A Category of Higher-Dimensional Automata

Uli Fahrenberg

Department of Mathematical Sciences
Aalborg University

Foundations of Software Science
and Computation Structures
Edinburgh, 6 April 2005

Introduction

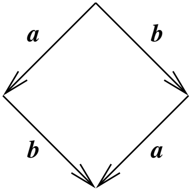
1 Introduction
- Parallelism vs. Concurrency
- Higher-Dimensional Automata
- The “van Glabbeek Hierarchy”
- The Link to Geometry
- Why is This Interesting

2 Simulation and Bisimulation

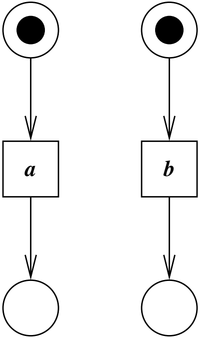
3 The Geometry of HDA

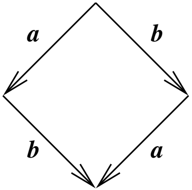
4 Bisimulation up to Equivalence

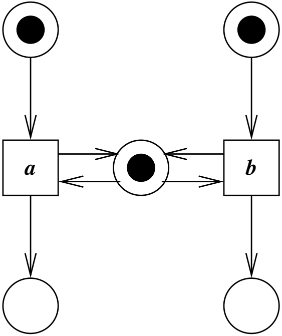
Parallelism vs. Concurrency



Parallelism vs. Concurrency



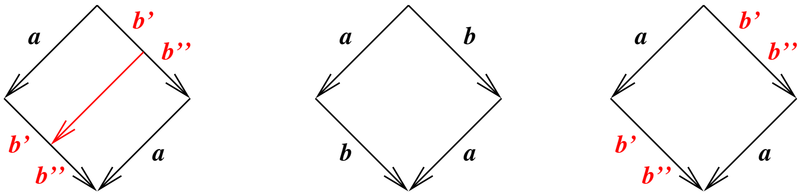




$a || b$

$a.b + b.a$

Parallelism vs. Concurrency

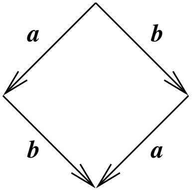


Action refinement

$a||b$ $a.b + b.a$

Parallelism vs. Concurrency

$D(a||b) =$
$\max(D(a), D(b))$



$D(a.b + b.a) =$
$D(a) + D(b)$

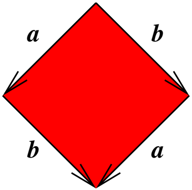
Real-time systems

$a||b$

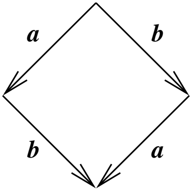
$a.b + b.a$

Parallelism vs. Concurrency

Solution:



$a||b$

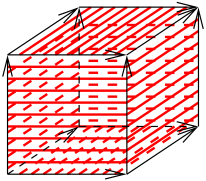


$a.b + b.a$

Parallelism vs. Concurrency

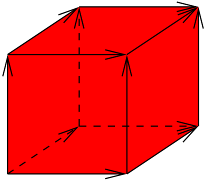
One dimension up:

- Three actions, any two of them in parallel:



(Think of three users sharing two printers.)

- Three actions in parallel:

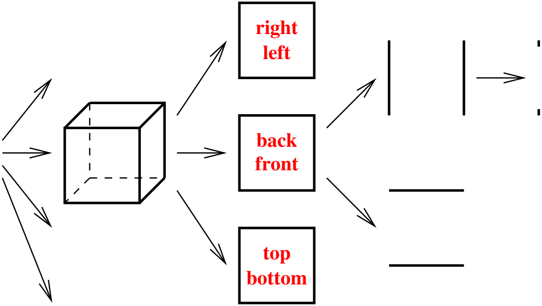


Higher-Dimensional Automata

So a higher-dimensional automaton is a pointed precubical set

$$A = \{A_n\}$$

$$\delta_i^0, \delta_i^1 : A_n \to A_{n-1} \quad (i = 1, \dots, n)$$



(The point $* \in A_0$ is the initial state.)

