Algebraic topology and Concurrency

Traces spaces and applications

Martin Raussen

Department of Mathematical Sciences
Aalborg University - Denmark

Universität Bremen, Fachbereich Mathematik
November 25th, 2008

Outline

1. Directed algebraic topology
   - Motivations – mainly from Computer Science
   - Directed topology: Algebraic topology with a twist
2. Trace spaces and their organization
   - Trace spaces: definition, properties, applications
   - A categorical framework (with examples and applications)

Main Collaborators:

- Lisbeth Fajstrup (Aalborg), Éric Goubault, Emmanuel Haucourt (CEA, France)

Conference: Algebraic Topological Methods in Computer Science III, July 2008, Paris

Outline

1. Directed algebraic topology
   - Motivations – mainly from Computer Science
   - Directed topology: Algebraic topology with a twist
2. Trace spaces and their organization
   - Trace spaces: definition, properties, applications
   - A categorical framework (with examples and applications)

Main Collaborators:

- Lisbeth Fajstrup (Aalborg), Éric Goubault, Emmanuel Haucourt (CEA, France)

Conference: Algebraic Topological Methods in Computer Science III, July 2008, Paris

Outline

1. Directed algebraic topology
   - Motivations – mainly from Computer Science
   - Directed topology: Algebraic topology with a twist
2. Trace spaces and their organization
   - Trace spaces: definition, properties, applications
   - A categorical framework (with examples and applications)

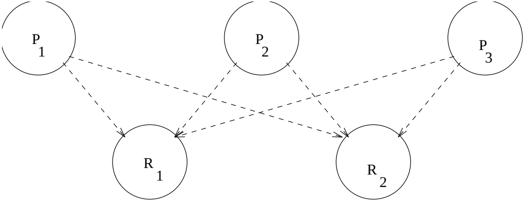
Main Collaborators:

- Lisbeth Fajstrup (Aalborg), Éric Goubault, Emmanuel Haucourt (CEA, France)

Conference: Algebraic Topological Methods in Computer Science III, July 2008, Paris

Motivation: Concurrency
Mutual exclusion

Mutual exclusion occurs, when n processes P_i compete for m resources R_j.



Only k processes can be served at any given time.

Semaphores!

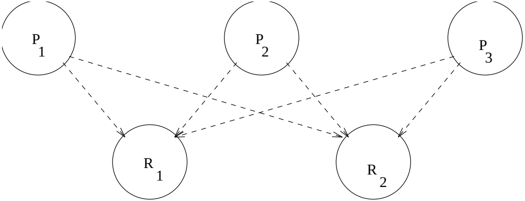
Semantics: A processor has to lock a resource and to relinquish the lock later on!

Description/abstraction $P_i : \dots PR_j \dots VR_j \dots$ (E.W. Dijkstra)

Motivation: Concurrency

Mutual exclusion

Mutual exclusion occurs, when n processes P_i compete for m resources R_j.



Only k processes can be served at any given time.

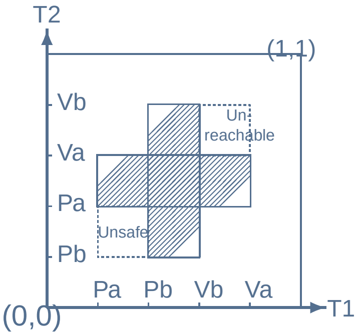
Semaphores!

Semantics: A processor has to lock a resource and to relinquish the lock later on!

Description/abstraction $P_i : \ldots PR_j \ldots VR_j \ldots$ (E.W. Dijkstra)

Schedules in "progress graphs"

The Swiss flag example



PV-diagram from

$P_1 : P_aP_bV_bV_a$

$P_2 : P_bP_aV_aV_b$

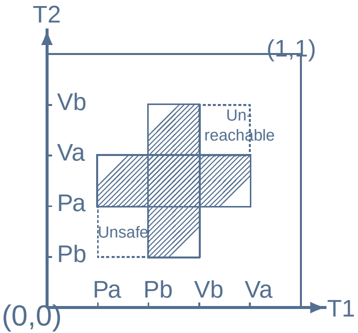
Executions are directed paths – since time flow is irreversible – avoiding a forbidden region (shaded).

Dipaths that are dihomotopic (through a 1-parameter deformation consisting of dipaths) correspond to equivalent executions.

Deadlocks, unsafe and unreachable regions may occur.

Schedules in "progress graphs"

The Swiss flag example



PV-diagram from

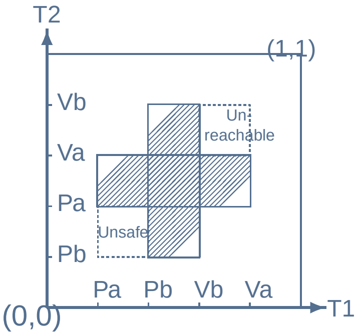
$P_1 : P_a P_b V_b V_a$

$P_2 : P_b P_a V_a V_b$

Executions are directed paths – since time flow is irreversible – avoiding a forbidden region (shaded). Dipaths that are dihomotopic (through a 1-parameter deformation consisting of dipaths) correspond to equivalent executions.

Deadlocks, unsafe and unreachable regions may occur.

Schedules in "progress graphs"
The Swiss flag example



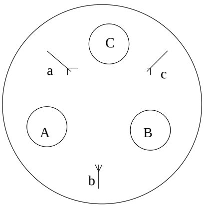
PV-diagram from
$P_1 : P_a P_b V_b V_a$
$P_2 : P_b P_a V_a V_b$

Executions are directed paths – since time flow is irreversible – avoiding a forbidden region (shaded). Dipaths that are dihomotopic (through a 1-parameter deformation consisting of dipaths) correspond to equivalent executions.

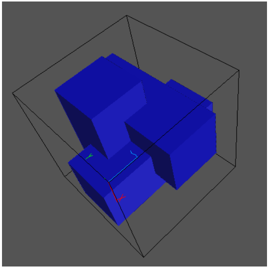
Deadlocks, unsafe and unreachable regions may occur.

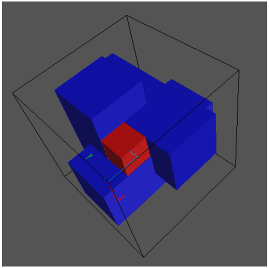
Higher dimensional automata (HDA) 1

Example: Dining philosophers; dimension 3 and beyond



A=Pa.Pb.Va.Vb
B=Pb.Pc.Vb.Vc
C=Pc.Pa.Vc.Va



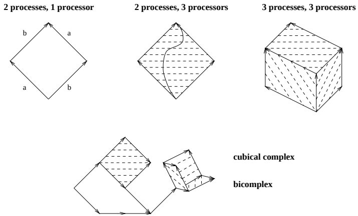


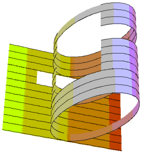
Higher dimensional complex with a forbidden region consisting of isothetic hypercubes and an unsafe region.

Higher dimensional automata (HDA) 2

seen as (geometric realizations of) pre-cubical sets

Vaughan Pratt, Rob van Glabbeek, Eric Goubault...





Squares/cubes/hypercubes are filled in iff actions on boundary are independent.

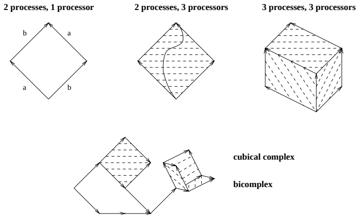
Higher dimensional automata are pre-cubical sets:

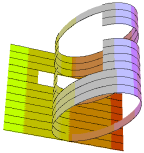
- like simplicial sets, but modelled on (hyper)cubes instead of simplices; glueing by face maps
- additionally: preferred directions – not all paths allowable.

Higher dimensional automata (HDA) 2

seen as (geometric realizations of) pre-cubical sets

Vaughan Pratt, Rob van Glabbeek, Eric Goubault...





Squares/cubes/hypercubes are filled in iff actions on boundary are independent.

Higher dimensional automata are pre-cubical sets:

- like simplicial sets, but modelled on (hyper)cubes instead of simplices; glueing by face maps
- additionally: preferred directions – not all paths allowable.

Transition systems

The “traditional” method to handle concurrency

Every single program corresponds to a (labelled) graph (with trivial idle loops at every vertex). Form the product of these graphs as a graph:

Vertices Sequences of vertices from the original graphs

Edges Sequences of (labelled) edges (including idle loops) from the original graphs

“Successive edges” are connected by a diamond – in general a hypercube. Need to handle independence relations algebraically.

Alternative: Petri nets – places, transitions, directed arcs, tokens...

Transition systems

The “traditional” method to handle concurrency

Every single program corresponds to a (labelled) graph (with trivial idle loops at every vertex). Form the product of these graphs as a graph:

Vertices Sequences of vertices from the original graphs

Edges Sequences of (labelled) edges (including idle loops) from the original graphs

“Successive edges” are connected by a diamond – in general a hypercube. Need to handle independence relations algebraically.

Alternative: Petri nets – places, transitions, directed arcs, tokens...

Transition systems

The “traditional” method to handle concurrency

Every single program corresponds to a (labelled) graph (with trivial idle loops at every vertex). Form the product of these graphs as a graph:

Vertices Sequences of vertices from the original graphs

Edges Sequences of (labelled) edges (including idle loops) from the original graphs

“Successive edges” are connected by a diamond – in general a hypercube. Need to handle independence relations algebraically.

Alternative: Petri nets – places, transitions, directed arcs, tokens...

