}_{\mathrm{a}}(x) \leq s(a)$ for all $a \in \mathcal{O}$. It is straightforward to see that $r^{\mathrm{g}}_{\mathrm{a}}(\Phi_{\mathrm{g}}\kappa) = r_a(\kappa)$ for all configurations κ, hence κ is allowed if and only if $\Phi_{\mathrm{g}}\kappa$ is allowed. The set of all allowed points in $\Phi_{\mathrm{g}}\mathcal{P}$ is denoted $\Phi^{\mathrm{a}}_{\mathrm{g}}\mathcal{P}$. It can be shown that $\Phi_{\mathrm{g}}\mathcal{P}$ is a compact local po-space, and that $\Phi^{\mathrm{a}}_{\mathrm{g}}\mathcal{P}$ is a closed subspace of $\Phi_{\mathrm{g}}\mathcal{P}$.

If $\trianglelefteq \subseteq \Phi_{\mathrm{g}}\mathcal{P} \times \Phi_{\mathrm{g}}\mathcal{P}$ denotes the local order[1] on the local po-space $\Phi_{\mathrm{g}}\mathcal{P}$, it is easy to see that $\kappa_1 \leq^* \kappa_2$ if and only if $\Phi_{\mathrm{g}}\kappa_1 \trianglelefteq \Phi_{\mathrm{g}}\kappa_2$. The following

[1] Compare [9, def. 26] for a definition of local order.

proposition, where $x \preccurlyeq^* y$ denotes the property that there exists a dipath from x to y in $\Phi_{\mathrm{g}}^{\mathrm{a}}\mathcal{P}$, is much more difficult to prove and constitutes the core of the applicability of the geometric realization technique for untimed PV programs.

Proposition 2.1 *Given two configurations κ_1, κ_2, then $\kappa_1 \mapsto^* \kappa_2$ if and only if $\Phi_{\mathrm{g}}\kappa_1 \preccurlyeq^* \Phi_{\mathrm{g}}\kappa_2$, and as a consequence, κ_1 is a deadlock or unsafe if and only if $\Phi_{\mathrm{g}}\kappa_1$ is a deadlock or unsafe.*

2.4 Linear PV Programs

A PV automaton $P = (Q, q^0, E, f, L)$ is said to be linear if there exist labelings $Q = (q_0, \dots, q_m)$, $E = (e_1, \dots, e_m)$, with $q_0 = q^0$, such that $f(e_j) = (q_{j-1}, q_j)$ for all $j = 1, \dots, m$. Linear PV automata correspond to the *loopless processes* of [7].

A PV *program* is called linear if all the automata that constitute it are linear. The geometric realization of a linear PV program is a (global) pospace, and in [6], an efficient algorithm is developed for finding deadlocks and unsafe points in the geometric realization of linear PV programs.

In [5], a technique called *delooping* is introduced, which makes the algorithm of [6] applicable to PV programs *with loops.* This is done by associating with a looping PV program a number of linear PV programs such that the unsafe region of the original program can be found as the intersection of the unsafe regions of the linear programs.

3 Timed PV Automata

As hinted in the introduction, we define a timed PV automaton to be a timed automaton $P = (Q, q^0, E, f, L, C, \varphi, \rho)$ on the alphabet $\Sigma = \{\tau, \mathtt{P}_a, \mathtt{V}_a \mid a \in \mathcal{O}\}$. In this expression, Q, q^0, E, f, and L are as in the untimed case, C is a finite set of clocks, $\varphi : Q \cup E \to \Phi(C)$ is a mapping assigning a *clock constraint* to each location and every edge, and $\rho : E \to 2^C$ assigns a set of clocks to be reset to every edge.

As before, all locations are accepting. For clock constraints, clock valuations, and valuation resets we use the terminology of [1], except that we also allow constraints on differences of clocks.

The following is standard in the timed automata formalism, see e.g. [1]; we state it here only to fix notation: A *state* of a timed PV automaton P is an element (q, v) of the set $S = Q \times \mathbb{R}_{\ge 0}^C$. A state (q, v) is *allowed* if $v \vDash \varphi(q)$. If (q_1, v_1), (q_2, v_2) are allowed states, then $(q_1, v_1) \mapsto (q_2, v_2)$ if either

- $q_2 = q_1$, and there exists $t \in \mathbb{R}_{\ge 0}$ such that $v_2 = v_1 + t$ and for all $0 \le t' \le t$, $v + t' \vDash \varphi(q_1)$, or
- there is $e \in E$ such that $f(e) = (q_1, q_2)$, $v_1 \vDash \varphi(e)$, and $v_2 = v_1[\rho(e) \leftarrow 0]$.

To maintain analogy with the untimed case, we need to define a "successor" relation $\le$ on S such that the *allowed-successor* relation $\mapsto$ is a subset of $\le$.

This is done by declaring that $(q_1, v_1) \leq (q_2, v_2)$ if there is $t \in \mathbb{R}_{\geq 0}$ such that $q_2 = q_1$ and $v_2 = v_1 + t$, or there is $e \in E$ such that $f(e) = (q_1, q_2)$ and $v_2 = v_1[\rho(e) \leftarrow 0]$. Note that $\leq$ coincides with $\mapsto$ if and only if $\varphi(Q \cup E) = \{\texttt{true}\}$, i.e. if the automaton has no timing constraints at all.

The *initial state* of the timed PV automaton P is $(q^0, v^0) \in S$, where v^0 is the clock valuation given by $v^0(c) = 0$ for all $c \in C$. We demand the initial state to be allowed; denoting the set of allowed states by S^{a}, the semantics of P is given by the transition system $(S^{\mathrm{a}}, \mapsto, (q^0, v^0))$.

We introduce the resource-use characteristics mapping $r : \mathcal{O} \times Q \to \mathbb{Z}$ as in the untimed case; this also gives us a notion of well-behaved timed PV automata. The mapping is extended to the *states* of P by decreeing that $r_a(q, v) = r_a(q)$.

4 Timed PV Programs

A timed PV program is a finite set $\mathcal{P} = \{P_1, \dots, P_n\}$ of well-behaved timed PV automata. The semantics of a timed PV program is again given by introducing a set of configurations and defining a transition system on it:

A configuration of $\mathcal{P}$ is a mapping $\kappa : \mathcal{P} \to \bigcup_{i=1}^n S_{P_i}$ such that $\kappa P_i \in S_{P_i}$ for all $P_i \in \mathcal{P}$. The initial configuration κ^0 is given by $\kappa^0 P_i = (q^0_{P_i}, v^0)$ for all $P_i \in \mathcal{P}$. Again we define a successor relation $\leq$ on configurations by declaring that $\kappa_1 \leq \kappa_2$ if and only if $\kappa_1 P_i \leq_{P_i} \kappa_2 P_i$ for all $P_i \in \mathcal{P}$.

Also as before, the resource-use characteristics of a configuration κ is $r_a(\kappa) = \sum_{i=1}^n r_a(\kappa P_i)$. κ is *allowed* if κP_i is allowed in S_{P_i} for all $P_i \in \mathcal{P}$ *and* $r_a(\kappa) \leq s(a)$ for all $a \in \mathcal{O}$. As initial states of timed PV automata are defined to be allowed and $r_a(\kappa^0) = 0$ for all $a \in \mathcal{O}$, the initial configuration κ^0 is allowed itself.

A successor relation $\kappa_1 \leq \kappa_2$ is an *allowed computation step*, again denoted $\kappa_1 \mapsto \kappa_2$, if κ_1, κ_2 are allowed and $\kappa_1 P_i \mapsto_{P_i} \kappa_2 P_i$ for all $P_i \in \mathcal{P}$. Again letting $\mathrm{Con}^{\mathrm{a}} \mathcal{P}$ denote the set of allowed configurations of $\mathcal{P}$, the semantics of $\mathcal{P}$ is given by the transition system $(\mathrm{Con}^{\mathrm{a}} \mathcal{P}, \kappa^0, \mapsto)$.

Note that, compared to the untimed case, the definition of allowed configurations and allowed computations now has an extra component stemming from the individual automata the program is composed of.

5 Geometric Realization

To obtain an analogy of the geometric realization notion for timed PV programs, we have to apply the technique of section 2.3 *twice*. The geometric realization of a single (untimed) PV automaton was simply a digraph, i.e. a one-dimensional local po-space, whereas the geometric realization of a timed PV automaton with d clocks will be a $(d+1)$-dimensional local po-space. Also, we will have the notion of allowed and forbidden points already for the geometric realization of timed PV automata, not only for the programs.

Given a set $C = \{c_1, \dots, c_d\}$ of clocks, there is a bijective correspondence between clock valuations in $\mathbb{R}^C_{\geq 0}$ and points of the space $\mathbb{R}^d_{\geq 0}$ given by $\tilde{v} = (v(c_1), \dots, v(c_d))$. Also, given a clock constraint $\varphi \in \Phi(C)$, we can define an associated subset $\tilde{\varphi} \in \mathbb{R}^d_{\geq 0}$—sometimes called a *clock zone*—by $\tilde{\varphi} = \{\tilde{v} \mid v \vDash \varphi\}$. Dividing out logical equivalence in $\Phi(C)$, the mapping $\varphi \mapsto \tilde{\varphi}$ also becomes bijective. With a valuation reset $v[D \leftarrow 0]$ we associate a projection mapping $\pi_D : \mathbb{R}^d_{\geq 0} \to \mathbb{R}^d_{\geq 0}$ setting all coordinates x_i with $c_i \in D$ to 0 and leaving the others untouched. For later use we record the following basic facts about the interplay between these three mappings:

Lemma 5.1 *Given $v \in \mathbb{R}^C_{\geq 0}$, $D \subseteq C$, and $\varphi \in \Phi(C)$, then*

- *$v \vDash \varphi$ if and only if $\tilde{v} \in \tilde{\varphi}$,*
- *$(v[D \leftarrow 0])^\sim = \pi_D(\tilde{v})$, and*
- *$v[D \leftarrow 0] \vDash \varphi$ if and only if $\tilde{v} \in \pi_D^{-1}(\tilde{\varphi})$.*

Proof. The first two assertions are clear from the definitions. As for the last, $v[D \leftarrow 0] \vDash \varphi$ if and only if $(v[D \leftarrow 0])^\sim \in \tilde{\varphi}$, which in turn is the case if and only if $\pi_D(\tilde{v}) \in \tilde{\varphi}$, and the latter is equivalent to $\tilde{v} \in \pi_D^{-1}(\tilde{\varphi})$. □

5.1 Timed PV Automata

The geometric realization of a timed PV automaton $P = (Q, q^0, E, f, L, C, \varphi, \rho)$, with $C = \{c_1, \dots, c_d\}$, is the space $\Phi_{\mathrm{g}} P = (Q, E, f) \times \vec{\mathbb{R}}^d_{\geq 0}$, where (Q, E, f) again is to be understood as a local po-space, and $\vec{\mathbb{R}}^d_{\geq 0}$ is the space $\mathbb{R}_{\geq 0}$ with the standard order

$$(x_1, \dots, x_d) \leq (y_1, \dots, y_d) \quad \text{iff} \quad x_i \leq y_i \text{ for all } i = 1, \dots, d.$$

$\Phi_{\mathrm{g}} P$ itself is a local po-space, however unless $d = 0$, it is not compact. We call (Q, E, f) the *location space*, $\vec{\mathbb{R}}^d_{\geq 0}$ the *clock space.*

Given a state $(q, v) \in S$, its geometric realization is the point $(q, \tilde{v}) \in \Phi_{\mathrm{g}} P$. A point $(x, \tilde{v}) \in \Phi_{\mathrm{g}} P$ is said to be *allowed* if either $x \in Q$ and $\tilde{v} \in \tilde{\varphi}(x)$, or $x \in e$ for some $e \in E$ and

$$\tilde{v} \in \tilde{\varphi}(f^0(e)) \cap \tilde{\varphi}(e) \cap \pi^{-1}_{\rho(e)}(\tilde{\varphi}(f^1(e))). \tag{1}$$

The following proposition, where $\Phi^{\mathrm{a}}_{\mathrm{g}} P \subseteq \Phi_{\mathrm{g}} P$ denotes the set of allowed points in $\Phi_{\mathrm{g}} P$, shows that our definition of $\Phi^{\mathrm{a}}_{\mathrm{g}} P$ is the "right" one:

Proposition 5.2 *Given $(x, \tilde{v}) \in \Phi_{\mathrm{g}} P$, then $(x, \tilde{v}) \in \Phi^{\mathrm{a}}_{\mathrm{g}} P$ if and only if, either $x \in Q$ and $v \vDash \varphi(x)$, or $x \in e$ for some $e \in E$ and*

- *$v \vDash \varphi(f^0(e))$,*
- *$v \vDash \varphi(e)$, and*
- *$v[\rho(e) \leftarrow 0] \vDash \varphi(f^1(e))$.*

Proof. The case $x \in Q$ is trivial. If $x \in e$ for some $e \in E$, the first two items correspond to $\tilde{v} \in \tilde{\varphi}(f^0(e)) \cap \tilde{\varphi}(e)$ in equation (1) above. Hence we are left with showing that $\tilde{v} \in \pi^{-1}_{\rho(e)}(\tilde{\varphi}(f^1(e)))$ if and only if $v[\rho(e) \leftarrow 0] \models \varphi(f^1(e))$, which however is clear by the last item of lemma 5.1. □

Resource-use characteristics is defined on $\Phi_{\mathrm{g}} P$ by first introducing it on the *space* (Q, E, f) as in section 2.3, and then "fattening it up" by declaring that $r_a(x, \tilde{v}) = r_a(x)$.

5.2 Timed PV *Programs*

The geometric realization of a timed PV program $\mathcal{P} = \{P_1, \dots, P_n\}$ is the local po-space $\Phi_{\mathrm{g}}\mathcal{P} = \prod_{i=1}^n \Phi_{\mathrm{g}} P_i$. For resource-use characteristics we have to change the definition of the untimed case, as now the components of $\Phi_{\mathrm{g}}\mathcal{P}$ are not (one-dimensional) digraphs but multi-dimensional local po-spaces. Hence now we have to project on *subspaces* rather than *coordinates*; given $x \in \Phi_{\mathrm{g}}\mathcal{P}$, we set

$$r_a^{\mathrm{g}}(x) = \sum_{i=1}^n r_a^{\mathrm{g}}(\psi_i x),$$

where ψ_i is projection on the ith subspace $\Phi_{\mathrm{g}} P_i$ of $\Phi_{\mathrm{g}}\mathcal{P}$. However as $r_a^{\mathrm{g}}(\pi_j\, x) = 0$ for all $a \in \mathcal{O}$ if π_j is projection on a *time coordinate*, the sum can as well be taken over all projections on the coordinates.