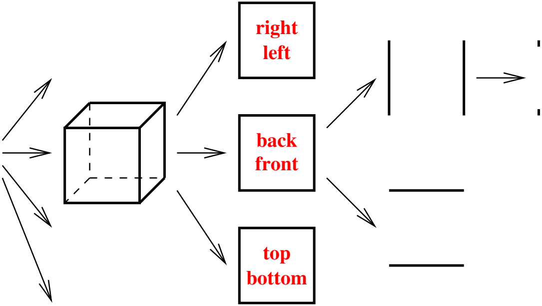
[Serre 1951; Pratt, van Glabbeek 1991]

Higher-Dimensional Automata

So a higher-dimensional automaton is a pointed precubical set

$$A = \{A_n\}$$

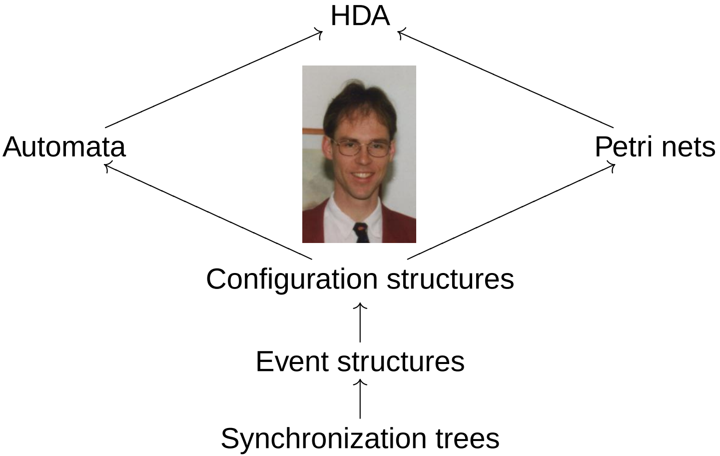
$$\delta_i^0, \delta_i^1 : A_n \to A_{n-1} \quad (i = 1, \dots, n)$$



(The point $* \in A_0$ is the initial state.)

[Serre 1951; Pratt, van Glabbeek 1991]

The “van Glabbeek Hierarchy”



arrows = embeddings up to history preserving bisimulation

[van Glabbeek 2004]

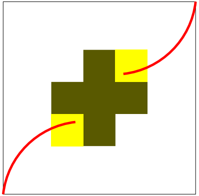
The Link to Geometry

Geometric realisation:

$$\text{precubical set } A \longrightarrow \text{topological space } |A|$$

The geometry of $|A|$ gives information about the behaviour of the HDA A:

HDA A	Space $|A|$
Mutual exclusion	Hole
Deadlock	Upper corner
Unreachable state	Lower corner
etc.	



Papers by Goubault, Fajstrup, Raussen, . . .

The Link to Geometry

Geometric realisation is a functor:

$$\begin{array}{ccc} A & & |A| \\ f\downarrow & \longrightarrow & \downarrow|f| \\ B & & |B| \end{array}$$

My contribution:

HDA-map f	continuous function $|f|$
Property x	Property x'

The Link to Geometry

Geometric realisation is a functor:

$$\begin{array}{c} A \\ f \big\downarrow \\ B \end{array} \quad \longrightarrow \quad \begin{array}{c} |A| \\ \big\downarrow |f| \\ |B| \end{array}$$

My contribution:

HDA-map f	continuous function $|f|$
bisimulation	path-lifting
bisimulation up to equivalence	path-lifting up to homotopy

The Link to Geometry

Geometric realisation is a functor:

$$\begin{array}{c} A \\ \downarrow f \\ B \end{array} \longrightarrow \begin{array}{c} |A| \\ \downarrow |f| \\ |B| \end{array}$$

My contribution:

HDA-map f	continuous function $|f|$
bisimulation	path-lifting
bisimulation up to equivalence	path-lifting up to homotopy

Why is This Interesting

HDA-map f	continuous function $\|f\|$
bisimulation	path-lifting
bisimulation up to equivalence	path-lifting up to homotopy

- Topology is good at showing negative properties
- So the above should be useful for deciding that two given HDA are not bisimilar

Simulation and Bisimulation

1. Introduction
2. Simulation and Bisimulation
   - Morphisms of HDA
   - Bisimulation
3. The Geometry of HDA
4. Bisimulation up to Equivalence

Morphisms of HDA

Morphisms of HDA should be simulations:

$A \to B$ iff whatever A can compute, B can compute, too.