Transition systems

The “traditional” method to handle concurrency

Every single program corresponds to a (labelled) graph (with trivial idle loops at every vertex). Form the product of these graphs as a graph:

Vertices Sequences of vertices from the original graphs

Edges Sequences of (labelled) edges (including idle loops) from the original graphs

“Successive edges” are connected by a diamond – in general a hypercube. Need to handle independence relations algebraically.

Alternative: Petri nets – places, transitions, directed arcs, tokens...

Discrete versus continuous models

How to handle the state-space explosion problem?

Discrete models for concurrency (transition graph models) suffer a severe problem if the number of processors and/or the length of programs grows: The number of states (and the number of possible schedules) grows exponentially:
This is known as the state space explosion problem.
You need clever ways to find out which of the schedules yield equivalent results – e.g., to check for correctness – for general reasons. Then check only one per equivalence class.
Alternative: Infinite continuous models allowing for well-known equivalence relations on paths (homotopy = 1-parameter deformations) – but with an important twist!
Analogy: Continuous physics as an approximation to (discrete) quantum physics.

Discrete versus continuous models

How to handle the state-space explosion problem?

Discrete models for concurrency (transition graph models) suffer a severe problem if the number of processors and/or the length of programs grows: The number of states (and the number of possible schedules) grows exponentially:

This is known as the state space explosion problem.

You need clever ways to find out which of the schedules yield equivalent results – e.g., to check for correctness – for general reasons. Then check only one per equivalence class.

Alternative: Infinite continuous models allowing for well-known equivalence relations on paths (homotopy = 1-parameter deformations) – but with an important twist!

Analogy: Continuous physics as an approximation to (discrete) quantum physics.

Discrete versus continuous models

How to handle the state-space explosion problem?

Discrete models for concurrency (transition graph models) suffer a severe problem if the number of processors and/or the length of programs grows: The number of states (and the number of possible schedules) grows exponentially:

This is known as the state space explosion problem.

You need clever ways to find out which of the schedules yield equivalent results – e.g., to check for correctness – for general reasons. Then check only one per equivalence class.

Alternative: Infinite continuous models allowing for well-known equivalence relations on paths (homotopy = 1-parameter deformations) – but with an important twist!

Analogy: Continuous physics as an approximation to (discrete) quantum physics.

Concepts from algebraic topology 1

Homotopy

Top: the category of topological spaces and continuous maps.
$I = [0, 1]$ the unit interval.

Definition

- A continuous map $H : X \times I \to Y$ (one-parameter deformation) is called a homotopy.
- Continuous maps $f, g : X \to Y$ are called homotopic to each other ($f \simeq g$) if there is a homotopy H with $H(x, 0) = f(x), H(x, 1) = g(x), x \in X$.
- $[X, Y]$ the set of homotopy classes of continuous maps from X to Y.
- A continuous map $f : X \to Y$ is called a homotopy equivalence if it has a "homotopy inverse" $g : Y \to X$ such that $g \circ f \simeq id_X, f \circ g \simeq id_Y$.

Concepts from algebraic topology 2

The fundamental group. Higher homotopy groups

Definition

- Variation: pointed continuous maps $f : (X, *) \to (Y, *)$ and pointed homotopies $H : (X \times I, * \times I) \to (Y, *)$.
- Loops in Y as the special case $X = S^1$ (unit circle).
- Fundamental group $\pi_1(Y, y) = [(S^1, *), (Y, y)]$ with product arising from concatenation and inverse from reversal.
- Higher homotopy groups $\pi_i(Y, y) = [(S^i, *), (Y, y)]$ – product from the pinch coproduct on S^k; abelian for $k > 1$.
- A continuous map $f : X \to Y$ induces group homomorphisms $f_\sharp : \pi_i(X, x) \to \pi_i(Y, f(x))$ (isomorphisms for a homotopy equivalence).
- The map f is called k-connected if it induces isomorphisms on π_i, $i \leq k$ and a surjection on π_{k+1}.

Concepts from algebraic topology 2

The fundamental group. Higher homotopy groups

Definition

- Variation: pointed continuous maps $f : (X,*) \to (Y,*)$ and pointed homotopies $H : (X \times I, * \times I) \to (Y,*)$.
- Loops in Y as the special case $X = S^1$ (unit circle).
- Fundamental group $\pi_1(Y,y)$= $[(S^1,*),(Y,y)]$ with product arising from concatenation and inverse from reversal.
- Higher homotopy groups $\pi_i(Y,y)$= $[(S^i,*),(Y,y)]$ – product from the pinch coproduct on S^k; abelian for $k > 1$.
- A continuous map $f : X \to Y$ induces group homomorphisms $f_\sharp : \pi_i(X,x) \to \pi_i(Y,f(x))$ (isomorphisms for a homotopy equivalence).
- The map f is called k-connected if it induces isomorphisms on π_i, $i \leq k$ and a surjection on π_{k+1}.

A framework for directed topology

Directed paths as part of the structure: d-spaces, M. Grandis (03)

X a topological space. $\vec{P}(X) \subseteq X^I = \{p : I = [0,1] \to X \text{ cont.}\}$ a set of d-paths ("directed" paths $\leftrightarrow$ executions) satisfying

- $\{$ constant paths $\} \subseteq \vec{P}(X)$
- $\varphi \in \vec{P}(X)(x,y), \psi \in \vec{P}(X)(y,z) \Rightarrow \varphi * \psi \in \vec{P}(X)(x,z)$
- $\varphi \in \vec{P}(X), \alpha \in I^I$ a nondecreasing reparametrization $\Rightarrow \varphi \circ \alpha \in \vec{P}(X)$

The pair $(X, \vec{P}(X))$ is called a d-space.

Observe: $\vec{P}(X)$ is in general not closed under reversal:

$$\alpha(t) = 1 - t, \varphi \in \vec{P}(X) \not\Rightarrow \varphi \circ \alpha \in \vec{P}(X)!$$

Examples:

- An HDA with directed execution paths.
- A space-time(relativity) with time-like or causal curves.

A framework for directed topology

Directed paths as part of the structure: d-spaces, M. Grandis (03)

X a topological space. $\vec{P}(X) \subseteq X^I = \{p : I = [0,1] \to X \text{ cont.}\}$ a set of d-paths ("directed" paths $\leftrightarrow$ executions) satisfying

- $\{$ constant paths $\} \subseteq \vec{P}(X)$
- $\varphi \in \vec{P}(X)(x,y), \psi \in \vec{P}(X)(y,z) \Rightarrow \varphi * \psi \in \vec{P}(X)(x,z)$
- $\varphi \in \vec{P}(X), \alpha \in I^I$ a nondecreasing reparametrization $\Rightarrow \varphi \circ \alpha \in \vec{P}(X)$

The pair $(X, \vec{P}(X))$ is called a d-space.
Observe: $\vec{P}(X)$ is in general not closed under reversal:

$$\alpha(t) = 1 - t,\ \varphi \in \vec{P}(X) \not\Rightarrow \varphi \circ \alpha \in \vec{P}(X)!$$

Examples:

- An HDA with directed execution paths.
- A space-time(relativity) with time-like or causal curves.

A framework for directed topology

Directed paths as part of the structure: d-spaces, M. Grandis (03)

X a topological space. $\vec{P}(X) \subseteq X^I = \{p : I = [0,1] \to X \text{ cont.}\}$ a set of d-paths ("directed" paths $\leftrightarrow$ executions) satisfying

- $\{$ constant paths $\} \subseteq \vec{P}(X)$
- $\varphi \in \vec{P}(X)(x,y), \psi \in \vec{P}(X)(y,z) \Rightarrow \varphi * \psi \in \vec{P}(X)(x,z)$
- $\varphi \in \vec{P}(X), \alpha \in I^I$ a nondecreasing reparametrization $\Rightarrow \varphi \circ \alpha \in \vec{P}(X)$

The pair $(X, \vec{P}(X))$ is called a d-space.

Observe: $\vec{P}(X)$ is in general not closed under reversal:

$$\alpha(t) = 1 - t, \varphi \in \vec{P}(X) \not\Rightarrow \varphi \circ \alpha \in \vec{P}(X)!$$

Examples:

- An HDA with directed execution paths.
- A space-time(relativity) with time-like or causal curves.

d-maps, dihomotopy

A **d-map** $f : X \to Y$ is a continuous map satisfying

- $f(\vec{P}(X)) \subseteq \vec{P}(Y)$.

Let $\vec{P}(I) = \{\sigma \in I^I | \sigma \text{ nondecreasing reparametrization}\}$, and $\vec{I} = (I, \vec{P}(I))$. Then

- $\vec{P}(X) =$ set of d-maps from $\vec{I}$ to X.

A **dihomotopy** $H : X \times I \to Y$ is a continuous map such that

- every H_t a d-map

i.e., a 1-parameter deformation of d-maps.

d-maps, dihomotopy

A d-map $f : X \to Y$ is a continuous map satisfying

- $f(\vec{P}(X)) \subseteq \vec{P}(Y)$.

Let $\vec{P}(I) = \{\sigma \in I^I | \sigma \text{ nondecreasing reparametrization}\}$, and $\vec{I} = (I, \vec{P}(I))$. Then

- $\vec{P}(X) =$ set of d-maps from $\vec{I}$ to X.

A dihomotopy $H : X \times I \to Y$ is a continuous map such that

- every H_t a d-map

i.e., a 1-parameter deformation of d-maps.

d-maps, dihomotopy

A **d-map** $f : X \to Y$ is a continuous map satisfying

- $f(\vec{P}(X)) \subseteq \vec{P}(Y)$.