Configurations are again mapped into $\Phi_{\mathrm{g}}\mathcal{P}$ by defining $\Phi_{\mathrm{g}}\kappa = (\kappa P_1, \dots, \kappa P_n)$, where the correspondence between states (q, v) and points $(q, \tilde{v})$ is implicit.

As for allowed points in $\Phi_{\mathrm{g}}\mathcal{P}$, we again have the duality between points being forbidden due to over-use of resources and points being forbidden in the respective timed PV automata: We say that a point $x \in \Phi_{\mathrm{g}}\mathcal{P}$ is allowed if $\psi_i x$ is allowed in all $\Phi_{\mathrm{g}} P_i$ and $r_a^{\mathrm{g}}(x) \leq s(a)$ for all $a \in \mathcal{O}$.

Again we have $r_a^{\mathrm{g}}(\Phi_{\mathrm{g}}\kappa) = r_a(\kappa)$, hence by proposition 5.2, κ is allowed if and only if $\Phi_{\mathrm{g}}\kappa$ is allowed.

6 Applying the Geometric Realization

So far we have introduced a timed PV formalism and defined a geometric realization function in close analogy to what had been done previously for untimed PV programs. To actually make our proposed geometric realization technique *work*, we should provide an analog of proposition 2.1 of section 2.3: Given configurations κ_1, κ_2, there should be an (allowed) execution path from κ_1 to κ_2 if and only if there is a dipath (in $\Phi_{\mathrm{g}}^{\mathrm{a}}$) from $\Phi_{\mathrm{g}}\kappa_1$ to $\Phi_{\mathrm{g}}\kappa_2$. In the present section we shall see that this is *not* the case, and we shall propose different ways to handle the problems encountered.

For sake of simplicity, we confine ourselves to treat only timed PV *automata* in this section; let P be a given timed PV automaton with d clocks.

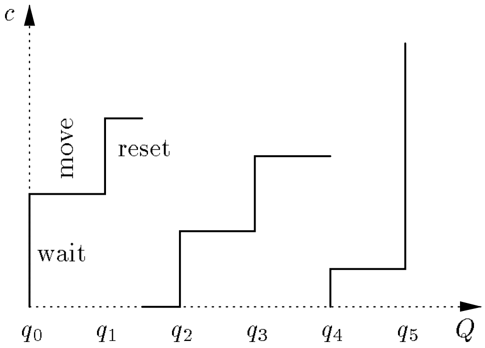


Fig. 2. A typical execution path in a timed automaton with one clock. As the clock is reset during the transition from q_1 to q_2, the execution path is not continuous.

6.1 The Reset Problem

An essential feature of timed automata is their ability to reset clocks. Indeed, if the timed automaton in question never resets its clocks, then all clocks have the same value in any reachable state, hence we might as well have only *one* clock—and the results of [8] imply that a timed automaton with one clock is strictly less expressive than one with two or more clocks.

However if clocks can be reset, there exist execution paths which are neither continuous nor directed, cf. figure 2.

This problem can be solved in several different ways. A first attempt is to simply change the order relation on $\Phi_{\mathrm{g}} P$, such that now

$$(p, x_1, \dots, x_d) \le (q, y_1, \dots, y_d) \quad \text{iff} \quad p \le q \text{ and } \forall i : (x_i \le y_i \text{ or } y_i = 0).$$

This makes execution paths directed, but still not continuous; however it also means that $\Phi_{\mathrm{g}} P$ is not a local po-space anymore: Given any point $x = (q, 0, \dots, 0)$, there exists no neighbourhood of x in which $\le$ is a partial order. This last problem might be avoided by defining the new order relation in another way, but execution paths are still not continuous.

Our second proposal is to identify certain points in $\Phi_{\mathrm{g}} P$: If m_e, for any $e \in E$, denotes the *midpoint* of e (where e is seen as a directed unit interval), define an equivalence relation $\sim$ on $\Phi_{\mathrm{g}} P$ by

$$(m_e, \tilde{v}) \sim (m_e, \pi_{\rho(e)}(\tilde{v}))$$

for any $e \in E$, $\tilde{v} \in \mathbb{R}^d_{\ge 0}$, and pass to the quotient $\Phi_{\mathrm{g}} P/\!\sim$. Certainly execution paths in $\Phi_{\mathrm{g}} P/\!\sim$ are continuous (if the convention is applied that clocks are reset at the midpoints of edges); we believe that $\Phi_{\mathrm{g}} P/\!\sim$ is a local po-space, and that execution paths in $\Phi_{\mathrm{g}} P/\!\sim$ are dipaths.

This approach has the caveat that $\Phi_{\mathrm{g}} P/\!\sim$, even though it might be a local po-space, is a rather involved space which is likely to be difficult to handle in applications. A third way to attack this problem is to enhance the timed automata formalism such that the values of certain clocks can remain zero

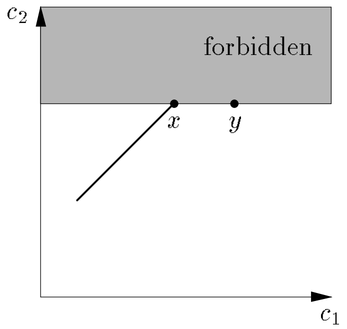


Fig. 3. An example of an execution path through a clock space constituted of two clocks. Both x and y are allowed, and there is a dipath from x to y, yet there is no execution path from x to y. In fact, x is a deadlock, as is y.

while the others are already started, an approach we will detail in section 6.3.

6.2 The Global-Time Problem

In the timed automata formalism, time is global, i.e. all clocks proceed at the same speed. This implies that execution paths run *diagonally* through the clock space, hence not all dipaths correspond to execution paths, cf. figure 3.

This situation can again be remedied by changing the order relation on $\Phi_{\mathrm{g}} P$, the new one being

$$(p, x_1, \dots, x_d) \le (q, y_1, \dots, y_d) \quad \text{iff} \quad \begin{aligned} & p \le q, \\ & \forall i : x_i \le y_i, \text{ and} \\ & y_1 - x_1 = \dots = y_d - x_d. \end{aligned}$$

With this order relation, all dipaths are execution paths. However this approach fits badly with our solution to the Reset Problem proposed in the next section.

If we want to stay with the standard order on $\mathbb{R}^d_{\ge 0}$, there is no other solution to this problem than abandoning the global-time approach altogether and consider local-time formalisms instead: As long as at least some of the clocks are synchronized with each other, there will be dipaths which do not correspond to execution paths.

6.3 Linear Timed PV Automata

In analogy to the approach in the untimed case, we should develop techniques to find deadlocks and unsafe configurations in linear timed PV programs, and we should attempt to transfer the delooping techniques of [5] to the timed case. In this paper we concentrate on the former; the latter is left open for future research.

As in the untimed case, a timed PV automaton $P = (Q, q^0, E, f, L, C, \varphi, \rho)$ is said to be linear if there exist labelings $Q = (q_0, \dots, q_m)$, $E = (e_1, \dots, e_m)$, with $q_0 = q^0$, such that $f(e_j) = (q_{j-1}, q_j)$ for all $j = 1, \dots, m$. The geometric realization of a linear timed PV automaton is a po-space.

For linear timed PV automata, the Reset Problem of section 6.1 has a third and much more elegant solution: to avoid resets altogether by introducing new clocks. Let P be the automaton from above, e_j one of its edges, and c_k one of its clocks, and assume that $c_k \in \rho(e_j)$. Introduce a new clock $\tilde{c}_{k,j}$, and in all invariants $\varphi(q_i)$, $\varphi(e_{i+1})$, $i \geq j$, replace any occurrence of c_k by $\tilde{c}_{k,j}$. The clock $\tilde{c}_{k,j}$ is to be started only when location q_j is reached, i.e. its value remains 0 before q_j; it is clear that this replacement does not alter the semantics of P.

After all resets have been resolved by the procedure above, we have an automaton without resets, but instead with a new *start function* $\sigma : C \to Q$, assigning to every clock the location in which it is started. In the geometric realization of this new automaton, execution paths are dipaths.

6.4 Deadlocks and Unsafe Configurations

In the untimed case, the notions of deadlock and unsafe configuration are very intuitive: A non-final configuration κ is a deadlock if and only if there does not exist any κ' such that $\kappa \mapsto \kappa'$, and κ is unsafe if and only if no final configuration can be reached from κ, i.e. if $\uparrow\kappa \cap \mathcal{F} = \emptyset$. Here $\mathcal{F}$ is the set of final configurations, and $\uparrow\kappa = \{\kappa' \mid \kappa \mapsto^* \kappa'\}$.

For *linear* untimed PV programs, these notions are connected in that *a configuration is unsafe if and only if any execution from it reaches a deadlock.*[2] The algorithm of [6] finds all unsafe configurations in a given linear untimed PV program by recursively finding all deadlocks and "tracing them back" to find their associated unsafe configurations.

For timed PV automata the situation is somewhat more complicated. First, there are two kinds of deadlocks: Applying the definition from above gives a notion of *state deadlock*; a state is a state deadlock if no location switch can occur and time cannot increase. However we also need a second notion; a *location deadlock* is a state in which time is permitted to increase, but at no reachable future state a location switch is possible. Hence the automaton is locked in the present location, but time might increase indefinitely:

Definition 6.1 A state $(q, v) \in Q \times \mathbb{R}_{\geq 0}^C$ is called a location deadlock if, for all $t \in \mathbb{R}_{\geq 0}$ such that $(q, v) \mapsto (q, v + t)$ and for all $e \in E$ such that $f^0(e) = q$,

- $(q, v + t)$ is non-final, and
- $v + t \not\models \varphi(e)$ or $(v + t)[\rho(e) \leftarrow 0] \not\models \varphi(f^1(e))$.

It is clear that every state deadlock is also a location deadlock.

Second, turning our attention to *linear* timed PV automata and assuming that the final states are exactly those whose location is q_m, i.e. the "last" loca-

[2] Here it is to be assumed that the only final configuration is the one consisting of the final locations of the individual automata.

tion, we see that *a state is unsafe if and only if any execution from it reaches a location deadlock*. Hence we should find all *location* deadlocks and trace them back to find their associated unsafe states, however a straightforward application of the algorithm of [6] only finds *state* deadlocks. In the next section we shall propose a rectification of this problem.

6.5 Compactness

If we want to imitate the algorithm of [6], we should not just work in a po-space, but in a *compact po-space*—however the geometric realization of a linear timed PV automaton is not compact.

It is known that any timed automaton can be converted into an equivalent one whose clocks are guaranteed not to exceed a certain maximal value, and if M is this maximal value, the geometric realization of such a timed automaton can be taken to be $(Q, E, f) \times [0, M]^d$, which is a compact space. However the conversion introduces a number of new loops into the automaton and involves resetting clocks to values other than 0. We prefer another solution which also copes with the "location deadlock problem" of the previous section:

Instead of the time axes in the geometric realization being $\vec{\mathbb{R}}_{\geq 0}$, we propose to take their one-point compactifications $\vec{\mathbb{R}}_{\geq 0} \cup \{\infty\}$, which are dihomeomorphic to the ordered unit interval $\vec{I}$. The geometric realization of a timed PV automaton is then $(Q, E, f) \times \vec{I}^d$, which is a compact local po-space.

In this new setting, location deadlocks can be detected in a straightforward way as "deadlocks at time $1 = \infty$", which makes the algorithm of [6] applicable for detecting location deadlocks as-is. However for finding unsafe points, it will have to be adapted, as the backtracking in the timed setting is different from the untimed setting. Note also that now execution paths do not necessarily run diagonally through the clock space anymore.

6.6 What We End Up With

With the changes proposed in sections 6.3 and 6.5, our geometric realization technique takes the following form:

Let $P = (Q, q^0, E, f, L, C, \varphi, \sigma)$ be a linear timed PV automaton, where we instead of the reset mapping ρ now have a start function $\sigma : C \to Q$, and let $C = \{c_1, \dots, c_d\}$, $Q = \{q_0, \dots, q_m\}$, $E = \{e_1, \dots, e_m\}$, where $q_0 = q^0$ and $f(e_j) = (q_{j-1}, q_j)$ for all $j = 1, \dots, m$. The geometric realization of P is

$$\Psi_{\mathrm{g}} P = \overrightarrow{[0, m]} \times \vec{I}^d,$$

which is a compact po-space. We use the symbol Ψ_{g} to distinguish it from the "old" geometric realization $\Phi_{\mathrm{g}} P$.

Let $\Theta : (\mathbb{R}_{\geq 0} \cup \{\infty\})^d \to I^d$ be the dihomeomorphism given by

$$\Theta(x_1, \dots, x_d) = \Big(\frac{x_1}{1+x_1}, \dots, \frac{x_d}{1+x_d} \Big).$$

With this dihomeomorphism in use, a clock valuation $v \in \Phi(C)$ of P corresponds to the point $\hat{v} = \Theta(\tilde{v}) \in I^d$, and the definition of clock zone has to be adapted accordingly.

A state (q_j, v) of P corresponds to the point $(j, \hat{v}) \in \Psi_{\mathrm{g}} P$, and if tildes are replaced by hats, the definition of allowed points in $\Psi_{\mathrm{g}} P$ (equation (1) on page 7) can be taken over unchanged.

As now clocks are started in locations different from q^0, we need to keep track of which way execution paths are allowed to run in the geometric realization. We do this by introducing a mapping $\hat{\delta} : Q \times I^d \to \mathbb{R}^d_{\geq 0}$, derived from the start function σ and defining a vector field on every instance of the $m+1$ clock spaces. Execution paths are then (piecewise smooth) curves through $\Psi_{\mathrm{g}} P$ which are integral curves to the vector fields in the clock spaces and run horizontally in $[0, m]$.

First, define an accumulated start function $R : Q \to 2^C$, where $R(q)$ contains all clocks which are *running*, i.e. have been started already, in q:

$$R(q_j) = \{c \in C \mid \sigma(c) = q_i, i \leq j\}$$

This mapping translates to a function $\tilde{\delta} : Q \to \{0,1\}^d$, by $\tilde{\delta}_i(q) = 1$ if and only if $c_i \in R(q)$. The mapping $\tilde{\delta}(q, \tilde{v}) = \tilde{\delta}(q)$ defines a (very simple) vector field on all clock spaces in the *original* space $\Phi_{\mathrm{g}} P = [0, m] \times \mathbb{R}^d_{\geq 0}$, and in this space execution paths are integral curves to these vector fields.