So what is a computation?

[van Glabbeek 1991]

So simulations are just morphisms of pointed precubical sets:

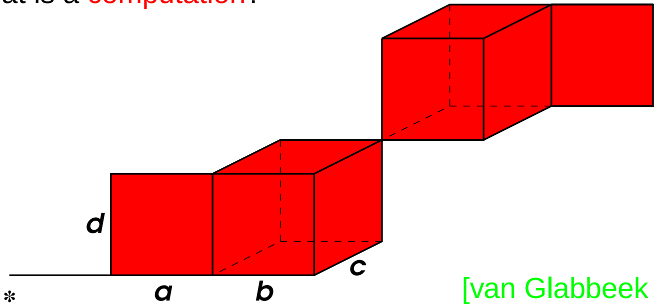
$$f : A \to B \quad \text{is} \quad f = \{f_n : A_n \to B_n\} \quad \text{s.t.} \quad \delta_i^\nu \circ f_n = f_{n-1} \circ \delta_i^\nu$$

Morphisms of HDA

Morphisms of HDA should be simulations:

$A \to B$ iff whatever A can compute, B can compute, too.

So what is a computation?



[van Glabbeek 1991]

So simulations are just morphisms of pointed precubical sets:

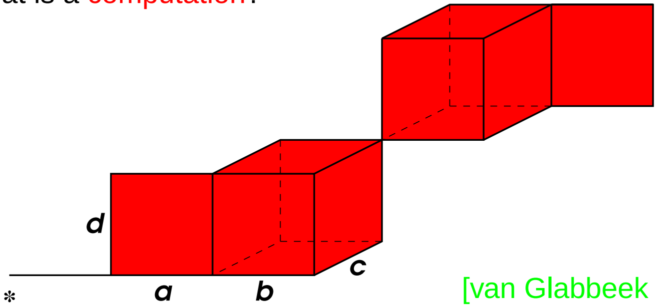
$f : A \to B$ is $f = \{f_n : A_n \to B_n\}$ s.t. $\delta_i^\nu \circ f_n = f_{n-1} \circ \delta_i^\nu$

Morphisms of HDA

Morphisms of HDA should be simulations:

$A \to B$ iff whatever A can compute, B can compute, too.

So what is a computation?



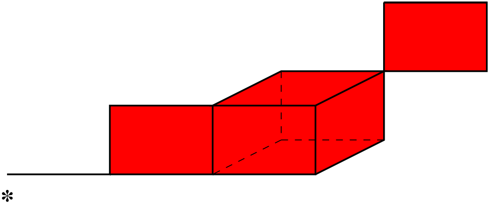
[van Glabbeek 1991]

So simulations are just morphisms of pointed precubical sets:

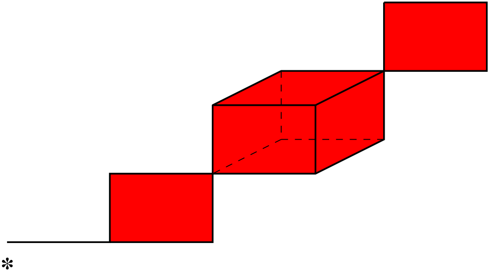
$$f : A \to B \quad \text{is} \quad f = \{f_n : A_n \to B_n\} \quad \text{s.t.} \quad \delta_i^\nu \circ f_n = f_{n-1} \circ \delta_i^\nu$$

Other People, Other Computations . . .

Me:



Cattani/Sassone 1996, Worytkiewicz 2004:



Labels, Compositions, etc.

- Labeled HDA ✓
- Idle transitions ✓
- Compositions ✓

Labels, Compositions, etc.

- Labeled HDA ✓
- Idle transitions ✓
- Compositions ✓

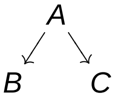
Bisimulation

Two HDA A, B are bisimilar if whatever A can compute, B can also compute, and *vice versa*.

So a morphism $f : A \to B$ is an open map if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to the computation in B.

(For simplicity, we ignore reachability issues: For this talk, all cubes are assumed to be reachable by a computation.)