Let $\vec{P}(I) = \{\sigma \in I^I | \sigma \text{ nondecreasing reparametrization}\}$, and $\vec{I} = (I, \vec{P}(I))$. Then

- $\vec{P}(X) =$ set of d-maps from $\vec{I}$ to X.

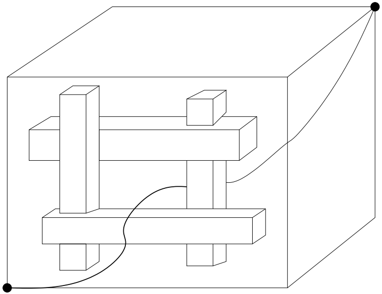
A **dihomotopy** $H : X \times I \to Y$ is a continuous map such that

- every H_t a d-map

i.e., a 1-parameter deformation of d-maps.

Dihomotopy is finer than homotopy with fixed endpoints

Example: Two L-shaped wedges as the forbidden region

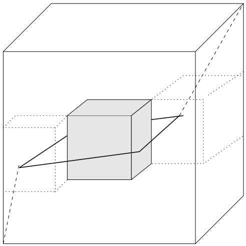


All dipaths from minimum to maximum are homotopic.
A dipath through the "hole" is not dihomotopic to a dipath on the boundary.

The twist has a price

Neither homogeneity nor cancellation nor group structure

Ordinary topology:
Path space = loop space (within each path component).
A loop space is an H-space with concatenation, inversion, cancellation.



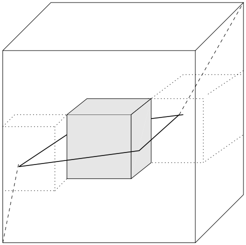
"Birth and death" of d-homotopy classes

Directed topology:
Loops do not tell much; concatenation ok, cancellation not!
Replace group structure by category structures!

The twist has a price

Neither homogeneity nor cancellation nor group structure

Ordinary topology:
Path space = loop space (within each path component).
A loop space is an H-space with concatenation, inversion, cancellation.



"Birth and death" of
d-homotopy classes

Directed topology:
Loops do not tell much; concatenation ok, cancellation not!
Replace group structure by category structures!

D-paths, traces and trace categories

Getting rid of reparametrizations – like in differential geometry of curves

X a (saturated) d-space.
$\varphi, \psi \in \vec{P}(X)(x, y)$ are called reparametrization equivalent if there are $\alpha, \beta \in \vec{P}(I)$ such that $\varphi \circ \alpha = \psi \circ \beta$ ("same oriented trace").

Theorem

(Fahrenberg-R., 07): Reparametrization equivalence is an equivalence relation (transitivity!).

$\vec{T}(X)(x, y) = \vec{P}(X)(x, y)/_\simeq$ makes $\vec{T}(X)$ into the (topologically enriched) trace category – composition associative.
A d-map $f : X \to Y$ induces a functor $\vec{T}(f) : \vec{T}(X) \to \vec{T}(Y)$.

D-paths, traces and trace categories

Getting rid of reparametrizations – like in differential geometry of curves

X a (saturated) d-space.
$\varphi, \psi \in \vec{P}(X)(x,y)$ are called reparametrization equivalent if there are $\alpha, \beta \in \vec{P}(I)$ such that $\varphi \circ \alpha = \psi \circ \beta$ ("same oriented trace").

Theorem

(Fahrenberg-R., 07): Reparametrization equivalence is an equivalence relation (transitivity!).

$\vec{T}(X)(x,y) = \vec{P}(X)(x,y)/_{\simeq}$ makes $\vec{T}(X)$ into the (topologically enriched) trace category – composition associative.
A d-map $f : X \to Y$ induces a functor $\vec{T}(f) : \vec{T}(X) \to \vec{T}(Y)$.

Two main objectives

- Investigation/calculation of the homotopy type of trace spaces $\vec{T}(X)(x,y)$ for relevant d-spaces X
- Investigation of topology change under variation of end points:

$$\vec{T}(X)(x',y) \overset{\sigma^*_{x'x}}{\longleftarrow} \vec{T}(X)(x,y) \xrightarrow{\sigma_{yy'*}} \vec{T}(X)(x,y')$$

Categorical organization, leading to components of end points – without topology change

Application: Enough to check one d-path among all paths through the same components!

Two main objectives

- Investigation/calculation of the homotopy type of trace spaces $\vec{T}(X)(x,y)$ for relevant d-spaces X
- Investigation of topology change under variation of end points:

$$\vec{T}(X)(x',y) \overset{\sigma^*_{x'x}}{\longleftarrow} \vec{T}(X)(x,y) \overset{\sigma_{yy'*}}{\longrightarrow} \vec{T}(X)(x,y')$$

Categorical organization, leading to components of end points – without topology change

Application: Enough to check one d-path among all paths through the same components!

Two main objectives

- Investigation/calculation of the homotopy type of trace spaces $\vec{T}(X)(x,y)$ for relevant d-spaces X
- Investigation of topology change under variation of end points:

$$\vec{T}(X)(x',y) \overset{\sigma^*_{x'x}}{\longleftarrow} \vec{T}(X)(x,y) \xrightarrow{\sigma_{yy'*}} \vec{T}(X)(x,y')$$

Categorical organization, leading to components of end points – without topology change

Application: Enough to check one d-path among all paths through the same components!

Aim: Decomposition of trace spaces

Method: Investigation of concatenation maps

Let $L \subset X$ denote a (properly chosen) subspace "between" $X_0, X_1 \subset X$.
Investigate the concatenation map
$c_L : \vec{T}(X_0)(x_0, L) \times_L \vec{T}(X_1)(L, x_1) \to \vec{T}(X)(x_0, x_1),\ (p_0, p_1) \mapsto p_0 * p_1$

onto? fibres? Topology of the pieces?

Generalization: L_{ij} a sequence of (properly chosen) subspaces between subdivision $X_1, \cdots, X_n$. Investigate the concatenation map on
$\vec{T}(X_0)(x_0, L_{01}) \times_{L_{12}} \cdots \times_{L_{j-1,j}} \vec{T}(X_j)(L_{j-1,j}, L_{j,j+1}) \times_{L_{j,j+1}} \cdots \times_{L_{n-1,n}} \vec{T}(X_n)(L_{n-1,n}, x_1)$.

onto? fibres? Topology of the pieces?

Aim: Decomposition of trace spaces

Method: Investigation of concatenation maps

Let $L \subset X$ denote a (properly chosen) subspace "between" $X_0, X_1 \subset X$.
Investigate the concatenation map
$c_L : \vec{T}(X_0)(x_0, L) \times_L \vec{T}(X_1)(L, x_1) \to \vec{T}(X)(x_0, x_1),\ (p_0, p_1) \mapsto p_0 * p_1$

onto? fibres? Topology of the pieces?

Generalization: L_{ij} a sequence of (properly chosen) subspaces between subdivision $X_1, \cdots, X_n$. Investigate the concatenation map on
$\vec{T}(X_0)(x_0, L_{01}) \times_{L_{12}} \cdots \times_{L_{j-1,j}} \vec{T}(X_j)(L_{j-1,j}, L_{j,j+1}) \times_{L_{j,j+1}} \cdots \times_{L_{n-1,n}} \vec{T}(X_n)(L_{n-1,n}, x_1)$.

onto? fibres? Topology of the pieces?

Tool: The Vietoris-Begle mapping theorem

Stephen Smale's version for homotopy groups

What does a surjective map $p : X \to Y$ with highly connected fibres $p^{-1}(y), y \in Y$, tell about invariants of X, Y?

The Vietoris-Begle mapping theorem compares the Alexander-Spanier cohomology groups of X, Y.

Stephen Smale, *A Vietoris Mapping Theorem for Homotopy*, Proc. Amer. Math. Soc. **8** (1957), no. 3, 604 – 610.

Theorem

Let $f : X \to Y$ denote a proper surjective map between connected locally compact separable metric spaces. Let moreover X be locally n-connected, and for each $y \in Y$, let $f^{-1}(y)$ be locally $(n-1)$-connected and $(n-1)$-connected.

1. *Y is locally n-connected, and*
2. *$f_{\#} : \pi_r(X) \to \pi_r(Y)$ is an isomorphism for all $0 \leq r \leq n-1$ and onto for $r = n$.*

Tool: The Vietoris-Begle mapping theorem

Stephen Smale's version for homotopy groups

What does a surjective map $p : X \to Y$ with highly connected fibres $p^{-1}(y), y \in Y$, tell about invariants of X, Y?

The Vietoris-Begle mapping theorem compares the Alexander-Spanier cohomology groups of X, Y.

Stephen Smale, *A Vietoris Mapping Theorem for Homotopy*, Proc. Amer. Math. Soc. **8** (1957), no. 3, 604 – 610:

Theorem

Let $f : X \to Y$ denote a proper surjective map between connected locally compact separable metric spaces. Let moreover X be locally n-connected, and for each $y \in Y$, let $f^{-1}(y)$ be locally $(n-1)$-connected and $(n-1)$-connected.