Conjugating the vector fields given by $\tilde{\delta}$ with the homeomorphism Θ then yields our new mapping $\hat{\delta} : Q \times I^d \to \mathbb{R}^d_{\geq 0}$:

$$\hat{\delta}(q, \hat{v}) = \big((1 - \hat{v}_1)^2 \tilde{\delta}_1(q), \dots, (1 - \hat{v}_d)^2 \tilde{\delta}_d(q) \big).$$

With this in place, we are now able to give an exact characterization of the execution paths in $\Psi_{\mathrm{g}} P$: A dipath $\gamma : \vec{I} \to \Psi_{\mathrm{g}} P = \overrightarrow{[0, m]} \times \vec{I}^d$ is an execution path if and only if there exists a partition $0 = t_0, t_1, \dots, t_n = 1$ of I such that

- $\gamma(t_i) \in Q \times I^d$ for all $i = 0, \dots, n$,
- $\gamma_{|[t_i, t_{i+1}]}$ is smooth for all $i = 0, \dots, n-1$, and
- for all $i = 0, \dots, n-1$, either

$$\mathrm{d}\gamma_{|[t_i, t_{i+1}]}|_{\gamma(t)} = \big((t_{i+1} - t_i)^{-1}, 0, \dots, 0 \big)$$

for all $t \in [t_i, t_{i+1}]$, or

$$\mathrm{d}\gamma_{|[t_i, t_{i+1}]}|_{\gamma(t)} = \big(0, (1 - \gamma_1(t))^2 \tilde{\delta}_1(\gamma(t_i)), \dots, (1 - \gamma_d(t))^2 \tilde{\delta}_d(\gamma(t_i)) \big)$$

for all $t \in [t_i, t_{i+1}]$.

Here $d\gamma_{|[t_i,t_{i+1}]}|_{\gamma(t)}$ denotes the differential of γ restricted to the interval $[t_i, t_{i+1}]$, taken in the point $\gamma(t)$. Note that any *path* fulfilling these constraints is automatically a dipath, hence the condition on γ being a dipath can be removed.

7 Future Work

We believe that in the setting laid out in section 6.6, we are not far from being able to apply the algorithm of [6] to find deadlocks and unsafe configurations in timed PV programs. There are two issues still to consider; however to us they are mainly of a computational, rather than conceptual, nature:

First, in the algorithm of [6], deadlocks are critical intersection points of *isothetic hyperrectangles*, where in our setting they are critical intersection points of "skew" regions in space. Computing intersections of hyperrectangles is easy, computing intersections of skew regions is more difficult. Also, in [6] deadlocks are critical intersection points of n hyperrectangles, where n is the dimension of the po-space in question, whereas in our setting, *all* clocks are stopped from running by a restriction on just *one* clock (since time is global), hence for us, deadlocks are critical intersection points of just *two* forbidden regions.

Second, in the original algorithm the set of unsafe points associated to a certain deadlock is a hyperrectangle "below" the critical intersection point. In our setting, the unsafe set would again be (the interior of) a "skew" region in space, bounded by certain hypersurfaces adhering to the constraints on execution paths above. Computing this set might be computationally expensive, but it is certainly possible.

We also believe that a generalization of our approach to a certain class of *hybrid automata* is feasible, namely these hybrid automata (cf. [2]) where clock resets are deterministic. For timed automata, our approach of section 6.6 might be a bit clumsy, but for hybrid automata we believe it is the right way to go.

Acknowledgement

I am indebted to Lisbeth Fajstrup, Martin Raussen, Kim G. Larsen, and Eric Goubault for sharing many valuable thoughts with me, and to John Leth for correcting an early version of this article.

References

[1] R. Alur. Timed automata. In *Proc. CAV'99*, volume 1633 of *Lecture Notes in Computer Science*, pages 8–22. Springer-Verlag, 1999.

[2] F. Cassez, T. A. Henzinger, and J.-F. Raskin. A comparison of control problems for timed and hybrid systems. In *Proc. HSCC'02*, volume 2289 of *Lecture Notes*

in Computer Science, pages 134–148. Springer-Verlag, 2002.

[3] E. Dijkstra. Co-operating sequential processes. In F. Genuys, editor, *Programming Languages*, pages 43–112. Academic Press, New York, 1968.

[4] U. Fahrenberg. Towards an efficient algorithm for detecting unsafe states in timed concurrent systems. Master's thesis, Aalborg University, Dept. of Mathematical Sciences, 9220 Aalborg East, Denmark, May 2002.

[5] L. Fajstrup. Loops, ditopology and deadlocks. *Mathematical Structures in Computer Science*, 10:459–480, 2000.

[6] L. Fajstrup, E. Goubault, and M. Raussen. Detecting deadlocks in concurrent systems. In *Proc. CONCUR'98*, volume 1466 of *Lecture Notes in Computer Science*, pages 332–347. Springer-Verlag, 1998.

[7] L. Fajstrup and S. Sokołowski. Infinitely running concurrent processes with loops from a geometrical viewpoint. *Electronic Notes in Theoretical Computer Science*, 39(2), 2000.

[8] T. A. Henzinger, P. W. Kopke, and H. Wong-Toi. The expressive power of clocks. In *Proc. ICALP'95*, volume 944 of *Lecture Notes in Computer Science*, pages 417–428. Springer-Verlag, 1995.

[9] S. Sokołowski. A case for po-manifolds. To be presented at the GETCO'02 workshop, 2002.

A case for po-manifolds
in chase after a good topological model for concurrency

Stefan Sokołowski*

Gdańsk, 2nd October 2002

Abstract

It has repeatedly been argued that the *semi-cubical complexes*[1], which derive directly from the higher-dimensional automata, are as general a model for concurrency as one may ever need. In fact, most (but not all) classical examples of interacting concurrent processes have an adequate semi-cubical description. I believe this is only due to the fact that most classical examples are built around the notion of discrete event, or action, which makes the transition from the example's idea to higher-dimensional automata, to semi-cubical complexes, very natural.

But it is not difficult to provide realistic examples of concurrency not based on such discrete events, which leads to *local po-spaces* with no "cubification". Although the concept of local po-space is very general and admits many pathologies, the examples of concurrency I have in mind are well-behaved and not a pathology at all, so the generality of the local po-spaces is for them a large overkill. Still, they do not fit into the semi-cubical strait jacket.

This report puts forward the notion of po-manifold, a subcategory of the local po-spaces slightly bigger than the semi-cubical complexes. A po-manifold is locally homeomorphic to a block, which is an extremely nice *global po-space*. The local homeomorphisms satisfy a simple consistency condition. This means the po-manifolds are "locally nice" even though they may be too smooth to admit a cubification.

*Institute of Computer Science, Polish Academy of Sciences, Gdańsk Division, ul. Abrahama 18, 81-825 Sopot, Poland, e-mail: `stefan@ipipan.gda.pl`.

[1] The italicized technical terms in this abstract are given definitions within the paper. Still, the reader is assumed to be more or less familiar with them. The main reference is [5] by Fajstrup, Goubault and Raussen; for some other, see inside.

Contents

1 Generalized this and that

Let us have a closer look at new incarnations of some examples of concurrency that we all know and love. The variants considered will differ from the original versions in the uniform way: the events will take time to happen and their effects will accumulate gradually.

1.1 Generalized producer and consumer

Your employer is paying you money while magazines are taking it from you. These two activities must be coordinated in time but the coordination is not exactly a succession of discrete "earn" and "pay" events. That is, unless one wants to use a very small grain, like earning or paying a single *zloty*, which would make the picture extremely complicated obscuring the main idea of the coordination.

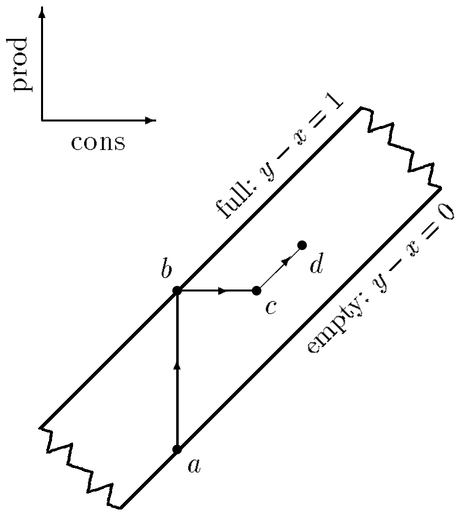
The spark-plugs in your car are drawing from the battery while the alternator is charging it. People are dying while other people are being born. All these are examples of the generalized *producer and consumer* problem in which the buffer is very large. So large that it is infeasible, or at least highly impractical, to regard single atomic *in*s and *out*s as happening instantly — one would rather view them as smeared over a larger time span. But the big buffer can still be overflown or exhausted, if one of the processes does not keep pace with the other. Can we model such coordination geometrically?

There is, certainly, a nice po-space for it — the diagonal ribbon

$$X \overset{\text{def}}{=} \left\{ \langle x, y \rangle \in \mathbb{R}^2 \,\middle|\, 0 \leq y - x \leq 1 \right\}$$

depicted opposite. The picture presents a dipath, which begins in point a with the empty buffer; then the producer works alone until in b the buffer is full; then the producer rests while the consumer works alone until in c the buffer is half full; and then both the producer and the consumer work together at the same pace until d. At every point $\langle x, y \rangle$, the difference $y - x$ measures the occupancy of the buffer.

If, for some reason, a compact space is preferable to the infinite ribbon, one can wrap it around a cylinder $\widehat{X}$ obtaining a local instead of a global po-space. Neither the po-space X nor the local po-space $\widehat{X}$ are geometric realizations of finite semi-cubical complexes.



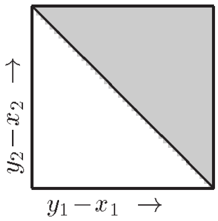
1.2 Not quite exclusive access

The diagonal ribbon X from Sec. 1.1 will now be used to a different game. Let it correspond to a *single* process whose demand for a resource may be changing with time (which flows diagonally: to the right and upwards). This demand is now measured by $y - x$:

$y - x = 0$ — the process does not need the resource,
$y - x = 1$ — the process needs the *whole* resource,
$0 < y - x < 1$ — the process needs a part of the resource (and is prepared to share it with another process).

(classically, the latter case is not considered).

Two such processes running concurrently and competing for the same resource give rise to a subset of the Cartesian square of X:



$$Y \overset{\text{def}}{=} \left\{ \langle \langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle \rangle \in X \times X \,\middle|\, y_1 - x_1 + y_2 - x_2 \leq 1 \right\}$$

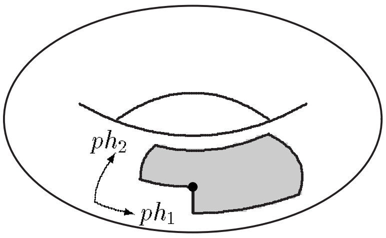
— the demands for the resource from both processes must remain within the white triangle in the picture opposite (the time directions of both processes are not depicted).

Do not overlook a subtlety: *topologically*, Y is the Cartesian product of the white triangle in the picture by the plane $\mathbb{R}^2$; but *ditopologically*, it is not. The triangle is not a po-space; in particular, for any two points a and b in the triangle there exists a dipath in Y whose projection onto the triangle goes from a to b.

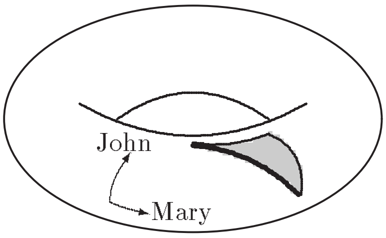
And, as before, neither the global po-space Y nor the compact local po-space $\widehat{Y}$ resulting from Y by wrapping the $\mathbb{R}^2$ component around a torus, are geometric realizations of finite semi-cubical complexes.

1.3 Generalized philosophers

The 2-dining philosophers example is the case where two processes compete for the exclusive access to two resources (cf. [12]). The resources are allocated independently, so this may lead to deadlock, if each process grabs one resource and waits for the other. The directions in which the two philosophers are moving and the deadlock point are depicted opposite. This local po-space is a geometric realization of a semi-cubical complex.



Should every process need the exclusive access to more resources, rather than to two forks as the philosophers do, the number of deadlock points would grow. Assume, for instance, that Mary and John have 1500 *złotych* together, enough either for a bicycle for John, or for skis for Mary, but not for both. If each of them takes some (more than zero) money from the common pool and would not give it back again, then neither a bicycle nor skis can be bought.



In this case, the corresponding local po-space is the torus with a triangular hole and the deadlock set is the hypotenuse of the triangle (the solid line). Again, this is not a geometric realization of a semi-cubical complex.

1.4 Non-cubical motivations

The concept of semi-cubical complexes arises very naturally from the notion of *higher dimensional automata* (cf. [14] by van Glabbeek). This is a formalism for *discrete* actions performed concurrently.

Playing with concurrent processes which are *continuous*, i.e., not based on discrete events, is not a novelty in computer science. Every mature formalism eventually attempts to do this. For instance, see [2] or [3] by David and Alla for continuous and hybrid Petri nets.

My feeling is that the geometric/topological view of concurrency is, by its very nature, better tailored to continuity than, e.g., Petri nets, so it would be a shame to leave this direction of research unexplored. Admittedly, the concept of local po-space is very general and allows for many pathologies. It seems a good idea though, to generalize the semi-cubical complexes just slightly, to overcome their finitary and combinatorial nature. The following section will show how this aim is served by the notion of *po-manifold.* So please, read on.

Please, refer to App. A on page 15 and to App. B on page 18 for short reviews of directed topology and of semi-cubical complexes.

2 Pleading for po-manifolds

The po-spaces, global or local, such as the ones from Sec. 1, are simple and have strong concurrency motivations. Still, they do not correspond to any semi-cubical complexes. The moral is that semi-cubical complexes are not a sufficient formalism for some strongly motivated concurrent systems. On the other extreme, local po-spaces are way too general. One would need a more specific category of "nice" local po-spaces, good enough for non-pathological real life examples. The generalization presented in this paper begins, in a sense, from the finite semi-cubical complexes and goes in two directions:

- The local po-spaces are glued together from "nice" subsets of the directed cubes $\vec{\mathbb{I}}^n$ rather than from the whole cubes. This way, the resulting local po-spaces are still "locally nice", but not necessarily discrete.
- They are not necessarily glued along their faces — a certain overlap is allowed, under natural consistency conditions on the common parts.

This reminds of a topological manifold with a structure (such as *smooth* manifold), which is why I call them po-manifolds. The main difference is that not all the cubes taking part in the construction of po-manifolds are necessarily in the same dimension. This makes po-manifolds a proper generalization of semi-cubical complexes (as shown in Sec. 3 on page 12).