And two HDA B, C are bisimilar if there are open maps [Joyal, Nielsen, Winskel 1996]



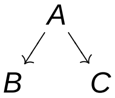
Bisimulation

Two HDA A, B are bisimilar if whatever A can compute, B can also compute, and *vice versa*.

So a morphism $f : A \to B$ is an open map if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to the computation in B.

(For simplicity, we ignore reachability issues: For this talk, all cubes are assumed to be reachable by a computation.)

And two HDA B, C are bisimilar if there are open maps [Joyal, Nielsen, Winskel 1996]



Bisimulation

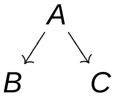
Two HDA A, B are bisimilar if whatever A can compute, B can also compute, and *vice versa*.

So a morphism $f : A \to B$ is an open map if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to the computation in B.

Or equivalently, if

$$\forall\, a \in A, \forall\, c' \in B : f(a) = \delta_i^0 c',$$
$$\exists\, c \in A : c' = f(c), a = \delta_i^0 c$$

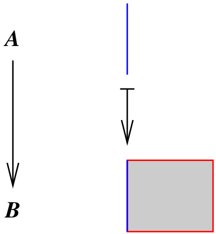
And two HDA B, C are bisimilar if there are open maps [Joyal, Nielsen, Winskel 1996]



Bisimulation

Two HDA A, B are bisimilar if whatever A can compute, B can also compute, and *vice versa*.

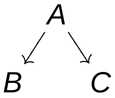
So a morphism $f : A \to B$ is an open map if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to the computation in B.



Or equivalently, if

$$\forall\, a \in A, \forall\, c' \in B : f(a) = \delta_i^0 c',$$
$$\exists\, c \in A : c' = f(c), a = \delta_i^0 c$$

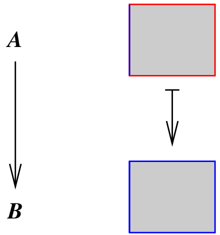
And two HDA B, C are bisimilar if there are open maps [Joyal, Nielsen, Winskel 1996]



Bisimulation

Two HDA A, B are bisimilar if whatever A can compute, B can also compute, and *vice versa*.

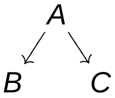
So a morphism $f : A \to B$ is an open map if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to the computation in B.



Or equivalently, if

$$\forall\, a \in A, \forall\, c' \in B : f(a) = \delta_i^0 c',$$
$$\exists\, c \in A : c' = f(c), a = \delta_i^0 c$$

And two HDA B, C are bisimilar if there are open maps



[Joyal, Nielsen, Winskel 1996]

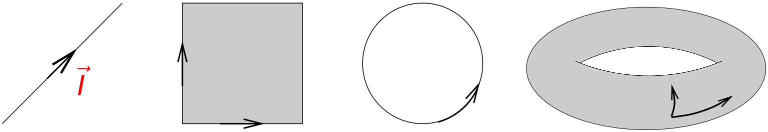
The Geometry of HDA

1. Introduction
2. Simulation and Bisimulation
3. The Geometry of HDA
   - Local po-spaces
   - Directed Maps
   - The Main Result
4. Bisimulation up to Equivalence

Local po-spaces

The geometric realisation of a precubical set is a local po-space; a topological space X with a relation $\leq$ which is

- reflexive,
- antisymmetric,
- *locally* transitive, and *locally* closed.



Local po-spaces

The geometric realisation of a precubical set is a local po-space; a topological space X with a relation $\leq$ which is reflexive, antisymmetric, and locally transitive and closed.

Geometric realisation of precubical set A:

$$|A| = \bigsqcup_{n \in \mathbb{N}} A_n \times [0,1]^n / \equiv$$

where $\equiv$ is the equivalence induced by

$$(\delta_i^\nu a; t_1, \dots, t_{n-1}) \equiv (a; t_1, \dots, t_{i-1}, \nu, t_i, \dots, t_{n-1})$$

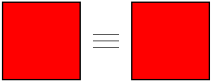
and $\leq$ is induced by the natural order on the cubes $[0,1]^n$.

Local po-spaces

The geometric realisation of a precubical set is a local po-space; a topological space X with a relation $\leq$ which is reflexive, antisymmetric, and locally transitive and closed.