1. *Y is locally n-connected, and*
2. *$f_\# : \pi_r(X) \to \pi_r(Y)$ is an isomorphism for all $0 \le r \le n-1$ and onto for $r = n$.*

An important special case

All fibres contractible and locally contractible

Corollary

Let $f : X \to Y$ denote a proper surjective map between locally compact separable metric spaces. Let moreover X be locally contractible, and for each $y \in Y$, let $f^{-1}(y)$ be contractible and locally contractible. Then

1. *Y is locally contractible, and*
2. *f is a weak homotopy equivalence, i.e., $f_\sharp : \pi_i(X) \to \pi_i(Y)$ is an isomorphism for all i.*

Trace spaces in a pre-cubical complex X

Arc length and arc length parametrization

A pre-cubical complex X is glued together out of a set of hypercubes $\square^n$ along their boundaries – similar to pre-simplicial sets/complexes. Every hypercube defines d-paths $\vec{P}(\square^n)$. Concatenations of these gives rise to $\vec{P}(X)$.

l^1 "arc length" parametrization for d-paths in a pre-cubical complex – glued out of hypercubes of various dimensions: On each hypercube, arc length is the l^1-distance of end-points. Additive continuation $\rightsquigarrow$

Subspace of arc-length parametrized d-paths $\vec{P}_n(X) \subset \vec{P}(X)$.

Dihomotopic paths in $\vec{P}_n(X)(x, y)$ have the same arc length!

Trace spaces in a pre-cubical complex X

Arc length and arc length parametrization

A pre-cubical complex X is glued together out of a set of hypercubes $\Box^n$ along their boundaries – similar to pre-simplicial sets/complexes. Every hypercube defines d-paths $\vec{P}(\Box^n)$. Concatenations of these gives rise to $\vec{P}(X)$.

l^1 "arc length" parametrization for d-paths in a pre-cubical complex – glued out of hypercubes of various dimensions: On each hypercube, arc length is the l^1-distance of end-points. Additive continuation $\rightsquigarrow$

Subspace of arc-length parametrized d-paths $\vec{P}_n(X) \subset \vec{P}(X)$.

Dihomotopic paths in $\vec{P}_n(X)(x,y)$ have the same arc length!

Topology of trace spaces in a pre-cubical complex X

MR, Trace spaces in a pre-cubical complex, Aalborg preprint

The spaces $\vec{P}_n(X)$ and $\vec{T}(X)$ are homeomorphic, $\vec{P}(X)$ is homotopy equivalent to both.

Theorem

X a pre-cubical set; $x, y \in X$. Then $\vec{T}(X)(x, y)$

- *is metrizable, locally contractible and locally compact.*
- *has the homotopy type of a CW-complex – (using Milnor 1959).*

$\rightsquigarrow$ Vietoris theorem applicable!

Topology of trace spaces in a pre-cubical complex X

MR, Trace spaces in a pre-cubical complex, Aalborg preprint

The spaces $\vec{P}_n(X)$ and $\vec{T}(X)$ are homeomorphic,
$\vec{P}(X)$ is homotopy equivalent to both.

Theorem

X a pre-cubical set; $x, y \in X$. Then $\vec{T}(X)(x, y)$

- *is metrizable, locally contractible and locally compact.*
- *has the homotopy type of a CW-complex – (using Milnor 1959).*

$\rightsquigarrow$ Vietoris theorem applicable!

Applications to trace spaces I

A simple case as illustration

Definition

A subset $L \subseteq X$ of a d-space X is called

achronal if all $p \in \vec{P}(L) \subset \vec{P}(X)$ are constant.

order convex if $p^{-1}(L)$ is either an interval or empty for all $p \in \vec{P}(X)$;

unavoidable from $B \subset X$ to $C \subset X$ if $\vec{P}(X \setminus L)(B, C) = \emptyset$.

Theorem

Let X denote a d-space, $x_0, x_1 \in X$ and $L \subset X$ a subspace that is achronal, order convex and unavoidable from x_0 to x_1. Then the concatenation map

$$c_L : \vec{T}(X)(x_0, L) \times_L \vec{T}(X)(L, x_1) \to \vec{T}(X)(x_0, x_1),\ (p_0, p_1) \mapsto p_0 * p_1$$

is a homeomorphism.

Applications to trace spaces I

A simple case as illustration

Definition

A subset $L \subseteq X$ of a d-space X is called

achronal if all $p \in \vec{P}(L) \subset \vec{P}(X)$ are constant.

order convex if $p^{-1}(L)$ is either an interval or empty for all $p \in \vec{P}(X)$;

unavoidable from $B \subset X$ to $C \subset X$ if $\vec{P}(X \setminus L)(B, C) = \emptyset$.

Theorem

*Let X denote a d-space, $x_0, x_1 \in X$ and $L \subset X$ a subspace that is **achronal, order convex** and **unavoidable** from x_0 to x_1. Then the **concatenation map***

$$c_L : \vec{T}(X)(x_0, L) \times_L \vec{T}(X)(L, x_1) \to \vec{T}(X)(x_0, x_1),\ (p_0, p_1) \mapsto p_0 * p_1$$

*is a **homeomorphism**.*

Applications to trace spaces I

Traces in pre-cubical complexes

Theorem

Let X be (the geometric realization of) a pre-cubical complex. Let $x_0, x_1 \in X, L \subset X$ a subcomplex that is order convex and unavoidable from x_0 to x_1. Then[a] the concatenation map
$c_L : \vec{T}(X)(x_0, L) \times_L \vec{T}(X)(L, x_1) \to \vec{T}(X)(x_0, x_1),\ (p_0, p_1) \mapsto p_0 * p_1$
is a homotopy equivalence.

[a] add an extra technical condition

An important special case

Corollary

If, moreover, $\vec{T}(X_0)(x_0,l)$ and $\vec{T}(X_1)(l,x_1)$ are contractible and locally contractible for every $l \in L \cap [x_0,x_1]$, then $\vec{T}(X)(x_0,x_1)$ is homotopy equivalent to $L \cap [x_0,x_1]$.

Remark: Since pieces are trivial, the topology resides in the glueing. “Huge” trace space identified with “small” space $L \cap [x_0,x_1] \subset X$

Proof.

The fibre over $l \in L$ of the “mid point” map $m : \vec{T}(X)(x_0,L) \times_L \vec{T}(X)(L,x_1) \to L \cap [x_0,x_1]$ is $m^{-1}(l) = \vec{T}(X)(x_0,l) \times \vec{T}(X)(l,x_1)$ – contractible. □

An important special case

Corollary

If, moreover, $\vec{T}(X_0)(x_0, l)$ and $\vec{T}(X_1)(l, x_1)$ are contractible and locally contractible for every $l \in L \cap [x_0, x_1]$, then $\vec{T}(X)(x_0, x_1)$ is homotopy equivalent to $L \cap [x_0, x_1]$.

Remark: Since pieces are trivial, the topology resides in the glueing. "Huge" trace space identified with "small" space $L \cap [x_0, x_1] \subset X$

Proof.

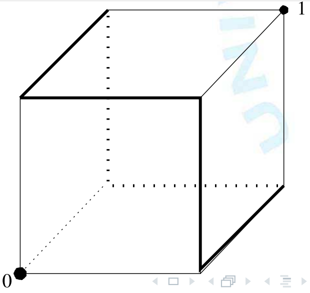
The fibre over $l \in L$ of the "mid point" map $m : \vec{T}(X)(x_0, L) \times_L \vec{T}(X)(L, x_1) \to L \cap [x_0, x_1]$ is $m^{-1}(l) = \vec{T}(X)(x_0, l) \times \vec{T}(X)(l, x_1)$ – contractible. □

First examples

Example

I^n the unit cube with boundary ∂I^n.
$X = \partial I^n = \{\mathbf{x} \in I^n \mid \exists i : x_i = 0 \vee x_i = 1\} \simeq S^{n-1}$
$L = \partial_{\pm} I^n = \{\mathbf{x} \in I^n \mid \exists i, j : x_i = 0, x_j = 1\} \simeq S^{n-2}$.

- $\vec{T}(I^n; \mathbf{x}, \mathbf{y})$ is contractible for all $x \preceq y \in I^n$;
- $\vec{T}(\partial I^n; \mathbf{0}, \mathbf{1})$ is homotopy equivalent to S^{n-2}.



Proof.

Choose $L = \partial_{\pm} I^n$. $\square$

Key points in the proof of Theorem

- Topological conditions ok.
- Check that path components are mapped into each other by bijection.
- Surjectivity of c_L corresponds to unavoidability.
- Order convexity ensures that every fibre $c_L^{-1}(p)$ is an interval, hence contractible.
- The weak homotopy equivalence is a homotopy equivalence since domain and codomain of c_L have the homotopy type of a CW-complex.

Applications to trace spaces II: A generalisation

Definition

pieces and separating layers: $x_0, x_1 \in X$. $[x_0, x_1] = \bigcup_{i \in J} X_i$; $L_{ij} \subseteq X_i \cap X_j$ such that

L_{**} order convex $p \in \vec{P}(X_i) \Rightarrow$

$$\begin{cases} p^{-1}(L_{ji}) = [0, a] & \text{for some} \quad a < 1 \quad (\emptyset \text{ if } a < 0) \\ p^{-1}(L_{ij}) = [b, 1] & \text{for some} \quad 0 < b \quad (\emptyset \text{ if } b > 1) \end{cases}$$

L_{**} unavoidable: $\vec{P}(X_i \cup X_j \setminus L_{ij})(X_i \setminus L_{ij}, X_j \setminus L_{ij}) = \emptyset$.

$L_{ij} \leq L_{jk}$ if $\vec{P}(X_j)(L_{ij}, L_{jk}) \neq \emptyset$.