2.1 $\mathcal{S}$-manifolds as such

Manifolds are topological spaces locally modeled on Euclidean spaces. It turns out that an analogous concept defined for a certain subcategory $\mathcal{S}$ of po-spaces instead of Euclidean spaces leads to a class of local po-spaces. In particular, if $\mathcal{S}$ consists of *all* compact po-spaces, then we get all local po-spaces (see below). Playing with $\mathcal{S}$, one gets restricted subcategories of *LPO*, the category of local po-spaces[2].

More precisely, the eligible categories $\mathcal{S}$ are the po-pattern categories as defined below:

1 Definition (po-pattern category)
A *po-pattern category* is an arbitrary full subcategory[3] $\mathcal{S}$ of *compact-PO*, the category of compact po-spaces, such that

- the closure of every open subspace of an $\mathcal{S}$-object is an $\mathcal{S}$-object, and
- for any $\mathcal{S}$-objects S_0, S_1 and S_2 and any dihomeomorphic embeddings $$\begin{tikzcd} S_0 \arrow[r, "\varphi"] \arrow[d, "\psi"'] & S_1 \\ S_2 & \end{tikzcd}$$, the pushout of φ and ψ is in $\mathcal{S}$.

An important example of a po-pattern category will be given in Sec. 2.3 on page 10.

2 Definition ($\mathcal{S}$-manifold)
Let $\mathcal{S}$ be a po-pattern category. An $\mathcal{S}$-*manifold* is a compact space X with

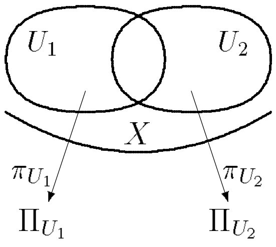
[2] See App. A on page 15 for a short review of basic notions and notations in the directed topology.

[3] A subcategory $\mathcal{S}$ of $\mathcal{T}$ is full iff every $\mathcal{T}$-morphism between $\mathcal{S}$-objects is an $\mathcal{S}$-morphism.

- an open cover $\mathcal{U}$, and
- a family $\Pi \overset{\text{def}}{=} \{\Pi_U \in \mathcal{S} \mid U \in \mathcal{U}\}$ of $\mathcal{S}$-objects (one for each open set from the cover), and
- a family $\pi \overset{\text{def}}{=} \{\pi_U : \mathrm{cl}_X U \to \Pi_U \mid U \in \mathcal{U}\}$ of homeomorphisms referred to as *coordinate patches* such that the composition

$$\pi_{U_2} \circ \pi_{U_1}^{-1}|_{\pi_{U_1}(\mathrm{cl}_X(U_1 \cap U_2))} : \pi_{U_1}(\mathrm{cl}_X(U_1 \cap U_2)) \to \Pi_{U_2} \qquad (1)$$

  is a dimap for all $U_1, U_2 \in \mathcal{U}$.



($\mathrm{cl}_X U$ denotes the closure of U in X).

Note that there is no *a priori* ordering in an $\mathcal{S}$-manifold nor in the opens in its cover. But the orderings may be retrieved from the orders in the po-spaces Π_U for $U \in \mathcal{U}$:

3 Proposition
Assume $\langle X, \mathcal{U}, \Pi, \pi\rangle$ is an $\mathcal{S}$-manifold. The relation $\leq \subseteq X \times X$ defined by

$$x \leq y \overset{\text{def}}{\iff} \exists_{U \in \mathcal{U}}\ x \leq_U y \quad \textit{where} \quad x \leq_U y \overset{\text{def}}{\iff} \pi_U x \leq_{S_U} \pi_U y$$

is a local order in X, making X a local po-space (see Def. 26 on page 17).

Proof of Prop. 3:

The order $\leq_U$ is carried over from the one in Π_U via the homeomorphism π_U^{-1}. Therefore, each $\langle \mathrm{cl}_X U, \leq_U\rangle$ is a po-space.

Take two opens $U, V \in \mathcal{U}$ and two points $x, y \in \mathrm{cl}_X(U \cap V)$. By the definition of the local orders,

$$\begin{aligned} x \leq_U y &\iff \pi_U x \leq_{S_U} \pi_U y && \text{(since } \pi_V \circ \pi_U^{-1} \text{ is a dimap)} \\ &\Longrightarrow (\pi_V \circ \pi_U^{-1})(\pi_U x) \leq_{S_V} (\pi_V \circ \pi_U^{-1})(\pi_U y) \\ &\iff \pi_V x \leq_{S_V} \pi_V y \\ &\iff x \leq_V y \end{aligned}$$

Since the orders are consistent (cf. Prop. 28 on page 18), X is a local po-space.
□

4 Definition (sub-manifold)
A *sub-manifold* of an $\mathcal{S}$-manifold $\langle X, \mathcal{U}, \Pi, \pi\rangle$ is any $\mathcal{S}$-manifold $\langle X', \mathcal{U}', \Pi', \pi'\rangle$ such that $X' \subseteq X$, $\mathcal{U}' = \{U \cap X' \mid U \in \mathcal{S}\}$ and for any $U \in \mathcal{S}$, the diagram

$$\begin{tikzcd} U \cap X' \arrow[r, hook] \arrow[d, "\pi'_{U\cap X'}"'] & U \arrow[d, "\pi_U"] \\ \Pi'_{U\cap X'} \arrow[r, hook] & \Pi_U \end{tikzcd}$$

commutes (this means the primed π's are the restrictions of the unprimed ones).

5 Definition ($\mathcal{S}$-manifold morphisms)
An *$\mathcal{S}$-manifold morphism* $\varphi : \langle X, \mathcal{U}, \Pi, \pi\rangle \to \langle Y, \mathcal{V}, \Sigma, \sigma\rangle$ is a continuous map $\varphi : X \to Y$ such that the composition $\sigma_V \circ \varphi \circ \pi_U^{-1} : \Pi_U \to \Sigma_V$ is a dimap for all $U \in \mathcal{U}$ and $V \in \mathcal{V}$.

An $\mathcal{S}$*-manifold homeomorphism* is an $\mathcal{S}$-manifold morphism for which there exists an inverse $\mathcal{S}$-manifold morphism. An $\mathcal{S}$-manifold morphism $\varphi : \langle X, \mathcal{U}, \Pi, \pi\rangle \to \langle Y, \mathcal{V}, \Sigma, \sigma\rangle$ is an $\mathcal{S}$*-manifold embedding* iff φ is an $\mathcal{S}$-manifold homeomorphism from $\langle X, \mathcal{U}, \Pi, \pi\rangle$ to a certain sub-manifold $\langle Y', \mathcal{V}', \Sigma', \sigma'\rangle$ of $\langle Y, \mathcal{V}, \Sigma, \sigma\rangle$.

6 Proposition
$\mathcal{S}$-manifolds with $\mathcal{S}$-manifold morphisms form a category. Its isomorphisms are the $\mathcal{S}$-manifold homeomorphisms.

The proof of Prop. 6 is left to the reader.

This category will be denoted by $\mathcal{S}$-$\underline{\underline{\mathit{Man}}}$.

7 Corollary
The introduction of the local order in an $\mathcal{S}$-manifold described in Prop. 3 is a functor from $\mathcal{S}$-$\underline{\underline{\mathit{Man}}}$ onto a full subcategory of $\underline{\underline{\mathit{LPO}}}$. This functor will be denoted by

$$|\ldots| : \mathcal{S}\text{-}\underline{\underline{\mathit{Man}}} \to \underline{\underline{\mathit{LPO}}}$$

and called the realization *of $\mathcal{S}$-manifolds.*

The proof of Cor. 7 is left to the reader.

The functor from Cor. 7 can be, with some restrictions, reversed. Assume

- $\langle X, \leq\rangle$ is a compact local po-space,
- $\mathcal{U}$ is its po-basis (cf. Prop. 27 on page 18),
- $\mathcal{V}$ is a "finer" basis: for each $V \in \mathcal{V}$ there is an $U \in \mathcal{U}$ s.t. $\mathrm{cl}_X V \subseteq U$ (this implies that $\mathcal{V}$ is a po-basis too).

8 Proposition
Given a compact local po-space and its po-bases, as required above, the quadruple $\langle X,\, \mathcal{U},\, \{\mathrm{cl}_X V \,|\, V \in \mathcal{V}\},\, \{Id_{\mathrm{cl}_X V} \,|\, V \in \mathcal{V}\}\rangle$ is a $\underline{\underline{\text{compact-PO}}}$-manifold. Moreover,

- *the manifolds obtained for different selections of po-bases $\mathcal{U}$ and $\mathcal{V}$ are homeomorphic to each other, and*
- *this construction is functorial.*

Cor. 7 and Prop. 8 imply together that

$$\underline{\underline{\text{compact-PO-}}}\underline{\underline{\mathit{Man}}} = \underline{\underline{\text{compact-LPO}}} \qquad \text{up to manifold homeomorphism.}$$

Proof of Prop. 8 (draft):

The important part of this proof is the demonstration that $\underline{\underline{\text{compact-PO}}}$ is a po-pattern category. This follows from Cor. 25 on page 17.

For the manifold homeomorphism related to the transiton between different po-bases, take the identity on X.

□

$\mathcal{S}$-manifolds are, in a sense, a way of assigning a category of local po-spaces to every po-pattern category $\mathcal{S}$.

2.2 The pushout property of $\mathcal{S}$-manifolds

9 Theorem

Assume $\mathcal{S}$ is a po-pattern category. Let $\langle X, \mathcal{U}, \Pi, \pi\rangle$, $\langle Y, \mathcal{V}, \Sigma, \sigma\rangle$ and $\langle Z, \mathcal{W}, \Theta, \theta\rangle$ be $\mathcal{S}$-manifolds. For every pair

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

of $\mathcal{S}$-manifold embeddings, there exists a pushout in $\mathcal{S}$-$\underline{\underline{Man}}$.

Proof of Thm. 9:

Let $\nu_1 : Y \to Y \oplus Z$ and $\nu_2 : Z \to Y \oplus Z$ be the natural embeddings of the topological spaces into their disjoint union. Take $\simeq_{\varphi\psi}$ to be the least equivalence relation in $Y \oplus Z$ such that $\nu_1(\varphi x) \simeq_{\varphi\psi} \nu_2(\psi x)$ for any $x \in X$. Define the quotient

$$Q \overset{\text{def}}{=} (Y \oplus Z) \big/ \simeq_{\varphi\psi}$$

with the natural quotient map $\kappa : Y \oplus Z \to Q$. By virtue of the compactness of X, Y and Z, the quotient Q is compact[4]. Now, the diagram opposite commutes and Q with $\kappa \circ \nu_1$ and $\kappa \circ \nu_2$ is the topological pushout of φ and ψ.

$$\begin{tikzcd} X \arrow[r, "\varphi"] \arrow[d, "\psi"'] & Y \arrow[d, "\kappa \circ \nu_1"] \\ Z \arrow[r, "\kappa \circ \nu_2"'] & Q \end{tikzcd}$$

It still needs to be demonstrated that Q is an $\mathcal{S}$-manifold pushout too. To this aim, a cover $\mathcal{R}$ and coordinate patches , and γ on Q have to be defined and respective correctness properties verified.

Select arbitrary opens from the covers: $U \in \mathcal{U}$, $V \in \mathcal{V}$ and $W \in \mathcal{W}$. The set

$$A \overset{\text{def}}{=} U \cap \varphi^{-1}(V) \cap \psi^{-1}(W)$$

is open in X. Since φ and ψ are $\mathcal{S}$-manifold embeddings, the set $\varphi(A)$ is open in $\varphi(X)$ and contained in V, while the set $\psi(A)$ is open in $\psi(X)$ and contained in W. Therefore, there exist opens $B \subseteq V \subseteq Y$ and $C \subseteq W \subseteq Z$ such that

$$A = \varphi^{-1}(B) \quad \text{and} \quad A = \psi^{-1}(C) \tag{2}$$

(the selection of such B and C is arbitrary). Now let

$$R \overset{\text{def}}{=} (\kappa \circ \nu_1)(B) \cup (\kappa \circ \nu_2)(C) \subseteq Q$$

This set is open in Q because its coimages $B = (\kappa \circ \nu_1)^{-1}(R)$ and $C = (\kappa \circ \nu_2)^{-1}(R)$ are open. Besides, $\mathrm{cl}_Q R$ is the topological pushout of the restrictions $\varphi|_{\mathrm{cl}_X A}$ and $\psi|_{\mathrm{cl}_X A}$. Define an appropriate $\mathcal{S}$-object $,_R$ and a coordinate patch $\gamma_R : R \to ,_R$ by the following diagram:

[4] Note that without this precaution, Q might even fail to be T_0 — see App. A.3 on page 17 for sufficient conditions for Q's compactness.

$$\begin{tikzcd}
\Pi_U \arrow[dr, phantom, sloped, "\supseteq"] & & & & & \Sigma_V \arrow[dl, phantom, sloped, "\supseteq"] \\
& \pi_U(\mathrm{cl}_X A) \arrow[rrr, "\tilde{\varphi}"] \arrow[ddd, "\tilde{\psi}"'] & & & \sigma_V(\mathrm{cl}_Y B) \arrow[ddd, "\lambda"] & \\
& & \mathrm{cl}_X A \arrow[ul, "\pi_U|_{\mathrm{cl}_X A}"', sloped] \arrow[r, "\varphi|_{\mathrm{cl}_X A}"] \arrow[d, "\psi|_{\mathrm{cl}_X A}"', sloped] \arrow[ul, phantom, "③", shift right=8ex] & \mathrm{cl}_Y B \arrow[ur, "\sigma_V|_{\mathrm{cl}_Y B}", sloped] \arrow[d, "\kappa \circ \nu_1", sloped] \arrow[ul, phantom, "②", shift left=4ex] \arrow[dl, phantom, "①"] & & \\
& & \mathrm{cl}_Z C \arrow[r, "\kappa \circ \nu_2"'] \arrow[dl, "\theta_W|_{\mathrm{cl}_Z C}", sloped] & \mathrm{cl}_Q R \arrow[dr, dashed, "\gamma_R"] & & \\
& \theta_W(\mathrm{cl}_Z C) \arrow[rrr, "\mu"] & & & ,_R & \\
\Theta_W \arrow[ur, phantom, sloped, "\subseteq"] & & & & &
\end{tikzcd}$$

The inner square ① accounts for the pushout property of $\mathrm{cl}_Q R$, $\kappa \circ \nu_1$ and $\kappa \circ \nu_2$. The coordinate patches π_U, σ_V and θ_W are homeomorphisms, so the sets $\pi_U(\mathrm{cl}_X A) \subseteq \Pi_U$, $\sigma_V(\mathrm{cl}_Y B) \subseteq \Sigma_V$ and $\theta_W(\mathrm{cl}_Z C) \subseteq \Theta_W$ are closed subsets of $\mathcal{S}$-objects, so they are $\mathcal{S}$-objects themselves.