Geometric realisation of precubical set A:

$$|A| = \bigsqcup_{n \in \mathbb{N}} A_n \times [0,1]^n / \equiv$$



where $\equiv$ is the equivalence induced by

$$(\delta_i^\nu a; t_1, \dots, t_{n-1}) \equiv (a; t_1, \dots, t_{i-1}, \nu, t_i, \dots, t_{n-1})$$

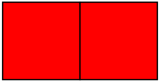
and $\leq$ is induced by the natural order on the cubes $[0,1]^n$.

Local po-spaces

The geometric realisation of a precubical set is a local po-space; a topological space X with a relation $\leq$ which is reflexive, antisymmetric, and locally transitive and closed.

Geometric realisation of precubical set A:

$$|A| = \bigsqcup_{n \in \mathbb{N}} A_n \times [0,1]^n / \equiv$$



where $\equiv$ is the equivalence induced by

$$(\delta_i^\nu a; t_1, \dots, t_{n-1}) \equiv (a; t_1, \dots, t_{i-1}, \nu, t_i, \dots, t_{n-1})$$

and $\leq$ is induced by the natural order on the cubes $[0,1]^n$.

Directed Maps

A dimap $f : X \to Y$ is a mapping which is continuous and locally increasing:

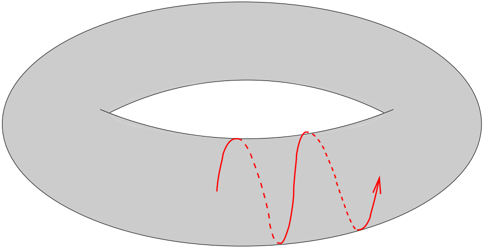
$$\forall x \in X, \exists U \ni x : \forall x_1 \leq x_2 \in U, f(x_1) \leq f(x_2) \in Y$$

Directed Maps

A dimap $f : X \to Y$ is a mapping which is continuous and locally increasing:

$$\forall x \in X, \exists U \ni x : \forall x_1 \leq x_2 \in U, f(x_1) \leq f(x_2) \in Y$$

A dipath in X is a dimap $\vec{I} \to X$.



Directed Maps

A dimap $f : X \to Y$ is a mapping which is continuous and locally increasing:

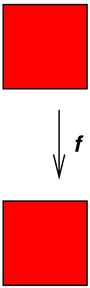
$$\forall\, x \in X, \exists\, U \ni x : \forall\, x_1 \leq x_2 \in U, f(x_1) \leq f(x_2) \in Y$$

Geometric realisation of precubical set A:

local po-space $|A| = \bigsqcup_{n \in \mathbb{N}} A_n \times [0,1]^n \big/ \equiv$

Geometric realisation of precubical map $f : A \to B$:

dimap $|f|(a; t_1, \dots, t_n) = (f(a); t_1, \dots, t_n)$



Directed Maps

A dimap $f : X \to Y$ is a mapping which is continuous and locally increasing:

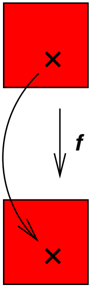
$$\forall x \in X, \exists U \ni x : \forall x_1 \leq x_2 \in U, f(x_1) \leq f(x_2) \in Y$$

Geometric realisation of precubical set A:

local po-space $|A| = \bigsqcup_{n \in \mathbb{N}} A_n \times [0,1]^n / \equiv$

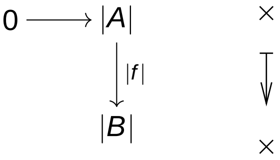
Geometric realisation of precubical map $f : A \to B$:

dimap $|f|(a; t_1, \dots, t_n) = (f(a); t_1, \dots, t_n)$



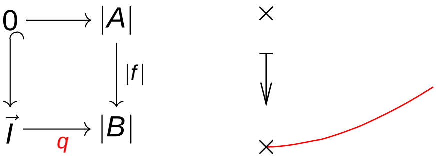
The Main Result

Theorem: $f: A \to B$ is an **open map** if and only if $|f|: |A| \to |B|$ has the **dipath-lifting** property