$S = (L_{i_k, i_{k+1}})_{0 \leq k \leq n}$ admissible from x_0 to x_1 if

$$x_0 \in X_{i_0}, x_1 \in X_{i_n}, L_{i_k, i_{k+1}} \leq L_{i_{k+1}, i_{k+2}}.$$

$$T_S(X)(x_0, x_1) = \vec{T}(X_{i_0})(x_0, L_{i_0, i_1}) \times_{L_{i_0, i_1}} \cdots \times_{L_{i_{k-1}, i_k}}$$
$$\vec{T}(X_{i_k})(L_{i_{k-1}, i_k}, L_{i_k, i_{k+1}}) \times_{L_{i_k, i_{k+1}}} \cdots \times_{L_{i_{n-1}, i_n}} \vec{T}(X_{i_n})(L_{i_{n-1}, i_n}, x_1)$$

Applications to trace spaces II: A generalisation

Definition

pieces and separating **layers**: $x_0, x_1 \in X$. $[x_0, x_1] = \bigcup_{i \in J} X_i$;

$L_{ij} \subseteq X_i \cap X_j$ such that

L_{**} **order convex** $p \in \vec{P}(X_i) \Rightarrow$

$$\begin{cases} p^{-1}(L_{ji}) = [0, a] & \text{for some} \quad a < 1 \quad (\emptyset \text{ if } a < 0) \\ p^{-1}(L_{ij}) = [b, 1] & \text{for some} \quad 0 < b \quad (\emptyset \text{ if } b > 1) \end{cases}$$

L_{**} **unavoidable**: $\vec{P}(X_i \cup X_j \setminus L_{ij})(X_i \setminus L_{ij}, X_j \setminus L_{ij}) = \emptyset$.

$L_{ij} \leq L_{jk}$ if $\vec{P}(X_j)(L_{ij}, L_{jk}) \neq \emptyset$.

$S = (L_{i_k, i_{k+1}})_{0 \leq k \leq n}$ **admissible** from x_0 to x_1 if

$x_0 \in X_{i_0}, x_1 \in X_{i_n}, L_{i_k, i_{k+1}} \leq L_{i_{k+1}, i_{k+2}}$.

$T_S(X)(x_0, x_1) = \vec{T}(X_{i_0})(x_0, L_{i_0, i_1}) \times_{L_{i_0, i_1}} \cdots \times_{L_{i_{k-1}, i_k}}$
$\vec{T}(X_{i_k})(L_{i_{k-1}, i_k}, L_{i_k, i_{k+1}}) \times_{L_{i_k, i_{k+1}}} \cdots \times_{L_{i_{n-1}, i_n}} \vec{T}(X_{i_n})(L_{i_{n-1}, i_n}, x_1)$

Applications to trace spaces II: A generalisation

Definition

pieces and separating **layers**: $x_0, x_1 \in X$. $[x_0, x_1] = \bigcup_{i \in J} X_i$;
$L_{ij} \subseteq X_i \cap X_j$ such that

L_{**} **order convex** $p \in \vec{P}(X_i) \Rightarrow$

$$\begin{cases} p^{-1}(L_{ji}) = [0, a] & \text{for some} \quad a < 1 \quad (\emptyset \text{ if } a < 0) \\ p^{-1}(L_{ij}) = [b, 1] & \text{for some} \quad 0 < b \quad (\emptyset \text{ if } b > 1) \end{cases}$$

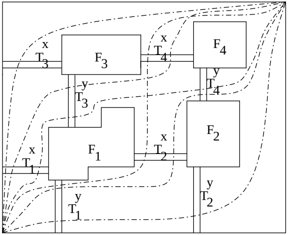
L_{**} **unavoidable**: $\vec{P}(X_i \cup X_j \setminus L_{ij})(X_i \setminus L_{ij}, X_j \setminus L_{ij}) = \emptyset$.

$L_{ij} \leq L_{jk}$ if $\vec{P}(X_j)(L_{ij}, L_{jk}) \neq \emptyset$.

$S = (L_{i_k, i_{k+1}})_{0 \leq k \leq n}$ **admissible** from x_0 to x_1 if
$x_0 \in X_{i_0}, x_1 \in X_{i_n}, L_{i_k, i_{k+1}} \leq L_{i_{k+1}, i_{k+2}}$.

$$T_S(X)(x_0, x_1) = \vec{T}(X_{i_0})(x_0, L_{i_0, i_1}) \times_{L_{i_0, i_1}} \cdots \times_{L_{i_{k-1}, i_k}} \vec{T}(X_{i_k})(L_{i_{k-1}, i_k}, L_{i_k, i_{k+1}}) \times_{L_{i_k, i_{k+1}}} \cdots \times_{L_{i_{n-1}, i_n}} \vec{T}(X_{i_n})(L_{i_{n-1}, i_n}, x_1)$$

Decomposition of d-path spaces



Theorem

The concatenation map $c : \bigcup_S \vec{T}_S(X)(x_o, x_1) \to \vec{T}(X)(x_o, x_1)$ is

1. *a homeomorphism if L_{ij} achronal.*
2. *a homotopy equivalence if L_{**} order convex collection of subcomplexes of a pre-cubical complex X.*

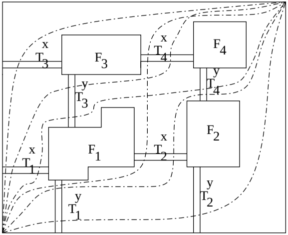
Proof.

Case (2): Apply Smale's Vietoris theorem.

Surjectivity: Every d-path can be decomposed along an admissible sequence (unavoidability)

Fibres are product of intervals, contractible!

Decomposition of d-path spaces



Theorem

The concatenation map
$c : \bigcup_S \vec{T}_S(X)(x_o, x_1) \to \vec{T}(X)(x_o, x_1)$ *is*

1. *a homeomorphism if L_{ij} achronal.*
2. *a homotopy equivalence if L_{**} order convex collection of subcomplexes of a pre-cubical complex X.*

Proof.

Case (2): Apply Smale's Vietoris theorem.

Surjectivity: Every d-path can be decomposed along an admissible sequence (unavoidability)

Fibres are product of intervals, contractible!

An important special case II

Reachability. For a given collection of pieces and layers $\mathcal{L} = \bigcup L_{ij}$ in X that is unavoidable from x_0 to x_1, let $R^{\mathcal{L}}(X)(x_0, x_1) = \{(x_{i_0 j_0}, \dots, x_{i_n j_n}) \in L_{i_0 j_0} \times \cdots \times L_{i_n j_n} \mid \vec{P}(X_{i_k})(x_{i_k j_k}, x_{i_{k+1} j_{k+1}}) \neq \emptyset, n \geq 0\}$ denote the space of mutually reachable points in the given layers.

Corollary

If, moreover, all path spaces $\vec{T}(X_{i_k})(x_{i_{k-1}}, x_{i_k}), x_{i_{k-1}} \in L_{i_{k-1} i_k}, x_{i_k} \in L_{i_k i_{k+1}}$ are contractible and locally contractible (resp. highly connected), then $\vec{T}(X)(x_0, x_1)$ is homotopy equivalent to $R^{\mathcal{L}}(X)(x_0, x_1)$ (resp. induces iso on a range of homotopy groups)

Again: Topologically trivial pieces $\rightsquigarrow$ All topology resides in glueing information. Huge trace space replaced by a much smaller space!

An important special case II

Reachability. For a given collection of pieces and layers $\mathcal{L} = \bigcup L_{ij}$ in X that is unavoidable from x_0 to x_1, let $R^{\mathcal{L}}(X)(x_0, x_1) = \{(x_{i_0j_0}, \ldots, x_{i_nj_n}) \in L_{i_0j_0} \times \cdots \times L_{i_nj_n} \mid \vec{P}(X_{i_k})(x_{i_kj_k}, x_{i_{k+1}j_{k+1}}) \neq \emptyset, n \geq 0\}$ denote the space of mutually reachable points in the given layers.

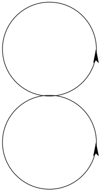
Corollary

If, moreover, all path spaces $\vec{T}(X_{i_k})(x_{i_{k-1}}, x_{i_k}), x_{i_{k-1}} \in L_{i_{k-1}i_k}, x_{i_k} \in L_{i_ki_{k+1}}$ are contractible and locally contractible (resp. highly connected), then $\vec{T}(X)(x_0, x_1)$ is homotopy equivalent to $R^{\mathcal{L}}(X)(x_0, x_1)$ (resp. induces iso on a range of homotopy groups)

Again: Topologically trivial pieces $\rightsquigarrow$ All topology resides in glueing information. Huge trace space replaced by a much smaller space!

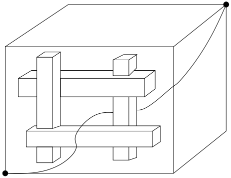
Examples

1



A wedge of two directed circles $X = \vec{S}^1 \vee_{x_0} \vec{S}^1$:
$\vec{T}(X)(x_0, x_0) \simeq \{1,2\}^*$.
(Choose $L_i = \{x_i\}, i = 1,2$ with $x_i \neq x_0$ on the two branches).

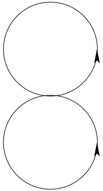
2



Y = cube with two wedges deleted:
$\vec{T}(Y)(\mathbf{0}, \mathbf{1}) \simeq * \sqcup (S^1 \vee S^1)$.
(L_i two vertical cuts through the wedges; product is homotopy equivalent to torus; reachability $\rightsquigarrow$ two components, one of which is contractible, the other a thickening of $S^1 \vee S^1 \subset S^1 \times S^1$.)

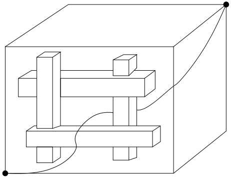
Examples

1



A wedge of two directed circles $X = \vec{S}^1 \vee_{x_0} \vec{S}^1$:
$\vec{T}(X)(x_0, x_0) \simeq \{1, 2\}^*$.
(Choose $L_i = \{x_i\}, i = 1, 2$ with $x_i \neq x_0$ on the two branches).