The mappings $\tilde{\varphi}$ and $\tilde{\psi}$ are easy to define so as to make the trapezoids ② and ③ commute (just reverse the homeomorphism $\pi_U|_{\mathrm{cl}_X A}$). These are di-embeddings because φ and ψ are $\mathcal{S}$-manifold embeddings. By the pushout property of $\mathcal{S}$, there exists an $\mathcal{S}$-object $,_R$ with two dimaps λ and μ such that everything in solid lines commutes. Now, since $(\lambda \circ \sigma_V) \circ \varphi|_{\mathrm{cl}_X A} = (\mu \circ \theta_W) \circ \psi|_{\mathrm{cl}_X A}$, there is a unique $\gamma_R : R \to\ ,_R$ (recall that R is the pushout of $\varphi|_{\mathrm{cl}_X A}$ and $\psi|_{\mathrm{cl}_X A}$) closing the diagram. To show an inverse to γ_R, reverse the homeomorphisms $\pi_U|_{\mathrm{cl}_X A}$, $\sigma_V|_{\mathrm{cl}_Y B}$ and $\theta_W|_{\mathrm{cl}_Z W}$ and use the fact that $,_R$ is a pushout of $\tilde{\varphi}$ and $\tilde{\psi}$.

Define now the required po-manifold $\langle Q, \mathcal{R}, , , \gamma \rangle$ by taking the least cover $\mathcal{R}$ of Q, the least set , of $\mathcal{S}$-objects and the least set γ of coordinate patches such that:

- for all $U \in \mathcal{U}$, $V \in \mathcal{V}$, $W \in \mathcal{W}$ and for all B and C satisfying (2):
  - $\mathcal{R}$ contains the open set R,
  - , contains the compact space $,_R$,
  - γ contains the homeomorphism γ_R

  as constructed above;

- for any $V \in \mathcal{V}$:
  - $\mathcal{R}$ contains the open set $(\kappa \circ \nu_1)(V \setminus \varphi(X))$,
  - , contains the compact space $\sigma_V(\mathrm{cl}_Y(V \setminus \varphi(X)))$,
  - γ contains the homeomorphism

$$\sigma_V \circ (\kappa \circ \nu_1)^{-1} : \mathrm{cl}_Q((\kappa \circ \nu_1)(V \setminus \varphi(X))) \to \sigma_V(\mathrm{cl}_Y(V \setminus \varphi(X)))$$

- for any $W \in \mathcal{W}$ — symmetrically.

It still remains to be verified that:

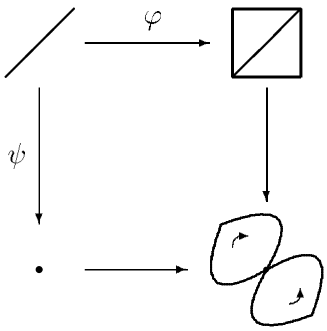
1. $\mathcal{R}$ is a cover of Q;
2. the coordinate patches in , satisfy the consistency condition (1) in Def. 2 on page 5;
3. $\kappa \circ \nu_1$ and $\kappa \circ \nu_2$ are manifold morphisms;
4. Q with $\kappa \circ \nu_1$ and $\kappa \circ \nu_2$ is the $\mathcal{S}$-manifold pushout required.

These verifications are left to the reader.

□

10 Example

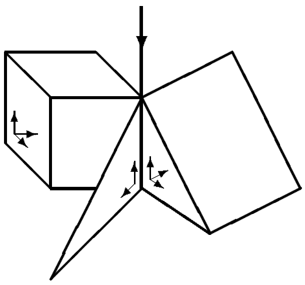
The $\mathcal{S}$-manifold embeddings φ and ψ from Thm. 9 cannot be replaced by arbitrary $\mathcal{S}$-manifold morphisms. At least not for *compact-PO*-manifolds or local po-spaces; in other words, the category *compact-PO-Man* is not *finitely cocomplete.*



To see this, consider the diagram opposite with φ — embedding a directed interval $\vec{\mathbb{I}}$ into a directed square $\vec{\mathbb{I}}^2$ as its diagonal; and ψ — contracting the interval to a point (the interval and the square are ordered from left to right and from bottom upwards). All three are respectable compact po-spaces. The topological pushout, however, has a "singularity": a point around which there does not exist a local order consistent with the circular arrows[5].

□

2.3 Blocks and po-manifolds



From the computational point of view, the most important of the $\mathcal{S}$-manifold categories is *BLCK-Man*, where *BLCK* is the category of blocks, presented below.

Informally, blocks result from glueing nice sub-po-spaces of the directed cubes $\vec{\mathbb{I}}^n$ respecting the orders. The picture opposite presents an example block consisting of $\vec{\mathbb{I}}^1$, $\vec{\mathbb{I}}^3$, a triangle which is a sub-po-space of $\vec{\mathbb{I}}^2$ and a prism which is a sub-po-space of $\vec{\mathbb{I}}^3$.

The family of blocks will now be described in more detail.

11 Definition (block)

BLCK is the least po-pattern category containing the directed cubes $\vec{\mathbb{I}}^n$ for all $n \geq 0$; its objects are called *blocks.*

12 Definition (po-manifold)

BLCK-manifolds are called *po-manifolds.* The category of po-manifolds is denoted by *PMAN*; this means *PMAN* $\stackrel{\text{def}}{=}$ *BLCK-Man*.

[5] This is not a complete argument for the non-existence of the pushout, but may easily be turned into one.

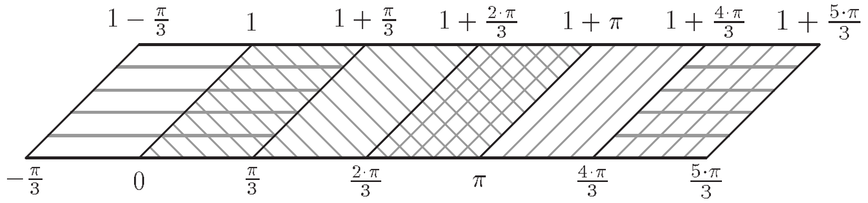
Do not confuse $\underline{\underline{\mathit{PMAN}}}$ with $\underline{\underline{\mathit{compact\text{-}PO\text{-}Man}}}$ which is equal to $\underline{\underline{\mathit{LPO}}}$ (cf. Prop. 8 on page 7).

13 Example

Take the image $\widehat{X}$ of the diagonal ribbon X from Sec. 1.1 on page 2 under the wrapping[6]

$$\begin{aligned} \rho :\ \ & X = \mathbb{R} \times \mathbb{I} \ \ \to \ \ \widehat{X} = \mathbb{S}^1 \times \mathbb{I} \\ \rho\,\langle x, y\rangle \ & \overset{\mathsf{def}}{=} \ \langle e^{i \cdot x}, y\rangle \end{aligned}$$

The result is, evidently, a local po-space. Consider its cover by three overlapping opens:



▤ $U_{-1} \overset{\mathsf{def}}{=} \rho\,(\{\langle x, y\rangle \in \mathbb{R} \times \mathbb{I} \,|\, y - \frac{2\cdot\pi}{3} < x < y + \frac{\pi}{3}\})$

▧ $U_0 \overset{\mathsf{def}}{=} \rho\,(\{\langle x, y\rangle \in \mathbb{R} \times \mathbb{I} \,|\, y < x < y + \pi\})$

▨ $U_1 \overset{\mathsf{def}}{=} \rho\,(\{\langle x, y\rangle \in \mathbb{R} \times \mathbb{I} \,|\, y + \frac{2\cdot\pi}{3} < x < y + \frac{5\cdot\pi}{3}\})$

(since $y + \frac{5\cdot\pi}{3} \equiv y - \frac{\pi}{3}$ $(\mathrm{mod}\,(2\cdot\pi))$ for all $y \in \mathbb{I}$, the left slanted side of the parallelogram is, under ρ, the same as its right slanted side).

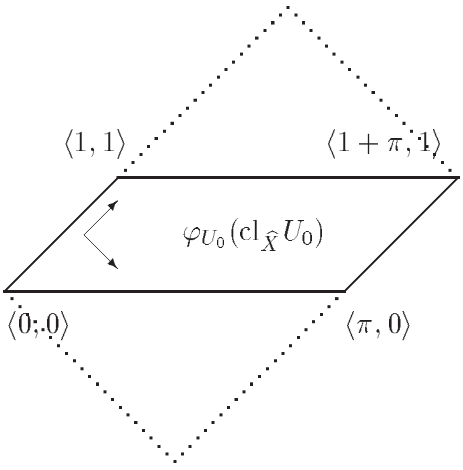
There exist coordinate patches

$$\varphi_{U_j} : \mathrm{cl}_{\widehat{X}} U_j \to \varphi_{U_j}(\mathrm{cl}_{\widehat{X}} U_j) \subseteq \vec{\mathbb{I}}^2$$

for $j \in \{-1, 0, 1\}$ defined by:

$$\begin{aligned} & \varphi_{U_j}(\rho\,\langle x, y\rangle) \overset{\mathsf{def}}{=} \\ & \quad \big\langle \tfrac{1}{\pi} \cdot (x - y - j \cdot \tfrac{2\cdot\pi}{3})\ ,\ \tfrac{1}{2+\pi} \cdot (x + y - j \cdot \tfrac{2\cdot\pi}{3})\big\rangle \end{aligned}$$

The picture opposite suggests the way φ_{U_0} embeds its domain into the unit square (do not overlook the preservation of the ordering!), the others are similar.



$\langle 1, 1\rangle$
$\langle 1+\pi, 1\rangle$
$\varphi_{U_0}(\mathrm{cl}_{\widehat{X}} U_0)$
$\langle 0, 0\rangle$
$\langle \pi, 0\rangle$

By a straightforward check, the compositions $\varphi_{j_1} \circ \varphi_{j_2}^{-1}$ are monotone, hence they are dimaps, as required by Def. 2 on page 5 of $\mathcal{S}$-manifold.

The reader may want to demonstrate on her/his own that the remaining two examples from Sec. 1 are po-manifolds as well.

□

[6] The ribbon is now positioned differently on the plane than in Sec. 1.1. In particular, coordinate x corresponds to time.

3 Are we more general than the semi-cubists?

Once we know what po-manifolds are (cf. Sec. 2), we still need to prove that they generalize semi-cubical complexes. This will also imply that they generalize higher dimensional automata and many other models for concurrency. In a nutshell, a semi-cubical complex is a blueprint for glueing cubes $\mathbb{I}^n$ (for various dimensions n) to make a po-manifold[7].

It is shown below how such a blueprint may be realized.

14 Definition (po-manifold realization)
The *po-manifold realization* is the assignment $\square$ of

- a po-manifold $\square M$ to any non self-linked semi-cubical complex M, and
- a po-manifold embedding

$$\square\chi : \square P \to \square M$$

to any semi-cubical embedding $\chi : P \to M$

defined by recursion on the total number of M's faces, as given in Fig. 1 on the next page.

15 Proposition
For any semi-cubical pushout diagram

$$\begin{tikzcd} M \arrow[r, "\varphi"] \arrow[d, "\psi"'] & P \arrow[d, "\lambda"] \\ Q \arrow[r, "\mu"'] & R \end{tikzcd}$$

where φ and ψ are embeddings, the translation

$$\begin{tikzcd} \square M \arrow[r, "\square\varphi"] \arrow[d, "\square\psi"'] & \square P \arrow[d, "\square\lambda"] \\ \square Q \arrow[r, "\square\mu"'] & \square R \end{tikzcd}$$

is a po-manifold pushout diagram.

16 Theorem (realization functor)
The po-manifold realization can be extended to a a functor $\square : \underline{\underline{SCC_0}} \to \underline{\underline{PMAN}}$ ($\underline{\underline{SCC_0}}$ is the category of finite non self-linked semi-cubical complexes) such that

1. *for any $M \in \underline{\underline{SCC_0}}$ and any $m \in M_d$:*
   - $\square I^d = \mathbb{I}^d$,
   - $\square K^{d-1} = \mathbb{K}^{d-1}$,
   - $\square(\delta_d : K^{d-1} \hookrightarrow I^d)$ *is the natural inclusion of $\mathbb{K}^{d-1}$ into $\mathbb{I}^d$;*
2. *for any semi-cubical embedding $\varphi : M \to P$, the corresponding manifold-mophism $\square\varphi : \square M \to \square P$ is a manifold-embedding;*
3. *for every semi-cubical pushout diagram*

$$\begin{tikzcd} M \arrow[r, "\varphi"] \arrow[d, "\psi"'] & P \arrow[d, "\lambda"] \\ Q \arrow[r, "\mu"'] & R \end{tikzcd}$$

where φ and ψ are embeddings, the translation

$$\begin{tikzcd} \square M \arrow[r, "\square\varphi"] \arrow[d, "\square\psi"'] & \square P \arrow[d, "\square\lambda"] \\ \square Q \arrow[r, "\square\mu"'] & \square R \end{tikzcd}$$

is a po-manifold pushout diagram.

[7] See App. B on page 18 for a short review of basic notions and notations on the semi-cubical complexes.

Case M is empty:

Put $\Box\, M \overset{\mathsf{def}}{=} \emptyset$ too.

Case $\dim M = d \geq 0$ with an $m \in M_d$:

Let M' be the complex given by $M'_d \overset{\mathsf{def}}{=} M_d \smallsetminus \{m\}$ and $M'_n \overset{\mathsf{def}}{=} M_n$ for $n \neq d$. By the induction hypothesis, for any semi-cubical embedding $\chi : P \hookrightarrow M'$,

- the po-manifolds $\Box\, P$ and $\Box\, M'$ are already constructed (by virtue of the embedding χ, this makes sense because P is "not bigger" than M'), and
- the po-manifold embedding $\Box\, \chi : \Box\, P \hookrightarrow \Box\, M'$ is already constructed.

$$\begin{tikzcd}
K^{d-1} \arrow[r, "\varphi"] \arrow[d, "\delta_d"'] \arrow[dr, phantom, "\text{push-out}"] & M' \arrow[d] \\
I^d \arrow[r] & M
\end{tikzcd}$$

Consider the semi-cubical pushout diagram from Prop. 47 on page 22. Define

- $\Box\, I^d$ as $\mathbb{I}^d$, and
- $\Box\, K^{d-1}$ as $\mathbb{K}^{d-1}$, and
- $\Box\, (\delta_d : K^{d-1} \hookrightarrow I^d)$ as the natural inclusion of $\mathbb{K}^{d-1}$ into $\mathbb{I}^d$, and
- $\Box\, M$, λ and μ as the po-manifold pushout of the embeddings $\Box\, \varphi$ and $\Box\, \delta_d$.

$$\begin{tikzcd}
\mathbb{K}^{d-1} \arrow[r, "\Box\, \varphi"] \arrow[d, "\Box\, \delta_d"'] \arrow[dr, phantom, "\text{push-out}"] & \Box\, M' \arrow[d, "\lambda"] \\
\mathbb{I}^d \arrow[r, "\mu"'] & \Box\, M
\end{tikzcd}$$

By Thm. 9 on page 8, this gives rise to the manifold pushout diagram opposite.