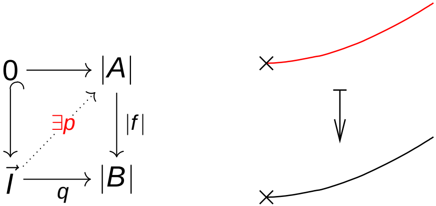
The Main Result

Theorem: $f : A \to B$ is an open map if and only if $|f| : |A| \to |B|$ has the dipath-lifting property



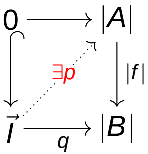
The Main Result

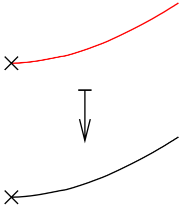
Theorem: $f : A \to B$ is an **open map** if and only if $|f| : |A| \to |B|$ has the **dipath-lifting** property



The Main Result

Theorem: $f : A \to B$ is an open map if and only if $|f| : |A| \to |B|$ has the dipath-lifting property





– connection to (directed) fibrations, obstruction theory, etc.

So What?

So two HDA B, C are bisimilar if and only if there is a diagram

$$\begin{array}{ccc} & A & \\ {}^{f}\swarrow & & \searrow^{g} \\ B & & C \end{array}$$

where $|f|$ and $|g|$ are dipath-lifting dimaps.

Enter Topology: Provide an algebraic invariant β such that if B and C are connected by a diagram like above, then $\beta(B) = \beta(C)$. This is future work.

Algorithm: Given two HDA B, C, compute $\beta(B)$ and $\beta(C)$. If $\beta(B) \neq \beta(C)$, then B and C are not bisimilar.

So What?

So two HDA B, C are bisimilar if and only if there is a diagram

$$B \xleftarrow{f} A \xrightarrow{g} C$$

where $|f|$ and $|g|$ are dipath-lifting dimaps.

Enter Topology: Provide an algebraic invariant β such that if B and C are connected by a diagram like above, then $\beta(B) = \beta(C)$. This is future work.

Algorithm: Given two HDA B, C, compute $\beta(B)$ and $\beta(C)$. If $\beta(B) \neq \beta(C)$, then B and C are not bisimilar.

So What?

So two HDA B, C are bisimilar if and only if there is a diagram

$$\begin{array}{ccc} & A & \\ {}^{f}\swarrow & & \searrow^{g} \\ B & & C \end{array}$$

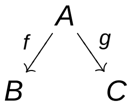
where $|f|$ and $|g|$ are dipath-lifting dimaps.

Enter Topology: Provide an algebraic invariant β such that if B and C are connected by a diagram like above, then $\beta(B) = \beta(C)$. This is future work.

Algorithm: Given two HDA B, C, compute $\beta(B)$ and $\beta(C)$. If $\beta(B) \neq \beta(C)$, then B and C are not bisimilar.

So What?

So two HDA B, C are bisimilar if and only if there is a diagram



where $|f|$ and $|g|$ are dipath-lifting dimaps.

Enter Topology: Provide an algebraic invariant β such that if B and C are connected by a diagram like above, then $\beta(B) = \beta(C)$. This is future work.

Algorithm: Given two HDA B, C, compute $\beta(B)$ and $\beta(C)$. If $\beta(B) \neq \beta(C)$, then B and C are not bisimilar.

Bisimulation up to Equivalence

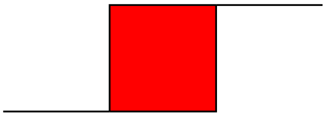
1 Introduction

2 Simulation and Bisimulation

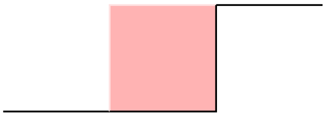
3 The Geometry of HDA

4 Bisimulation up to Equivalence
- Equivalence of Computations
- Bisimulation up to Equivalence

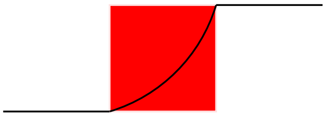
Equivalence of Computations



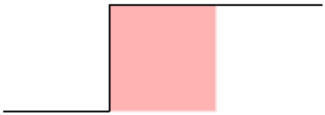
Equivalence of Computations



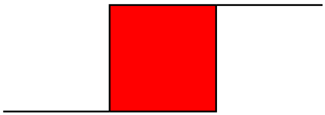
Equivalence of Computations



Equivalence of Computations



Equivalence of Computations



Two computations $(x_1, \dots, x_n)$, $(y_1, \dots, y_n)$ are adjacent if $x_i = y_i$ for all but one i.