2



Y = cube with two wedges deleted:
$\vec{T}(Y)(\mathbf{0}, \mathbf{1}) \simeq * \sqcup (S^1 \vee S^1)$.
(L_i two vertical cuts through the wedges; product is homotopy equivalent to torus; reachability $\rightsquigarrow$ two components, one of which is contractible, the other a thickening of $S^1 \vee S^1 \subset S^1 \times S^1$.)

Inductive Calculations concerning $\vec{T}(X)(x_0, x_1)$

In many cases, one can establish the connectivity of $\vec{T}(X)(x_0, x_1)$ by studying the spaces of mutually reachable pairs $\{(x_{ki}, x_{ij}) \in L_{ki} \times L_{ij} \mid x_{ki} \preceq x_{ij}\}$.

Theorem

If all spaces of mutually reachable pairs are k-connected, then $\vec{T}(X)(x_0, x_1)$ is k-connected.
If not connected, one can often inductively determine the connected components (in particular their number).

Maximal cube paths

Let X denote the geometric realization of a finite pre-cubical complex ($\square$-set) M, i.e., $X = \coprod (M_n \times \vec{I}^n)_{/\simeq}$.
X consists of "cells" e_α homeomorphic to I^{n_α}. A cell is called maximal if it is not in the image of a boundary map $\partial^\pm$.
D-path structure $\vec{P}(X)$ inherited from the $\vec{P}(\vec{I}^n)$ by "pasting".

Definition

- A maximal cube path in a pre-cubical set is a sequence $S = (e_{\alpha_1}, \dots, e_{\alpha_k})$ of maximal cells such that $\partial^+ e_{\alpha_i} \cap \partial^- e_{\alpha_{i+1}}$ is a (nonempty) cell of maximal dimension. Its length $|S| = k$.
- A d-path p (or its trace) is included in a maximal cube path S if there is a partition of the parameter interval I such that $p(I_j) \subset e_{\alpha_j}$, $j \leq k$.

Maximal cube paths

Let X denote the geometric realization of a finite pre-cubical complex ($\Box$-set) M, i.e., $X = \coprod(M_n \times \vec{I}^n)_{/\simeq}$.
X consists of “cells” e_α homeomorphic to I^{n_α}. A cell is called maximal if it is not in the image of a boundary map $\partial^\pm$.
D-path structure $\vec{P}(X)$ inherited from the $\vec{P}(\vec{I}^n)$ by “pasting”.

Definition

- A maximal cube path in a pre-cubical set is a sequence $S = (e_{\alpha_1}, \dots, e_{\alpha_k})$ of maximal cells such that $\partial^+ e_{\alpha_i} \cap \partial^- e_{\alpha_{i+1}}$ is a (nonempty) cell of maximal dimension. Its length $|S| = k$.
- A d-path p (or its trace) is included in a maximal cube path S if there is a partition of the parameter interval I such that $p(I_j) \subset e_{\alpha_j}$, $j \leq k$.

Maximal cube paths

Let X denote the geometric realization of a finite pre-cubical complex ($\square$-set) M, i.e., $X = \coprod(M_n \times \vec{I}^n)_{/\simeq}$.
X consists of "cells" e_α homeomorphic to I^{n_α}. A cell is called **maximal** if it is not in the image of a boundary map $\partial^\pm$.
D-path structure $\vec{P}(X)$ inherited from the $\vec{P}(\vec{I}^n)$ by "pasting".

Definition

- A **maximal cube path** in a pre-cubical set is a sequence $S = (e_{\alpha_1}, \dots, e_{\alpha_k})$ of maximal cells such that $\partial^+ e_{\alpha_i} \cap \partial^- e_{\alpha_{i+1}}$ is a (nonempty) cell of maximal dimension. Its length $|S| = k$.
- A d-path p (or its trace) is **included** in a maximal cube path S if there is a partition of the parameter interval I such that $p(I_j) \subset e_{\alpha_j}$, $j \leq k$.

Simplicial complex representations of trace spaces

Consider the posets $\mathcal{C}_k(X)(e)$ consisting of i-tuples of maximal cube paths of length k in X starting with e and including a common d-path. The nerve of this poset (ordered by inclusion) is a simplicial set (complex) $\mathcal{S}_k(X)(e)$ with

Vertices maximal cube paths of length k in X (start at e);

i-simplices $(i+1)$-tuples of maximal cube paths of length k starting with e including a common d-path.

(Modified version: only cube paths that end with e')

Theorem

$\vec{T}(X)(x_0, x_1) \simeq \coprod_k \mathcal{S}_k(X)(e_0, e_1); x_0 \in e_0, x_1 \in e_1.$

Proof.

The subspace of d-paths included in a maximal cube path (or included in several of them) is contractible. Nerve lemma! □

Simplicial complex representations of trace spaces

Consider the posets $\mathcal{C}_k(X)(e)$ consisting of i-tuples of maximal cube paths of length k in X starting with e and including a common d-path. The nerve of this poset (ordered by inclusion) is a simplicial set (complex) $\mathcal{S}_k(X)(e)$ with

Vertices maximal cube paths of length k in X (start at e);

i-simplices $(i+1)$-tuples of maximal cube paths of length k starting with e including a common d-path.

(Modified version: only cube paths that end with e')

Theorem

$\vec{T}(X)(x_0, x_1) \simeq \coprod_k \mathcal{S}_k(X)(e_0, e_1); x_0 \in e_0, x_1 \in e_1.$

Proof.

The subspace of d-paths included in a maximal cube path (or included in several of them) is contractible. Nerve lemma! □

Simplicial complex representations of trace spaces

Consider the posets $\mathcal{C}_k(X)(e)$ consisting of i-tuples of maximal cube paths of length k in X starting with e and including a common d-path. The nerve of this poset (ordered by inclusion) is a simplicial set (complex) $\mathcal{S}_k(X)(e)$ with

- **Vertices** maximal cube paths of length k in X (start at e);
- **i-simplices** $(i+1)$-tuples of maximal cube paths of length k starting with e including a common d-path.

(Modified version: only cube paths that end with e')

Theorem

$\vec{T}(X)(x_0, x_1) \simeq \coprod_k \mathcal{S}_k(X)(e_0, e_1); x_0 \in e_0, x_1 \in e_1.$

Proof.

The subspace of d-paths included in a maximal cube path (or included in several of them) is contractible. Nerve lemma! □

Simplicial complex representations of trace spaces

Consider the posets $\mathcal{C}_k(X)(e)$ consisting of i-tuples of maximal cube paths of length k in X starting with e and including a common d-path. The nerve of this poset (ordered by inclusion) is a simplicial set (complex) $\mathcal{S}_k(X)(e)$ with

Vertices maximal cube paths of length k in X (start at e);

i-simplices $(i+1)$-tuples of maximal cube paths of length k starting with e including a common d-path.

(Modified version: only cube paths that end with e')

Theorem

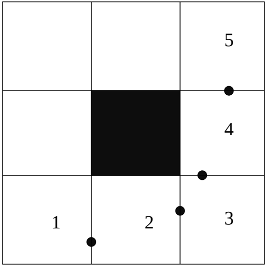
$\vec{T}(X)(x_0, x_1) \simeq \coprod_k \mathcal{S}_k(X)(e_0, e_1); x_0 \in e_0, x_1 \in e_1.$

Proof.

The subspace of d-paths included in a maximal cube path (or included in several of them) is contractible. Nerve lemma! □

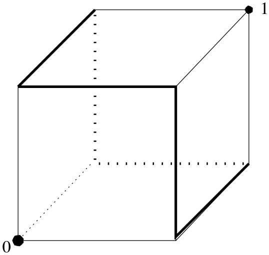
Simple examples

1



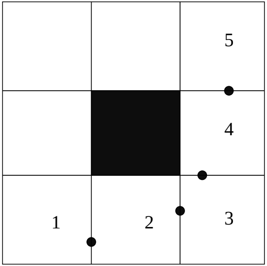
The two only maximal square paths from **0** to **1**, each of them contributing a vertex. Empty intersection.
$\mathcal{S}(X)(\mathbf{0},\mathbf{1}) \simeq * \sqcup *$.

2

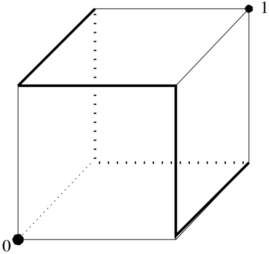


$X = \partial\vec{I}^n$. The maximal square paths from **0** to **1** have length 2.
$\mathcal{S}(\partial\vec{I}^n)(\mathbf{0},\mathbf{1}) = \partial_{\pm}\vec{I}^n \simeq S^{n-2}$.

Simple examples

1 

The two only maximal square paths from **0** to **1**, each of them contributing a vertex. Empty intersection. $\mathcal{S}(X)(\mathbf{0},\mathbf{1}) \simeq * \sqcup *$.

2 

$X = \partial \vec{I}^n$. The maximal square paths from **0** to **1** have length 2. $\mathcal{S}(\partial \vec{I}^n)(\mathbf{0},\mathbf{1}) = \partial_\pm \vec{I}^n \simeq S^{n-2}$.