We still need to explain how an arbitrary semi-cubical embedding $\chi : P \hookrightarrow M$ is translated into a po-manifold embedding $\Box\, \chi : \Box\, P \hookrightarrow \Box\, M$.

If there is no face $p \in P_d$ such that $\chi\, p = m$ then the whole image of χ sits in M' and, by the induction hypothesis, $\Box\, \chi : \Box\, P \hookrightarrow \Box\, M'$ is already defined; just compose it with $\lambda : \Box\, M' \hookrightarrow \Box\, M$.

$$\begin{tikzcd}
K^{d-1} \arrow[r, "\rho"] \arrow[d, "\delta_d"'] \arrow[rr, bend left, "\varphi"] \arrow[dr, phantom, "\text{push-out}"] & P' \arrow[r, "\chi|_{P'}"] \arrow[d] & M' \arrow[d] \\
I^d \arrow[r] \arrow[rr, bend right] & P \arrow[r, "\chi"'] & M
\end{tikzcd}$$

If there is a (unique) face $p \in P_d$ with $\chi\, p = m$ then the complex P may be decomposed into a complex P' — which is P without the face p — and a d-dimensional cube. Together with χ, this yields the diagram opposite. This diagram may be partially translated to po-manifolds, as done above for M; but note that so far we have no translation for χ.

$$\begin{tikzcd}
\mathbb{K}^{d-1} \arrow[r] \arrow[d, "\Box\, \delta_d"'] \arrow[rr, bend left, "\Box\, \varphi"] \arrow[dr, phantom, "\text{push-out}"] & \Box\, P' \arrow[r] \arrow[d] & \Box\, M' \arrow[d] \\
\mathbb{I}^d \arrow[r] \arrow[rr, bend right] & \Box\, P \arrow[r, dotted] & \Box\, M
\end{tikzcd}$$

But both the small square and the large rectangle (spanned by $\Box\, \varphi$ and $\Box\, \delta_d$) are pushout diagrams. Therefore, there exists a unique morphism denoted by the dotted line, closing the diagram. Call this morphism $\Box\, \chi$; it is, clearly, a po-manifold embedding.

It is obvious that the construction of $\Box\, M$ and of $\Box\, \chi$ does not depend on the initial selection of face $m \in M_d$.

Figure 1: The recursive po-manifold realization (cf. Def. 14 on the facing page).

Proof of Thm. 16:

The construction of $\square\chi$ for a semi-cubical morphism $\chi : M \to P$ is recursive on the number of M's faces and similar to the construction in Fig. 1. For the inductive step, consider the diagram opposite. By the induction hypothesis, the po-manifold morphism $\square(\chi \circ \lambda) : \square M \to \square P$ is already defined and, since $\chi \circ \mu : I^d \to M$ is an embedding (cf. Prop. 46 on page 22), $\square(\chi \circ \mu)$ is defined too.

$$\begin{tikzcd}
K^{d-1} \arrow[r, "\varphi"] \arrow[d, "\delta_d"'] \arrow[dr, phantom, "\text{push-out}"] & M' \arrow[d, "\lambda"] & \\
I^d \arrow[r, "\mu"'] & M \arrow[dr, "\chi"] & \\
& & P
\end{tikzcd}$$

$$\begin{tikzcd}
\mathbb{K}^{d-1} \arrow[r, "\square\varphi"] \arrow[d, "\square\delta_d"'] \arrow[dr, phantom, "\text{push-out}"] & \square M' \arrow[d, "\square\lambda"] \arrow[ddr, bend left, "\square(\chi\circ\lambda)"] & \\
\mathbb{I}^d \arrow[r, "\square\mu"'] \arrow[drr, bend right, "\square(\chi\circ\mu)"'] & \square M \arrow[dr, dotted] & \\
& & \square P
\end{tikzcd}$$

Since $\square M$ is the pushout (cf. the construction in Fig. 1), there exists a unique po-manifold morphism to $\square P$ (the dotted line). Call this morphism $\square\chi$.

It is easy to show that this morphism does not depend on the original choice of $m \in M_d$; that the defined assignment is a functor; and that the requirements 1–3 are met.

□

In [5], Fajstrup, Goubault and Raussen define a functor $|\ldots| : \underline{\underline{SCC_0}} \to \underline{\underline{LPO}}$ realizing semi-cubical complexes in local po-spaces[8]. The method of the proof is different: the *stars* of the vertices are translated directly to open sets in the cover of the local po-space.

I believe the essence of the construction is the same:

17 Conjecture
The diagram of functors opposite commutes ($|\ldots|_1$ is the realization functor from [5]; $|\ldots|_2$ is the realization functor from Cor. 7 on page 7; $\square$ is the functor from Thm. 16).

$$\begin{tikzcd}
\underline{\underline{SCC_0}} \arrow[rr, "\square"] \arrow[dr, "|\cdots|_1"'] & & \underline{\underline{PMAN}} \arrow[dl, "|\cdots|_2"] \\
& \underline{\underline{LPO}} &
\end{tikzcd}$$

But I have not proven this, sorry...

4 My grudge against complexes

Sec. 1 suggests that semi-cubical complexes are not an adequate formalism for "continuous concurrency". Generalizing them just a bit, as done in Sec. 2 can make the world of difference in this respect. But this is not the only problem bugging the semi-cubical complexes.

A minor problem is the definition of semi-cubical morphisms (Def. 41 on page 21). They preserve the dimensions, which makes them unsuitable for comparisons between systems of processes. One may, for instance, want to implement a higher level process by a system of cooperating lower level processes. For the correctness of such an implementation, one would need to prove that the two systems are dihomeomorphic, or dihomotopy equivalent, or whatever — in any case, one would need morphisms going both ways and meeting some correctness requirements. But this is impossible because of the arbitrary definition which rules out any morphisms from many processes to a single process.

[8] The construction in [5] is more general in that it does not require the finiteness of the complexes. It does, however, require their non self-linkedness, as mine does.

This problem can probably be treated by substituting cubical for semi-cubical complexes (see footnote 10 on page 19).

More importantly, the notion of semi-cubical complex is very combinatorial and does not fit smoothly with the rest of the directed topology.

The simplicial approximations of topological spaces are combinatorial too. This fact does not hurt because they are accompanied by respective invariance theorems. Continuous maps between topological spaces induce homomorphisms of the respective homologies or homotopies; and homeomorphisms induce isomorphisms (cf., e.g., Munkres [8], Chap. 2; or Spanier [13], Chap. 3). Changing simplicial subdivisions of a topological space does not affect the resulting invariants. The combinatoriality of the semi-cubical subdivisions of a local po-space is not sweetened by any property of the sort: a different cubification means different everything[9].

This may lead, therefore, to two opposing viewpoints: either

1. the computational reality is best reflected by semi-cubical complexes, while local po-spaces are only an auxiliary generalization, or
2. the computational reality is best reflected by local po-spaces, while semi-cubical complexes are only a way of getting to them.

In Sec. 1, I have given some evidence against viewpoint 1. But do we really need to pass through complexes in order to get to local po-spaces, as claimed in viewpoint 2?

In fact, a number of times I have seen a similar story happening: somebody comes up with a reasonable topological conjecture about computations, then a pathological counterexample is found, and then it is hoped that the conjecture is true at least for the geometric realizations of semi-cubical complexes. In most cases, however, this involves a combinatorial rather than topological reasoning, so nobody would volunteer to write down the ugly proof.

I believe most of these reasonable conjectures are true of po-manifolds and, due to the topological nature of the concept, their proofs may be made beautiful.

A Appendix: a bird's eye view on directed topology

For the convenience of a reader without a thorough knowledge of global and local po-spaces, this is a brief reminder of the basic concepts and facts (no proofs here!). The main reference for most of the knowledge is [5] by Fajstrup, Goubault and Raussen. Wherever particular formulations depart from [5], they follow [11] by Sokołowski.

A.1 Po-spaces

18 Definition (po-space)
A *po-space* (or: a *global po-space*) consists of

- a set X,

[9] As noted by a referee, the total homology for Kan semi-cubical complexes is invariant under subdivisions.

- a topology $\mathcal{O}$ on X, and
- a partial order $\leq$ in X

such that the order $\leq$ is closed as a subset of the Cartesian product $X \times X$.

The notion of po-space appeared in [9] by Nachbin. It was further studied in [5] by Fajstrup, Goubault and Raussen and in other papers on ditopology that followed.

19 Proposition *Every po-space is Hausdorff.*

20 Proposition ($\leq$-preservation in the limit)
Assume two convergent sequences in a po-space X are given

$$x = \lim_{n\to+\infty} x_n \quad \textit{and} \quad y = \lim_{n\to+\infty} y_n$$

Assume further that $x_n \leq y_n$ for all $n \in \mathbb{N}$. Then the same inequality holds for the limit points: $x \leq y$.

21 Example
The interval $[0\,..\,1]$ with the usual order is denoted by $\vec{\mathbb{I}}$; the same interval with the trivial order (equality) is denoted by $\mathbb{I}$. They are both po-spaces.

The Cartesian product $X \times Y$ of po-spaces X and Y is a po-space with the product topology and the coordinate-wise ordering:

$$\langle x_1, y_1\rangle \leq_{X\times Y} \langle x_1, y_1\rangle \overset{\mathsf{def}}{\iff} x_1 \leq_X x_2 \,\&\, y_1 \leq_Y y_2$$

In particular, the cubes $\vec{\mathbb{I}}^n$ and $\mathbb{I}^n$ are po-spaces.

The hollow cube $\mathbb{K}^{n-1} \overset{\mathsf{def}}{=} \big\{\langle x_1, \ldots, x_n\rangle \in \mathbb{I}^n \,\big|\, \exists_i\, x_i \in \{0,1\}\big\}$ is a sub-po-space of $\mathbb{I}^n$.

□

A.2 Dimaps of po-spaces

In order to make a respectable mathematical notion, the po-spaces should be equipped with morphisms.

22 Definition (dimap and dihomeomorphism)
A function $\varphi : X \to Y$ between two partially ordered sets is *monotone* if $\varphi x \leq \varphi y$ for any $x, y \in X$ such that $x \leq y$. A *dimap* (or: a *global dimap*) is any continuous and monotone mapping $\varphi : X \to Y$ between two po-spaces. A dimap with an inverse which is also a dimap is called a *dihomeomorphism*. A dimap φ is a *di-embedding* if it is a dihomeomorphism between X and the image $\varphi(X)$.

It is obvious that po-spaces with dimaps form a category. In this report, this category is denoted by $\underline{\underline{PO}}$. The dihomeomorphisms are isomorphisms in $\underline{\underline{PO}}$. By $\underline{\underline{compact\text{-}PO}}$ is denoted the full subcategory of $\underline{\underline{PO}}$ consisting of compact po-spaces only.

23 Definition (dipath and long dipath)
A *dipath* is a dimap $\alpha : \vec{\mathbb{I}} \to X$ of the directed interval into a po-space. A *long dipath* is a dimap $\alpha : \vec{\mathbb{R}}_{\geq 0} \to X$ of the directed half-line $\vec{\mathbb{R}}_{\geq 0}$ into a po-space.

Since the monotonicity is required, dipaths correspond to the “trajectories” of points in a po-space that only “move forwards”.

A.3 Factorization of po-spaces

When a po-space is factorized, we want to make sure that

1. the resulting topological quotient is Hausdorff,
2. the resulting quotient order is a partial order, and
3. the quotient order is closed in the quotient topology.

Simple examples show that each of these requirements may fail. For this reason, I will only concentrate on the special case of glueing two po-spaces along a common sub-po-space. Given compact po-spaces X, Y and Z and a pair of di-embeddings $\varphi : X \to Y$ and $\psi : X \to Z$, define $\simeq_{\varphi\psi}$ as the least equivalence in the disjoint union $Y \oplus Z$ such that

$$\nu_1(\varphi x) \simeq_{\varphi\psi} \nu_2(\psi x) \quad \text{for all } x \in X$$

where $\nu_1 : Y \to Y \oplus Z$ and $\nu_2 : Z \to Y \oplus Z$ are the natural embeddings of the components into the disjoint union. For the ordering in the quotient $(Y \oplus Z)/\simeq_{\varphi\psi}$, take the transitive hull of the orders from Y and Z.

24 Theorem *The result $(Y \oplus Z)/\simeq_{\varphi\psi}$ of the glueing is a compact po-space.*

25 Corollary
Under the assumptions as above, the quotient

$$Q \overset{\text{def}}{=} (Y \oplus Z)/\simeq_{\varphi\psi} \quad \textit{with the dimaps} \quad \kappa \circ \nu_1 : Y \to Q \quad \textit{and} \quad \kappa \circ \nu_2 : Z \to Q$$

is a pushout of φ and ψ in compact-PO.

A.4 Local po-spaces

The rationale for the notion of local po-space is in the situations, where several partial orders combine to a "local directedness" but cannot be globally reconciled to form a partial order.

The first formal definition of local po-spaces appeared in [5] by Fajstrup, Goubault and Raussen. Subtle changes were made to the notion later — first written down in [6] by Fajstrup and Sokołowski — because we had thought it did not correctly reflect the intuitions. Later, Fahrenberg [4] showed that the two original definitions of local po-space were equivalent and our counterexample was wrong. More importantly, he gave a different simpler definition and proved its equivalence with the two preceding. This simpler notion is the one appearing in Def. 26 below. Prop. 27 comes from [4] too.

26 Definition (local po-space)
A *local po-space* is a Hausdorff space X with a *local order*, i.e., a relation $\leq\ \subseteq X \times X$ satisfying the following local po-property: every point $x \in X$ has an open neighbourhood $U_x \ni x$ such that $\langle U_x, \leq|_{U_x \times U_x} \rangle$ is a po-space.

The local order is obviously reflexive and closed in $X \times X$ but it is only "locally" anti-symmetric and transitive.