Equivalence of computations is the equivalence relation generated by adjacency. [van Glabbeek 1991]

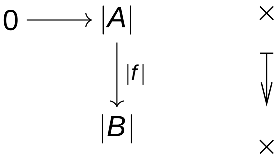
Bisimulation up to Equivalence

A morphism $f : A \to B$ is called an open map up to equivalence if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to a computation in B that is equivalent to the given one.

Bisimulation up to Equivalence

A morphism $f : A \to B$ is called an open map up to equivalence if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to a computation in B that is equivalent to the given one.

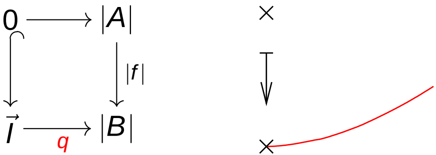
Conjecture: $f : A \to B$ is an open map up to equivalence if and only $|f| : |A| \to |B|$ lifts dipaths up to dihomotopy



Bisimulation up to Equivalence

A morphism $f : A \to B$ is called an open map up to equivalence if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to a computation in B that is equivalent to the given one.

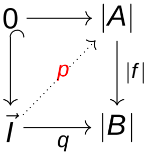
Conjecture: $f : A \to B$ is an open map up to equivalence if and only $|f| : |A| \to |B|$ lifts dipaths up to dihomotopy

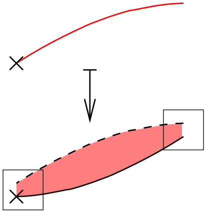


Bisimulation up to Equivalence

A morphism $f : A \to B$ is called an open map up to equivalence if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to a computation in B that is equivalent to the given one.

Conjecture: $f : A \to B$ is an open map up to equivalence if and only $|f| : |A| \to |B|$ lifts dipaths up to dihomotopy





Bisimulation up to Equivalence

A morphism $f : A \to B$ is called an open map up to equivalence if for any $a \in A$ and for any computation starting in $f(a)$, there is a computation starting in a which maps to a computation in B that is equivalent to the given one.

Conjecture: $f : A \to B$ is an open map up to equivalence if and only $|f| : |A| \to |B|$ lifts dipaths up to dihomotopy

Hypothesis (J. Srba): Bisimulation up to equivalence generalizes hereditary history-preserving bisimulation of asynchronous transition systems (and other formalisms).

Thank You!

Uli Fahrenberg
Ph.D. student, Aalborg University, Denmark

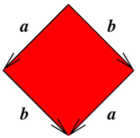
uli@math.aau.dk
http://www.math.aau.dk/∼uli

Selected Bibliography

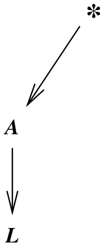
- V. Pratt, Modeling Concurrency with Geometry. Proc. 18th ACM Symposium on Principles of Programming Languages, 1991.
- R. van Glabbeek, Bisimulations for Higher Dimensional Automata. Email message, 1991.
- R. van Glabbeek, On the Expressiveness of Higher Dimensional Automata. Proc. EXPRESS 2004, to be published.
- E. Goubault, The Geometry of Concurrency. Ph.D. thesis, 1995.
- L. Fajstrup, E. Goubault, M. Raussen, Algebraic Topology and Concurrency. Theor.Comp.Sci., to be published.
- L. Fajstrup, Dipaths and Dihomotopies in a Cubical Complex. Adv.Appl.Math., to be published.