Future work

on the algebraic topology of trace spaces

- Is there an automatic way to place consecutive "antidiagonal cut" layers in complexes corresponding to PV-programs that allow to compare path spaces to subspaces of the products of these layers?
- Maximal cube paths and the d-paths included in them come in "rounds" (length). This gives hope for inductive calculations (as in the work of Herlihy, Rajsbaum and others in distributed computing).
- Explore the combinatorial alg. topology of trace spaces
  - with fixed end points and
  - what happens under variations of end points.
- Make this analysis useful for the determination of components and vice versa (extend the work of Fajstrup, Goubault, Haucourt, MR)
- d-geodesics? instead of d-paths?

Future work

on the algebraic topology of trace spaces

- Is there an automatic way to place consecutive "antidiagonal cut" layers in complexes corresponding to PV-programs that allow to compare path spaces to subspaces of the products of these layers?
- Maximal cube paths and the d-paths included in them come in "rounds" (length). This gives hope for inductive calculations (as in the work of Herlihy, Rajsbaum and others in distributed computing).
- Explore the combinatorial alg. topology of trace spaces
  - with fixed end points and
  - what happens under variations of end points.
- Make this analysis useful for the determination of components and vice versa (extend the work of Fajstrup, Goubault, Haucourt, MR)
- d-geodesics? instead of d-paths?

Future work

on the algebraic topology of trace spaces

- Is there an automatic way to place consecutive "antidiagonal cut" layers in complexes corresponding to PV-programs that allow to compare path spaces to subspaces of the products of these layers?
- Maximal cube paths and the d-paths included in them come in "rounds" (length). This gives hope for inductive calculations (as in the work of Herlihy, Rajsbaum and others in distributed computing).
- Explore the combinatorial alg. topology of trace spaces
  - with fixed end points and
  - what happens under variations of end points.
- Make this analysis useful for the determination of components and vice versa (extend the work of Fajstrup, Goubault, Haucourt, MR)
- d-geodesics? instead of d-paths?

Future work

on the algebraic topology of trace spaces

- Is there an automatic way to place consecutive "antidiagonal cut" layers in complexes corresponding to PV-programs that allow to compare path spaces to subspaces of the products of these layers?
- Maximal cube paths and the d-paths included in them come in "rounds" (length). This gives hope for inductive calculations (as in the work of Herlihy, Rajsbaum and others in distributed computing).
- Explore the combinatorial alg. topology of trace spaces
  - with fixed end points and
  - what happens under variations of end points.
- Make this analysis useful for the determination of components and vice versa (extend the work of Fajstrup, Goubault, Haucourt, MR)
- d-geodesics? instead of d-paths?

Future work

on the algebraic topology of trace spaces

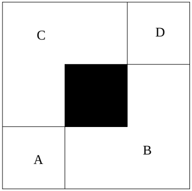
- Is there an automatic way to place consecutive "antidiagonal cut" layers in complexes corresponding to PV-programs that allow to compare path spaces to subspaces of the products of these layers?
- Maximal cube paths and the d-paths included in them come in "rounds" (length). This gives hope for inductive calculations (as in the work of Herlihy, Rajsbaum and others in distributed computing).
- Explore the combinatorial alg. topology of trace spaces
  - with fixed end points and
  - what happens under variations of end points.
- Make this analysis useful for the determination of components and vice versa (extend the work of Fajstrup, Goubault, Haucourt, MR)
- d-geodesics? instead of d-paths?

Categorical organization

First tool: The fundamental category

$\vec{\pi}_1(X)$ of a d-space X [Grandis:03, FGHR:04]:

- Objects: points in X
- Morphisms: d- or dihomotopy classes of d-paths in X
- Composition: from concatenation of d-paths



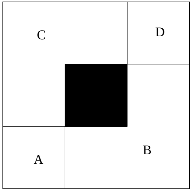
Property: van Kampen theorem (M. Grandis)
Drawbacks: Infinitely many objects. Calculations?
Question: How much does $\vec{\pi}_1(X)(x,y)$ depend on (x,y)?
Remedy: Localization, component category. [FGHR:04, GH:06]
Problem: This "compression" works only for loopfree categories

Categorical organization

First tool: The fundamental category

$\vec{\pi}_1(X)$ of a d-space X [Grandis:03, FGHR:04]:

- Objects: points in X
- Morphisms: d- or dihomotopy classes of d-paths in X
- Composition: from concatenation of d-paths



Property: van Kampen theorem (M. Grandis)
Drawbacks: Infinitely many objects. Calculations?
Question: How much does $\vec{\pi}_1(X)(x,y)$ depend on (x,y)?
Remedy: Localization, component category. [FGHR:04, GH:06]
Problem: This “compression” works only for loopfree categories

Preorder categories

Getting organised with indexing categories

A d-space structure on X induces the preorder $\preceq$:

$$x \preceq y \Leftrightarrow \vec{T}(X)(x,y) \neq \emptyset$$

and an indexing preorder category $\vec{D}(X)$ with

- Objects: (end point) pairs $(x,y), x \preceq y$
- Morphisms: $\vec{D}(X)((x,y),(x',y')) := \vec{T}(X)(x',x) \times \vec{T}(X)(y,y')$:

$$\begin{tikzcd} x' \arrow[r, bend left] \arrow[r, bend right] & x \arrow[r, "\preceq"] & y \arrow[r, bend left] \arrow[r, bend right] & y' \end{tikzcd}$$

- Composition: by pairwise contra-, resp. covariant concatenation.

A d-map $f : X \to Y$ induces a functor $\vec{D}(f) : \vec{D}(X) \to \vec{D}(Y)$.

Preorder categories

Getting organised with indexing categories

A d-space structure on X induces the preorder $\preceq$:

$$x \preceq y \Leftrightarrow \vec{T}(X)(x,y) \neq \emptyset$$

and an indexing preorder category $\vec{D}(X)$ with

- Objects: (end point) pairs $(x,y), x \preceq y$
- Morphisms:
  $\vec{D}(X)((x,y),(x',y')) := \vec{T}(X)(x',x) \times \vec{T}(X)(y,y')$:

$$\begin{tikzcd} x' \arrow[r, bend left] \arrow[r, bend right] & x \arrow[r, "\preceq"] & y \arrow[r, bend left] \arrow[r, bend right] & y' \end{tikzcd}$$

- Composition: by pairwise contra-, resp. covariant concatenation.

A d-map $f : X \to Y$ induces a functor $\vec{D}(f) : \vec{D}(X) \to \vec{D}(Y)$.

Preorder categories

Getting organised with indexing categories

A d-space structure on X induces the preorder $\preceq$:

$$x \preceq y \Leftrightarrow \vec{T}(X)(x,y) \neq \emptyset$$

and an indexing **preorder category** $\vec{D}(X)$ with

- **Objects**: (end point) **pairs** $(x,y), x \preceq y$
- **Morphisms**:
  $\vec{D}(X)((x,y),(x',y')) := \vec{T}(X)(x',x) \times \vec{T}(X)(y,y')$:

$$\begin{tikzcd} x' \arrow[r, bend left] \arrow[r, bend right] & x \arrow[r, "\preceq"] & y \arrow[r, bend left] \arrow[r, bend right] & y' \end{tikzcd}$$

- **Composition**: by pairwise contra-, resp. covariant concatenation.

A d-map $f : X \to Y$ induces a functor $\vec{D}(f) : \vec{D}(X) \to \vec{D}(Y)$.

The trace space functor

Preorder categories organise the trace spaces

The preorder category organises X via the
trace space functor $\vec{T}^X : \vec{D}(X) \to Top$

- $\vec{T}^X(x,y) := \vec{T}(X)(x,y)$
- $\vec{T}^X(\sigma_x, \sigma_y)$:

$$\begin{array}{ccc} \vec{T}(X)(x,y) & \longrightarrow & \vec{T}(X)(x',y') \\ [\sigma] & \longmapsto & [\sigma_x * \sigma * \sigma_y] \end{array}$$

Homotopical variant $\vec{D}_\pi(X)$ with morphisms
$\vec{D}_\pi(X)((x,y),(x',y')) := \vec{\pi}_1(X)(x',x) \times \vec{\pi}_1(X)(y,y')$
and trace space functor $\vec{T}_\pi^X : \vec{D}_\pi(X) \to Ho-Top$ (with homotopy classes as morphisms).

The trace space functor

Preorder categories organise the trace spaces

The preorder category organises X via the **trace space functor** $\vec{T}^X : \vec{D}(X) \to Top$

- $\vec{T}^X(x,y) := \vec{T}(X)(x,y)$
- $\vec{T}^X(\sigma_x,\sigma_y) :$

$$\begin{array}{ccc} \vec{T}(X)(x,y) & \longrightarrow & \vec{T}(X)(x',y') \\ & & \\ [\sigma] & \longmapsto & [\sigma_x * \sigma * \sigma_y] \end{array}$$

Homotopical variant $\vec{D}_\pi(X)$ with morphisms $\vec{D}_\pi(X)((x,y),(x',y')) := \vec{\pi}_1(X)(x',x) \times \vec{\pi}_1(X)(y,y')$ and trace space functor $\vec{T}^X_\pi : \vec{D}_\pi(X) \to Ho-Top$ (with homotopy classes as morphisms).

Sensitivity with respect to variations of end points

Questions from a persistence point of view

- How much does (the homotopy type of) $\vec{T}^X(x,y)$ depend on (small) changes of x,y?
- Which concatenation maps $\vec{T}^X(\sigma_x,\sigma_y):\vec{T}^X(x,y)\to\vec{T}^X(x',y'),\ [\sigma]\mapsto[\sigma_x*\sigma*\sigma_y]$ are homotopy equivalences, induce isos on homotopy, homology groups etc.?
- The persistence point of view: Homology classes etc. are born (at certain branchings/mergings) and may die (analogous to the framework of G. Carlsson etal.)
- Are there "components" with (homotopically/homologically) stable dipath spaces (between them)? Are there borders ("walls") at which changes occur?