27 Proposition
A Hausdorff space X with a relation $\leq\ \subseteq X \times X$ is a local po-space iff it has a basis $\mathcal{U}$ of open sets such that $\langle U, \leq|_{U\times U}\rangle$ is a po-space for any $U \in \mathcal{U}$. Every such basis is called a po-basis.

28 Proposition (glueing a local po-space from global po-spaces)
Assume X is a Hausdorff space with a cover $\mathcal{U}$ by open sets which are po-spaces, whose partial orders are consistent:

$$x \leq_{U_1} y \iff x \leq_{U_2} y \quad \text{for all } U_1, U_2 \in \mathcal{U} \text{ and all } x, y \in U_1 \cap U_2.$$

Then the relation $x \leq y \overset{\text{def}}{\iff} \exists_{U\in\mathcal{U}}\, x \leq_U y$ is a local order in X.

The idea is that when you look at a local po-space from very close, you see a po-space.

29 Proposition *Every po-space is, in a natural way, a local po-space.*

A.5 Local dimaps

Assume X and Y are local po-spaces with the bases (as required in Prop. 27) $\mathcal{U}$ and $\mathcal{V}$, respectively.

30 Definition (local dimap and local dihomeomorphism)
A function $\varphi : X \to Y$ is said to be *locally monotone* if for any point $x \in X$ there exist open neighbourhoods $U \in \mathcal{U}$, $U \ni x$ and $V \in \mathcal{V}$, $V \ni \varphi x$ such that

$$x_1 \leq_X x_2 \quad \text{implies} \quad \varphi x_1 \leq_Y \varphi x_2 \quad \text{for all } x_1, x_2 \in U \cap \varphi^{-1}(V) \tag{3}$$

Every function $\varphi : X \to Y$ which is continuous and locally monotone will be called a *local dimap*. A local dimap $\varphi : X \to Y$ with an inverse which is also a local dimap is called a *local dihomeomorphism*. A local dimap φ is a *local dihomeomorphic embedding* if it is a local dihomeomorphism between X and $\varphi(X) \subseteq Y$.

It is obvious that local po-spaces with local dimaps form a category. In this report, this category is denoted by $\underline{\underline{LPO}}$. The local dihomeomorphisms are the isomorphisms in $\underline{\underline{LPO}}$.

Note that $\underline{\underline{PO}}$ is not a full subcategory of $\underline{\underline{LPO}}$: a local dimap between two global po-spaces is not necessarily a global dimap.

B Appendix: a bird's eye view on semi-cubical complexes

For the convenience of the reader, the definition and the basic properties of a semi-cubical complex will be recalled (no proofs here again). This follows more or less [5] by Fajstrup, Goubault and Raussen who, in turn, attribute the development of the concept to [10] by Serre and to [1] by Brown and Higgins. Some minor changes come from [11] by Sokołowski.

B.1 Semi-cubical complexes

31 Definition (semi-cubical complex)
A *semi-cubical complex* is a family $M \overset{\text{def}}{=} \{ M_n \mid n \geq 0 \}$ of sets with *face maps*

$$\partial_i^k : M_n \to M_{n-1} \quad \text{for } 1 \leq i \leq n \text{ and } k \in \{0,1\}$$

satisfying the *semi-cubical relations*:

$$\partial_i^k \circ \partial_j^\ell = \partial_{j-1}^\ell \circ \partial_i^k \quad \text{whenever } i < j \tag{4}$$

The elements of M_0 are called *vertices*; the elements of M_n are called *n-dimensional faces* of the complex (for $n \geq 0$). Whenever $(\partial_{i_1}^{k_1} \circ \ldots \circ \partial_{i_p}^{k_p})m = m'$, m' is called a *face* of m, denoted $m' \trianglelefteq m$. A *finite semi-cubical complex* is one for which the set $\bigcup M$ of faces is finite[10].

32 Definition (dimension)
If $M_d \neq \emptyset$ and $M_n = \emptyset$ for all $n > d$ then the semi-cubical complex M is said to be *d-dimensional*, denoted: $\dim M = d$. If no such d exists and at least one of the sets $M_0, M_1, M_2, \ldots$ is nonempty then the complex is said to be *infinitely dimensional*, denoted $\dim M = +\infty$. By convention, the dimension of the empty complex is -1; the empty complex is M such that $M_n = \emptyset$ for all $n \geq 0$.

33 Definition (sub-complex)
A *sub-complex* of a semi-cubical complex $M \overset{\text{def}}{=} \{ M_n \mid n \geq 0 \}$ is any semi-cubical complex $P \overset{\text{def}}{=} \{ P_n \mid n \geq 0 \}$ with $P_n \subseteq M_n$ for $n \geq 0$ and with the face maps being the restrictions of the bigger complex's face maps: $\partial_{P,i}^k = \partial_{M,i}^k|_{P_n}$ for $1 \leq i \leq n$.

34 Definition (generated complex)
Let $M \overset{\text{def}}{=} \{ M_n \mid n \geq 0 \}$ be a semi-cubical complex. For any set $F \subseteq \bigcup_{n \geq 0} M_n$ of its faces, define the *sub-complex generated* by F, denoted $G_M F$, as the smallest sub-complex of M containing F.

35 Proposition
The sub-complex generated by a given set $F \subseteq \bigcup_{n \geq 0} M_n$ consists of the following faces:

$$(G_M F)_n = \{ m' \in M_n \mid \textit{there exists an } m \in F \textit{ s.t. } m' \trianglelefteq m \ \}$$

for $n \geq 0$.

36 Example (cube)
Let d be a natural number. A canonical example of a semi-cubical complex is the *d-dimensional cube* I^d defined by

$$I_n^d \overset{\text{def}}{=} \left\{ \langle x_1, \ldots, x_d \rangle \in \left\{ 0, \frac{1}{2}, 1 \right\}^d \,\middle|\, \begin{array}{l} \text{exactly } n \text{ components are } \frac{1}{2}, \text{ i.e.,} \\ \text{card} \{ i \in \{1, \ldots, d\} \mid x_i = \frac{1}{2} \} = n \end{array} \right\}$$

[10] Semi-cubical complexes are also called *precubical sets*. A category of *cubical complexes* (or *cubical sets*) is also considered; it is richer than the semi-cubical complexes by *degeneracy maps* going up the dimension. For some relations between these categories and also for other models for concurrency, see [7] by Goubault.

and the face maps given by

$$\partial_i^k \langle x_1, \ldots, x_d \rangle \overset{\mathsf{def}}{=} \langle x_1, \ldots, \underbrace{k}_{h}, \ldots, x_d \rangle \quad \text{for } \langle x_1, \ldots, x_d \rangle \in I_n^d,\ k \in \{0,1\} \text{ and } 1 \le i \le n$$

where h is the i-th least index such that $x_h = \frac{1}{2}$, i.e.,

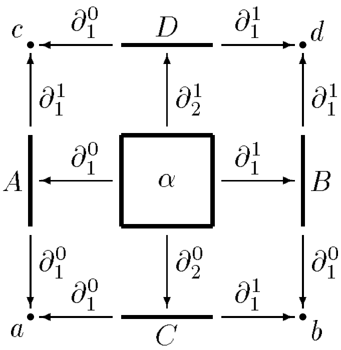
$$\operatorname{card}\left\{ j \in \{1, \ldots, h\} \,\middle|\, x_j = \tfrac{1}{2} \right\} = i$$

($\langle x_1, \ldots, \underbrace{k}_{j}, \ldots, x_n \rangle$ stands for: the tuple $\langle x_1, \ldots, x_n \rangle$ with k replaced for the j-th coordinate). Note that $I_n^d = \emptyset$ for $n > d$. In particular, the only nonempty set in the zero-dimensional cube is $I_0^0 = \left\{0, \frac{1}{2}, 1\right\}^0 = \{*\}$.

To visualize the 2-dimensional cube, denote

$$\begin{array}{lll} a \overset{\mathsf{def}}{=} \langle 0,0 \rangle & A \overset{\mathsf{def}}{=} \left\langle 0, \frac{1}{2} \right\rangle & \\ b \overset{\mathsf{def}}{=} \langle 1,0 \rangle & B \overset{\mathsf{def}}{=} \left\langle 1, \frac{1}{2} \right\rangle & \alpha \overset{\mathsf{def}}{=} \left\langle \frac{1}{2}, \frac{1}{2} \right\rangle \\ c \overset{\mathsf{def}}{=} \langle 0,1 \rangle & C \overset{\mathsf{def}}{=} \left\langle \frac{1}{2}, 0 \right\rangle & \\ d \overset{\mathsf{def}}{=} \langle 1,1 \rangle & D \overset{\mathsf{def}}{=} \left\langle \frac{1}{2}, 1 \right\rangle & \end{array}$$

The 2-dimensional cube $I^2 \overset{\mathsf{def}}{=} \{I_0^2, I_1^2, I_2^2, \ldots\}$ is given by



$$\begin{array}{ll} I_0^2 \overset{\mathsf{def}}{=} \{a, b, c, d\} & I_2^2 \overset{\mathsf{def}}{=} \{\alpha\} \\ I_1^2 \overset{\mathsf{def}}{=} \{A, B, C, D\} & I_n^2 \overset{\mathsf{def}}{=} \emptyset \quad \text{for } n > 2 \end{array}$$

and the face maps

$$\begin{cases} \partial_1^0, \partial_1^1, \partial_2^0, \partial_2^1 : I_2^2 \to I_1^2 \\ \partial_1^0 \alpha \overset{\mathsf{def}}{=} A \\ \partial_1^1 \alpha \overset{\mathsf{def}}{=} B \\ \partial_2^0 \alpha \overset{\mathsf{def}}{=} C \\ \partial_2^1 \alpha \overset{\mathsf{def}}{=} D \end{cases} \qquad \begin{cases} \partial_1^0, \partial_1^1 : I_1^2 \to I_0^2 & \\ \partial_1^0 A \overset{\mathsf{def}}{=} a & \partial_1^1 A \overset{\mathsf{def}}{=} c \\ \partial_1^0 B \overset{\mathsf{def}}{=} b & \partial_1^1 B \overset{\mathsf{def}}{=} d \\ \partial_1^0 C \overset{\mathsf{def}}{=} a & \partial_1^1 C \overset{\mathsf{def}}{=} b \\ \partial_1^0 D \overset{\mathsf{def}}{=} c & \partial_1^1 D \overset{\mathsf{def}}{=} d \end{cases}$$

□

37 Example (hollow cube)
Another example of a semi-cubical complex is the $(d-1)$-*dimensional hollow cube* K^{d-1} given by

$$K_n^{d-1} \overset{\mathsf{def}}{=} \begin{cases} I_n^d & \text{for } n \le d-1 \\ \emptyset & \text{for } n \ge d \end{cases}$$

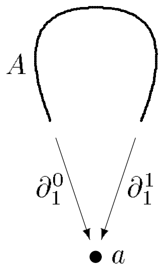
This is, clearly, a sub-complex of I^d generated as follows:

$$K^{d-1} = G_{I^d}(\Delta m) \quad \text{where } m \in I_d^d \text{ and } \quad \Delta m \overset{\mathsf{def}}{=} \left\{ \partial_i^k m \,\middle|\, 1 \le i \le d \;\&\; k \in \{0,1\} \right\}$$

□

38 Definition (non self-linkedness)
A semi-cubical complex is *non self-linked* if $\partial_i^k m = \partial_j^\ell m$ implies $k = \ell$ and $i = j$ for any $m \in \bigcup M$.

39 Example
The cubes I^d and K^{d-1} are non self-linked.



An example of a self-linked semi-cubical complex is M with

- $M_0 = \{a\}$ — a single vertex,
- $M_1 = \{A\}$ — a single face in dimension 1,
- $M_n = \emptyset$ for $n \geq 2$,
- $\partial_1^0 A = \partial_1^1 A = a$.

□

The non self-linkedness guarantees that, informally speaking, the complex has no loops consisting of a single face.

40 Proposition *A sub-complex of a non self-linked complex is non self-linked.*

B.2 Semi-cubical morphisms

41 Definition (semi-cubical morphism)
A *semi-cubical morphism* between two semi-cubical complexes M and P is a family $\{\varphi_n : M_n \to P_n \mid n \geq 0\}$ of maps commuting with the face maps:

$$\varphi_n \circ \partial^k_{M,i} = \partial^k_{P,i} \circ \varphi_{n+1}$$

$$\begin{tikzcd}
\vdots \arrow[d] & \vdots \arrow[d] \\
M_{n+1} \arrow[r, "\varphi_{n+1}"] \arrow[d, "\partial^k_{M,i}"'] & P_{n+1} \arrow[d, "\partial^k_{P,i}"] \\
M_n \arrow[r, "\varphi_n"] \arrow[d] & P_n \arrow[d] \\
\vdots & \vdots
\end{tikzcd}$$

The category of semi-cubical complexes with the semi-cubical morphisms is denoted by $\underline{\underline{SCC}}$. The full subcategory of $\underline{\underline{SCC}}$ consisting of the finite non self-linked semi-cubical complexes is denoted by $\underline{\underline{SCC_0}}$.

Note that the semi-cubical morphisms must preserve dimensions.

42 Definition (embedding)
A morphism $\varphi : M \to P$ between two semi-cubical complexes is a *semi-cubical embedding* if it is an isomorphism in $\underline{\underline{SCC}}$ between M and the complex $G_P\{\varphi\, m \mid m \in \bigcup_{n=0}^{\infty} M_n\}$.

43 Proposition
Given a semi-cubical complex M and its arbitrary sub-complex P, the natural inclusion of P into M is a semi-cubical embedding.

44 Corollary
The natural inclusion $\delta_d : K^{d-1} \hookrightarrow I^d$ is a semi-cubical embedding.

45 Proposition
Let M be a d-dimensional non self-linked complex and let $m \in M_d$ be one of its top-dimensional faces. Then

1. *the sub-complex $G_M\{m\}$ is isomorphic to the cube I^d, and*
2. *the sub-complex $G_M(\Delta m)$ (cf. Ex. 37) is isomorphic to the hollow cube K^{d-1}, and*

3. *M is the semi-cubical pushout of the embedding $\delta_d : K^{d-1} \hookrightarrow I^d$ and the embedding $\varphi : K^{d-1} \simeq G_M(\Delta m) \hookrightarrow M'$, where $M'_d \overset{\mathsf{def}}{=} M_d \setminus \{m\}$ and $M'_n \overset{\mathsf{def}}{=} M_n$ for $n \neq d$.*

Note that the assumption in Prop. 45 about the non self-linkedness of M cannot be skipped.

46 Proposition
An arbitrary semi-cubical morphism $\varphi : I^{d-1} \to M$, where M is non self-linked, is an embedding.