Labeled HDA

For labeling HDA, we work in a comma category of pointed precubical sets over a category of certain special alphabet precubical sets (which are ∞-tori). [Goubault 1995]



For idle transitions, we need to introduce *degeneracies*, i.e. to work with cubical sets instead of precubical. So the category of labeled HDA has diagrams like these:



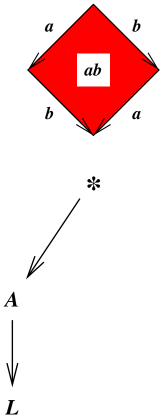
Black arrows: *precubical* morphisms
Red arrows: *cubical* morphisms

Labeled HDA

For labeling HDA, we work in a comma category of pointed precubical sets over a category of certain special alphabet precubical sets (which are ∞-tori). [Goubault 1995]

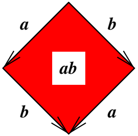
For idle transitions, we need to introduce *degeneracies*, i.e. to work with cubical sets instead of precubical. So the category of labeled HDA has diagrams like these:

Black arrows: *precubical* morphisms
Red arrows: *cubical* morphisms

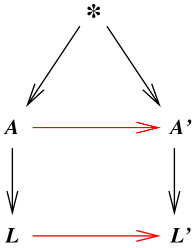


Labeled HDA

For labeling HDA, we work in a comma category of pointed precubical sets over a category of certain special alphabet precubical sets (which are ∞-tori). [Goubault 1995]



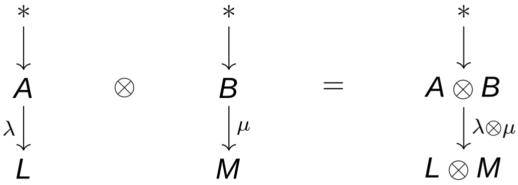
For idle transitions, we need to introduce *degeneracies*, i.e. to work with cubical sets instead of precubical. So the category of labeled HDA has diagrams like these:



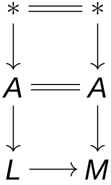
Black arrows: *precubical* morphisms
Red arrows: *cubical* morphisms

Compositions

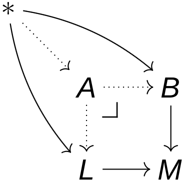
Product



Relabeling



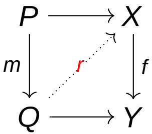
Restriction



Open Maps

Open maps are open in the sense of Joyal, Nielsen & Winskel with respect to the category CPath of acyclic rooted computation paths:

$f : A \to B$ is an open map iff, for any $m : P \to Q \in$ CPath, any diagram as below has a lift r:

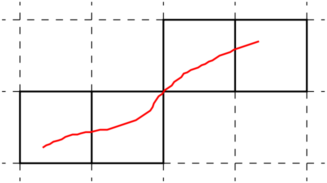


The Main Result

Theorem: $f : A \to B$ is an open map if and only if $|f| : |A| \to |B|$ has the dipath-lifting property

Key of proof: For any dipath, there is a unique computation through (the geometric realisation of) which it “runs.”

[Fajstrup 2003]



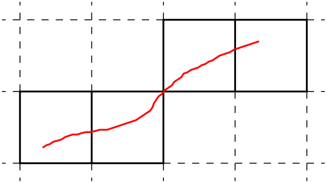
Only holds for locally finite HDA.

The Main Result

Theorem: $f : A \to B$ is an open map if and only if $|f| : |A| \to |B|$ has the dipath-lifting property

Key of proof: For any dipath, there is a unique computation through (the geometric realisation of) which it “runs.”

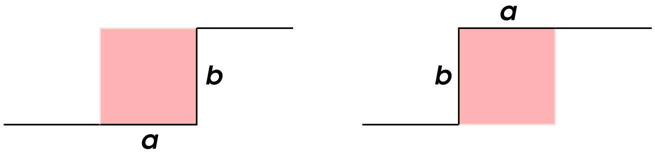
[Fajstrup 2003]



Only holds for locally finite HDA.

Bisimulation up to Equivalence

In this framework, simulations (and bisimulations) do not respect labels:

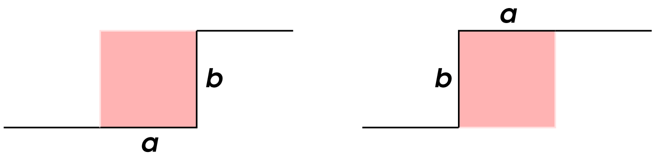


$\dots ab \dots$ $\dots ba \dots$

So we need to allow precubical morphisms that permute labels, to a certain extent.

Bisimulation up to Equivalence

In this framework, simulations (and bisimulations) do not respect labels:



$\ldots ab \ldots$ $\ldots ba \ldots$

So we need to allow precubical morphisms that permute labels, to a certain extent.