Sensitivity with respect to variations of end points

Questions from a persistence point of view

- How much does (the homotopy type of) $\vec{T}^X(x,y)$ depend on (small) changes of x,y?
- Which concatenation maps $\vec{T}^X(\sigma_x,\sigma_y): \vec{T}^X(x,y) \to \vec{T}^X(x',y'),\ [\sigma] \mapsto [\sigma_x * \sigma * \sigma_y]$ are homotopy equivalences, induce isos on homotopy, homology groups etc.?
- The persistence point of view: Homology classes etc. are born (at certain branchings/mergings) and may die (analogous to the framework of G. Carlsson etal.)
- Are there "components" with (homotopically/homologically) stable dipath spaces (between them)? Are there borders ("walls") at which changes occur?

Sensitivity with respect to variations of end points

Questions from a persistence point of view

- How much does (the homotopy type of) $\vec{T}^X(x,y)$ depend on (small) changes of x,y?
- Which concatenation maps
$\vec{T}^X(\sigma_x,\sigma_y):\vec{T}^X(x,y)\to\vec{T}^X(x',y'),\ [\sigma]\mapsto[\sigma_x*\sigma*\sigma_y]$
are homotopy equivalences, induce isos on homotopy, homology groups etc.?
- The persistence point of view: Homology classes etc. are born (at certain branchings/mergings) and may die (analogous to the framework of G. Carlsson etal.)
- Are there "components" with (homotopically/homologically) stable dipath spaces (between them)? Are there borders ("walls") at which changes occur?

Sensitivity with respect to variations of end points

Questions from a persistence point of view

- How much does (the homotopy type of) $\vec{T}^X(x,y)$ depend on (small) changes of x,y?
- Which concatenation maps $\vec{T}^X(\sigma_x,\sigma_y) : \vec{T}^X(x,y) \to \vec{T}^X(x',y'),\ [\sigma] \mapsto [\sigma_x * \sigma * \sigma_y]$ are homotopy equivalences, induce isos on homotopy, homology groups etc.?
- The persistence point of view: Homology classes etc. are born (at certain branchings/mergings) and may die (analogous to the framework of G. Carlsson etal.)
- Are there “components” with (homotopically/homologically) stable dipath spaces (between them)? Are there borders (“walls”) at which changes occur?

Examples of component categories
Standard example

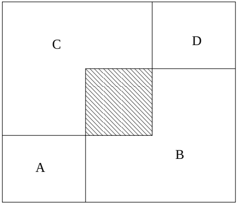


Figure: Standard example ↑ and component category →

$$
\begin{tikzcd}
AA \arrow[drr] \arrow[ddr] & & & & BB \arrow[dll] \arrow[ddl] \\
 & & AB \arrow[d] & \# & \\
 & AC \arrow[r] & AD & BD \arrow[l] & \\
 & \# & CD \arrow[u] & & \\
CC \arrow[uur] \arrow[urr] & & & & DD \arrow[ull] \arrow[uul]
\end{tikzcd}
$$

Components A, B, C, D – or rather $AA, AB, AC, AD, BB, BD, CC, CD, DD$.

Examples of component categories
Standard example

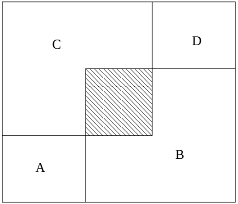
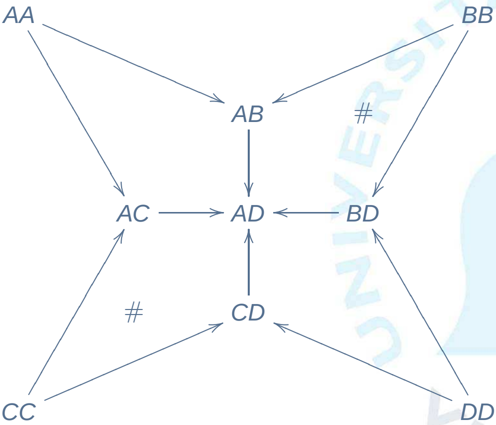


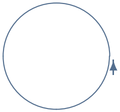
Figure: Standard example ↑ and component category →



Components A, B, C, D – or rather $AA, AB, AC, AD, BB, BD, CC, CD, DD$.

Examples of component categories
Oriented circle – with loops!

$X \qquad = \qquad \vec{S}^1$



$$\begin{tikzcd} \mathcal{C} : \Delta \arrow[r, bend left, "a"] & \bar{\Delta} \arrow[l, bend left, "b"] \end{tikzcd}$$

Δ the diagonal, $\bar{\Delta}$ its complement.
$\mathcal{C}$ is the free category generated by a, b.

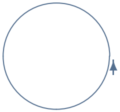
oriented circle

- Remark that the components are no longer products!
- It is essential in order to get a discrete component category to use an indexing category taking care of pairs (source, target).

Examples of component categories

Oriented circle – with loops!

$X \quad = \quad \vec{S}^1$



$$\mathcal{C}: \begin{tikzcd} \Delta \arrow[r, bend left, "a"] & \bar{\Delta} \arrow[l, bend left, "b"] \end{tikzcd}$$

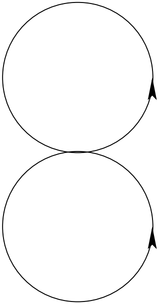
Δ the diagonal, $\bar{\Delta}$ its complement. $\mathcal{C}$ is the free category generated by a, b.

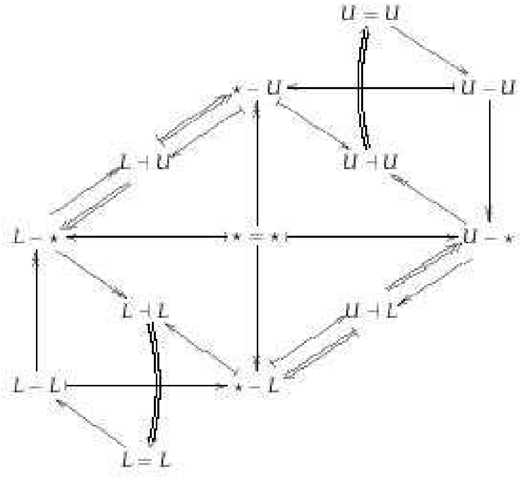
oriented circle

- Remark that the components are no longer products!
- It is essential in order to get a discrete component category to use an indexing category taking care of pairs (source, target).

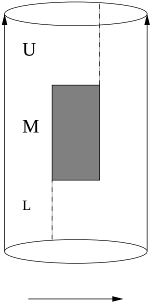
The component category of a wedge of two oriented circles

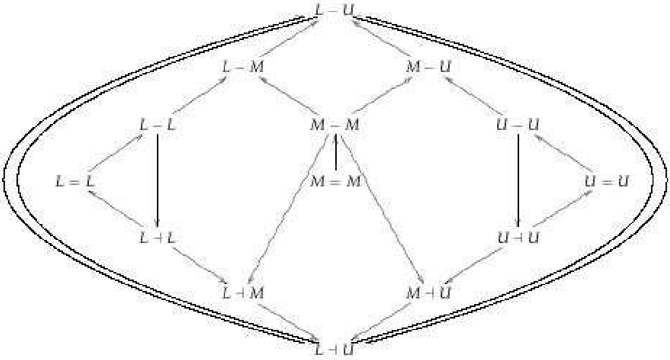
$$X = \vec{S}^1 \vee \vec{S}^1$$





The component category of an oriented cylinder with a deleted rectangle





Concluding remarks

- Component categories contain the essential information given by (algebraic topological invariants of) d-path spaces
- Compression via component categories is an antidote to the state space explosion problem
- Some of the ideas (for the fundamental category) are implemented and have been tested for huge industrial software from EDF (Éric Goubault & Co., CEA)
- Much more theoretical and practical work remains to be done!

Thanks for your attention!
Questions? Comments?

Concluding remarks

- Component categories contain the essential information given by (algebraic topological invariants of) d-path spaces
- Compression via component categories is an antidote to the state space explosion problem
- Some of the ideas (for the fundamental category) are implemented and have been tested for huge industrial software from EDF (Éric Goubault & Co., CEA)
- Much more theoretical and practical work remains to be done!

Thanks for your attention!
Questions? Comments?

Concluding remarks

- Component categories contain the essential information given by (algebraic topological invariants of) d-path spaces
- Compression via component categories is an antidote to the state space explosion problem
- Some of the ideas (for the fundamental category) are implemented and have been tested for huge industrial software from EDF (Éric Goubault & Co., CEA)
- Much more theoretical and practical work remains to be done!

Thanks for your attention!
Questions? Comments?

Concluding remarks

- Component categories contain the essential information given by (algebraic topological invariants of) d-path spaces
- Compression via component categories is an antidote to the state space explosion problem
- Some of the ideas (for the fundamental category) are implemented and have been tested for huge industrial software from EDF (Éric Goubault & Co., CEA)
- Much more theoretical and practical work remains to be done!

Thanks for your attention!

Questions? Comments?

Concluding remarks

- Component categories contain the essential information given by (algebraic topological invariants of) d-path spaces
- Compression via component categories is an antidote to the state space explosion problem
- Some of the ideas (for the fundamental category) are implemented and have been tested for huge industrial software from EDF (Éric Goubault & Co., CEA)
- Much more theoretical and practical work remains to be done!

Thanks for your attention!
Questions? Comments?