47 Proposition
For every non empty complex $M \in \underline{\underline{SCC_0}}$ and every top-dimensional face $m \in M_{\dim M}$, M is the pushout of two embeddings, see the diagram opposite. In this diagram, $d \overset{\mathsf{def}}{=} \dim M$; M' is given by $M'_d \overset{\mathsf{def}}{=} M_d \setminus \{m\}$ and $M'_n \overset{\mathsf{def}}{=} M_n$ for $n \neq d$; and $\varphi : K^{d-1} \simeq G_M(\Delta m) \hookrightarrow M'$ (cf. Prop. 45 on the page before).

$$\begin{tikzcd}
K^{d-1} \arrow[r, "\varphi"] \arrow[d, "\delta_d"'] \arrow[dr, phantom, "\text{push-out}"] & M' \arrow[d] \\
I^d \arrow[r] & M
\end{tikzcd}$$

References

[1] Brown, R., and Higgins, P. J. On the algebra of cubes. *Journal of Pure and Applied Algebra*, 21 (1981), 233–260.

[2] David, R., and Alla, H. Continuous Petri nets. In *8th European Workshop on Applications and Theory of Petri Nets* (Zaragoza, 1987).

[3] David, R., and Alla, H. On hybrid Petri nets. In *Discrete Event Dynamic Systems: Theory and Applications*, 11. Kluwer Academic Publishers, Boston, 2001, pp. 9–40.

[4] Fahrenberg, U. Ditopology. In *Geometry and Topology* by U. Fahrenberg and J. Leth, part 2, pp. 47–61. Student Project Report in Dept. of Mathematical Sciences, Aalborg University, Denmark, January 2002. Available as `http://www.math.auc.dk/~uli/mat5/gt.ps`.

[5] Fajstrup, L., Goubault, E., and Raussen, M. Algebraic topology and concurrency. Tech. Rep. R-99-2008, Department of Mathematical Sciences, Aalborg University, 1999. Available as `http://www.math.auc.dk/research/reports/1999.htm`.

[6] Fajstrup, L., and Sokołowski, S. Infinitely running concurrent processes with loops from a geometric viewpoint. In *GETCO 2000 — GEometry and Topology in COncurrency theory 2000, Proc. of a satellite workshop to Concur'2000, Pennsylvania State University* (2000), no. 39:2 in Electronic Notes in Theoretical Computer Science. Available as `http://www.elsevier.nl/gej-ng/31/29/23/66/27/show/Products/notes/index.htt`.

[7] Goubault, E. Cubical sets are generalized transition systems. In *LICS'02* (2002). (submitted).

[8] Munkres, J. R. *Elements of Algebraic Topology.* Addison-Wesley Publishing Co., 1984.

[9] Nachbin, L. *Topology and Order.* Van Nostrand, Princeton, New Jersey, 1965.

[10] Serre, J. P. *Homologie singulière des espaces fibrés. Applications.* PhD thesis, Ecole Normale Supérieure, 1951.

[11] Sokołowski, S. Introduction to directed topology. In preparation, 2002.

[12] Sokołowski, S. Investigation of concurrent processes by means of homotopy functors. To appear in *MSCS*, 2002. Available as `ftp://ftp.ipipan.gda.pl/stefan/reports/61-conc-hom.ps.gz`.

[13] Spanier, E. H. *Algebraic Topology.* McGraw-Hill Book Company, 1966.

[14] van Glabbeek, R. Bisimulations for higher dimensional automata. E-mail communication, Stanford, 1991. Available as `http://theory.stanford.edu/~rvg/hda`.

Electronic Notes in Theoretical Computer Science 52 No. 2 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume52.html 4 pages

A Convenient Category for The Homotopy Theory of Concurrency (Abstract)

Philippe Gaucher

Institut de Recherche Mathématique Avancée
ULP et CNRS
7 rue René Descartes
67084 Strasbourg
France
gaucher@math.u-strasbg.fr

1 The category of flows

This paper is a long abstract of [4].

One has introduced in [8] the category of *globular CW-complexes* **glCW** and an equivalence relation on it called *dihomotopy.*

Globular complexes are sufficient to model all higher dimensional automata (HDA). Indeed there is a implementation (see [1]) of the semantics of a real concurrent language in terms of *precubical sets*, demonstrating the relevance of this approach, and a functor from the category of precubical sets to that of globular CW-complexes.

Dihomotopy is an equivalence relation which preserves computer-scientific properties of globular CW-complexes (deadlocks, unreachable states, schedules of execution, final and initial points, serializability) [5] [7]. So one can work directly in the quotient category for the study of these computer-scientific properties.

However the category of globular CW-complexes does not have any "good" mathematical property. So we introduce in [4] a new category of *flows* **Flow**, a new equivalence relation called *weak dihomotopy*, and a functor *cat* : **glCW** → **Flow** such that

(i) The functor *cat* induces an equivalence of categories between the globular CW-complexes up to dihomotopy and the flows up to weak dihomotopy [1].

(ii) The category **Flow** is complete and cocomplete.

[1] The proof of this fact is more than 100-page long and requires a lot of technical works both in algebraic topology and in category theory.

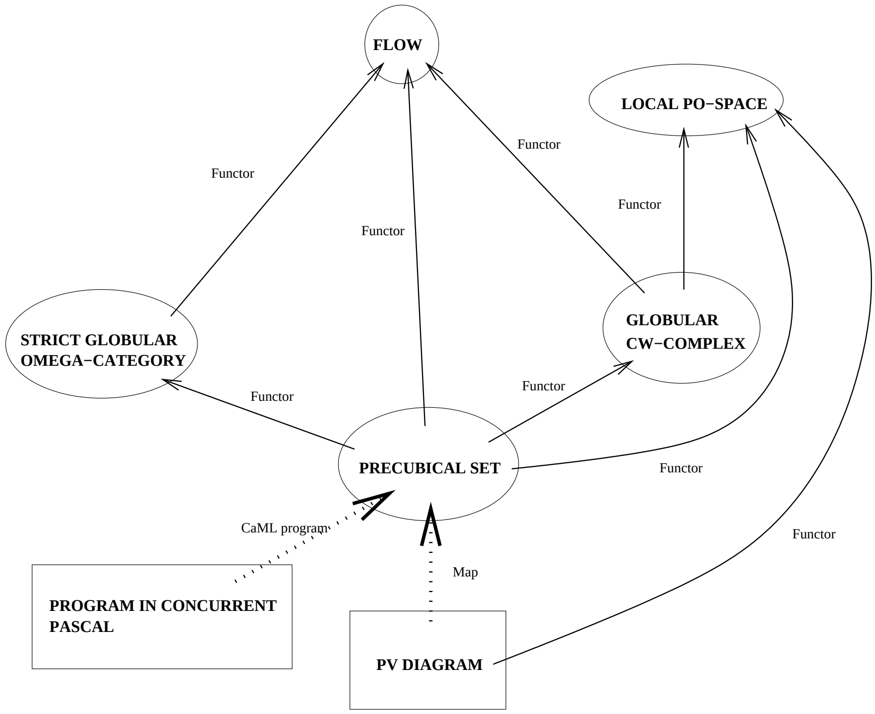


Fig. 1. Comparison of geometric models of HDA

(iii) The category **Flow** has a model structure (an abstract setting for doing homotopy which is introduced by Quillen in [10]) explaining partially the behaviour of dihomotopy.

(iv) The category **Flow** is not cartesian closed but does have a closed monoidal structure called *tensor product* corresponding to the interleaving of asynchronous processes ; in particular the tensor product of the flow corresponding to the m-dimensional cube by the flow corresponding to the n-dimensional cube is equal to the flow corresponding to the $(m + n)$-cube.

Definition 1.1 *A flow X is a (compactly generated) topological space $\mathbb{P}X$ together with a discrete space X^0 equipped with three continuous maps $s : \mathbb{P}X \to X^0$, $t : \mathbb{P}X \to X^0$ and $* : \{(x,y) \in \mathbb{P}X \times \mathbb{P}X, tx = sy\} \to \mathbb{P}X$ satisfying $(x*y)*z = x*(y*z)$, $s(x*y) = sx$, $t(x*y) = ty$. The elements of $\mathbb{P}X$ are called the non-constant execution paths of X.*

So the category of flows is at the same time mathematically rich enough and not too big modulo weak dihomotopy. It is then a convenient framework to study HDA and related problems.

2 Comparison with other geometric approaches

Figure 1 is a recapitulation of the other existing geometric approaches. Surely a lot of people by reading these pages will wonder whether there are links

Mathematical setting	Question (i)	Question (ii)	Question (iii)
precubical set	hopeless	yes	yes
PV diagram	no	no	meaningless
strict ω-category	yes	yes	yes
local po-space	no	no	meaningless
d-space	no	yes	meaningless
globular CW-complex	yes	no	meaningless
flow	yes	yes	meaningless

Table 1
Comparison of all geometric approaches for dihomotopy

with other approaches and maybe they will also wonder why I gave up the ω-categorical setting. I do not want to be polemical in this conclusion.

In this paper, we have introduced a new category (the category of flows) whose weak dihomotopy classes are exactly the dihomotopy classes of globular CW-complexes. So this new setting is still relevant for the study of HDA. Its main advantage is that the new category is complete and cocomplete.

We can by this way get rid of the combinatorial complexity of the strict ω-category setting as for instance the so-called thin elements conjecture (see conjectures in [6]). Indeed it is even not yet possible to prove that the branching homology of the n-cube vanishes in strictly positive dimension while it is trivial in the setting of globular CW-complexes. This means that we cannot yet prove in the strict ω-category setting that a HDA consisting of a n-dimensional transition (i.e. the concurrent execution of n 1-transitions) does not contain any non-deterministic branching !

The mathematical properties of the different frameworks are compared in Table 1. The questions are for each theoretical framework :

(i) Does one know a global notion of dihomotopy for the whole category ? By dihomotopy, one means an equivalence relation which preserves the computer-scientific properties of the corresponding HDA in the same equivalence class.

(ii) Is the category complete, cocomplete ? Does it have a closed monoidal structure, if possible representing something in computer science ?

(iii) Are there highly complicated combinatorial questions which prevent from carrying out the simplest homological calculations ?

The two first lines of the table treat the old cases of precubical sets and of PV diagrams [2]. They can be viewed as a subcategory of that of local po-spaces [3]. PV diagrams present the same drawbacks as the local po-space setting. The precubical set is the worst theoretical framework: there is even not enough morphism to take into account dihomotopy (see Introduction in [6]).

The answer for Question 1 is positive for the strict ω-category setting since it is easy to find the right notion of dihomotopy for these objects starting from what happens with flows.

The answer for Question 2 is negative for the local po-space setting. However we can make the conjecture that it could be possible to embed the category of compact local po-spaces in the category of flows.

To conclude, let us mention Grandis's approach [9]: he introduces another geometric setting (the one of d-spaces) having nice categorical properties. However the equivalence relations he defines do not seem to preserve the computer-scientific properties of the corresponding HDA because they contract the oriented segment. So Grandis's notion of d-homotopy does not correspond to our notion of dihomotopy. This is the reason why the answer for Question 1 is negative for the d-space framework even if d-homotopy could be also relevant for computer science (after bipointing the d-space for instance).

References

[1] R. Cridlig. Implementing a static analyzer of concurrent programs: Problems and perspectives. In *Logical and Operational Methods in the Analysis of Programs and Systems*, pages 244–259, 1996.

[2] E.W. Dijkstra. *Cooperating Sequential Processes.* Academic Press, 1968.

[3] L. Fajstrup, E. Goubault, and M. Raußen. Algebraic topology and concurrency. preprint, 1998.

[4] P. Gaucher. A Convenient Category for The Homotopy Theory of Concurrency. arXiv:math.AT/0201252.

[5] P. Gaucher. From concurrency to algebraic topology. In *Electronic Notes in Theoretical Computer Science*, volume 39, page 19pp, 2000.

[6] P. Gaucher. Combinatorics of branchings in higher dimensional automata. *Theory and Applications of Categories*, 8(12):pp.324–376, 2001.

[7] P. Gaucher. Investigating The Algebraic Structure of Dihomotopy Types. to appear in Electronic Notes in Computer Science, 2001. arxiv:math.AT/0105049.

[8] P. Gaucher and E. Goubault. Topological Deformation of Higher Dimensional Automata, 2001. arXiv:math.AT/0107060, to appear in Homology, Homotopy and Applications.

[9] M. Grandis. Directed homotopy theory, I. the fundamental category. preprint, 2001.

[10] D. G. Quillen. *Homotopical algebra.* Springer-Verlag, Berlin, 1967.

Recent BRICS Notes Series Publications

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 pp.

NS-02-4 Daniel Gudbjartsson, Anna Ingólfsdóttir, and Augustin Kong. *An BDD-Based Implementation of the Allegro Software*. August 2002. 2 pp.

NS-02-3 Walter Vogler and Kim G. Larsen, editors. *Preliminary Proceedings of the 3rd International Workshop on Models for Time-Critical Systems, MTCS '02,* (Brno, Czech Republic, August 24, 2002), August 2002. vi+141 pp.

NS-02-2 Zoltán Ésik and Anna Ingólfsdóttir, editors. *Preliminary Proceedings of the Workshop on Fixed Points in Computer Science, FICS '02,* (Copenhagen, Denmark, July 20 and 21, 2002), June 2002. iv+81 pp.

NS-02-1 Anders Møller and Michael I. Schwartzbach. *Interactive Web Services with Java: JSP, Servlets, and JWIG.* April 2002. 99 pp.

NS-01-8 Anders Møller and Michael I. Schwartzbach. *The XML Revolution (Revised)*. December 2001. 186 pp. This revised and extended report superseeds the earlier BRICS Report NS-00-8.

NS-01-7 Patrick Cousot, Lisbeth Fajstrup, Eric Goubault, Jeremy Gunawardena, Maurice Herlihy, Martin Raußen, and Vladimiro Sassone, editors. *Preliminary Proceedings of the Workshop on Geometry and Topology in Concurrency Theory, GETCO '01,* (Aalborg, Denmark, August 25, 2001), August 2001. vi+97 pp.

NS-01-6 Luca Aceto and Prakash Panangaden, editors. *Preliminary Proceedings of the 8th International Workshop on Expressiveness in Concurrency, EXPRESS '01,* (Aalborg, Denmark, August 20, 2001), August 2001. vi+139 pp.

NS-01-5 Flavio Corradini and Walter Vogler, editors. *Preliminary Proceedings of the 2nd International Workshop on Models for Time-Critical Systems, MTCS '01,* (Aalborg, Denmark, August 25, 2001), August 2001. vi+ 127pp.