Directed topology. An introduction

Martin Raussen

Department of mathematical sciences
Aalborg University, Denmark

Université de Neuchâtel, 9.1.2007

Outline

Outline

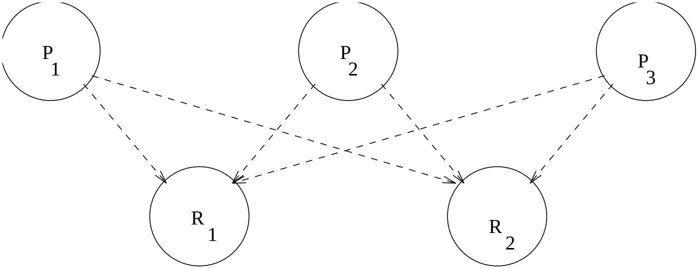
1. Motivations, mainly from Concurrency Theory
2. Directed topology: algebraic topology with a twist
3. A categorical framework (with examples)
4. “Compression” of ditopological categories: generalized congruences via homotopy flows

Main Collaborators:

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

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.

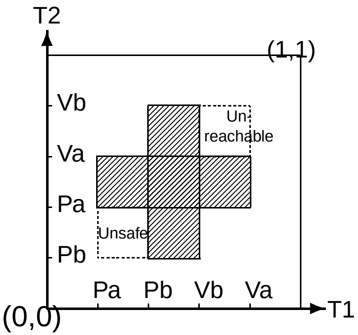
Semaphores!

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

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

Schedules in "progress graphs"

The Swiss flag example



PV-diagram from

$P_1 : P_a P_b V_b V_a$

$P_2 : P_b P_a V_a V_b$

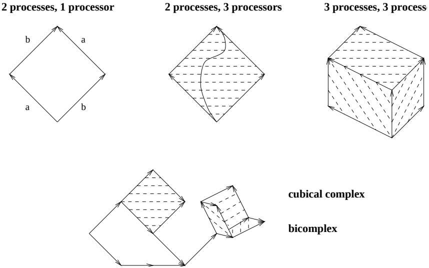
Executions are directed paths – since time flow is irreversible – avoiding a forbidden region (shaded).

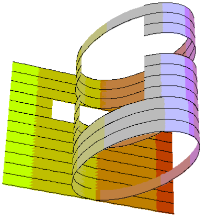
Dipaths that are dihomotopic (through a 1-parameter deformation consisting of dipaths) correspond to equivalent executions.

Deadlocks, unsafe and unreachable regions may occur.

Higher dimensional automata
seen as (geometric realizations of) cubical sets

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



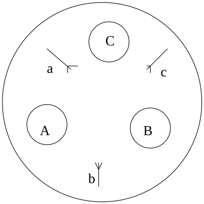


Squares/cubes/hypercubes are filled in iff actions on boundary are independent.

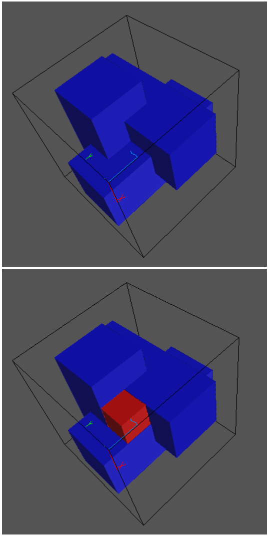
Higher dimensional automata are cubical sets:

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

Higher dimensional automata
Example: Dining philosophers



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.

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.

An 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.

A framework for directed topology

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.

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

Definition

- A continuous map $H : X \times I \to Y$ is called a homotopy.
- Continuous maps $f, g : X \to Y$ are called homotopic to each other 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.
- 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.

D-maps, Dihomotopy, d-homotopy

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

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

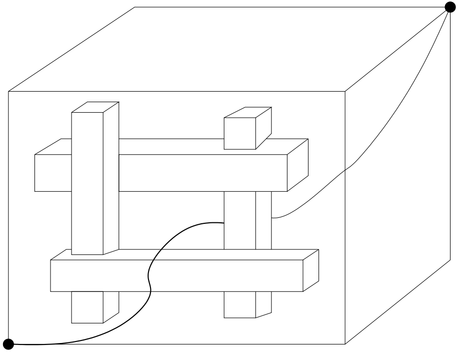
special case: $\vec{P}(I) = \{\sigma \in I^I | \sigma$ nondecreasing reparametrization$\}, \vec{I} = (I, \vec{P}(I))$.
Then $\vec{P}(X) =$ set of d-maps from $\vec{I}$ to X.

- Dihomotopy $H : X \times I \to Y$, every H_t a d-map
- elementary d-homotopy = d-map $H : X \times \vec{I} \to Y$ – $H_0 = f \xrightarrow{H} g = H_1$
- d-homotopy: symmetric and transitive closure ("zig-zag")

L. Fajstrup, 05: In cubical models (for concurrency, e.g., HDAs), the two notions agree for d-paths ($X = \vec{I}$). In general, they do not.

Dihomotopy is finer than homotopy with fixed endpoints

Example: Two wedges in the forbidden region

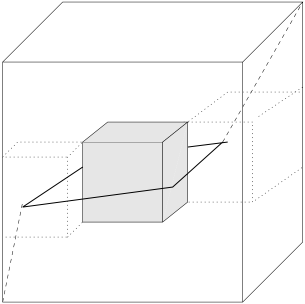


All dipaths from minimum to maximum are homotopic.
A dipath through the “hole” is not dihomotopic to a dipath on the boundary.

The twist has a price

Neither homogeneity nor cancellation nor group structure

In ordinary topology, it suffices to study loops in a space X with a given start=end point x (one per path component). Moreover: "Loops up to homotopy" $\rightsquigarrow$ fundamental group $\pi_1(X, x)$ – concatenation, inversion!



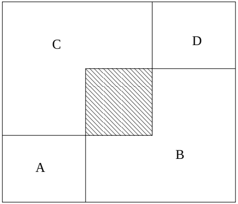
"Birth and death" of dihomotopy classes

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

A first remedy: 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: “Compression” works only for loopfree categories

Concepts from algebraic topology 2
Homotopy groups, homology groups, homotopy equivalences

- $\pi_n(X,x) = [(S^n,*),(X,x)]$; group structure: $S^n \to S^n \vee S^n$, abelian for $n > 1$. Easy to define, difficult to calculate.
- Homology and cohomology groups $H_n(X)$ and $H^n(X)$: abelian groups; definition more complicated, but essentially calculable for reasonable topological spaces. $H_0(X)$ free abelian group on path components of X. $H_1(X) = \pi_1(X)/_{[\pi_1(X),\pi_1(X)]}$.
- A continuous map $f:(X,x) \to (Y,y)$ induces group homomorphisms $f_\#: \pi_n(X,x) \to \pi_n(Y,y)$, and $f_*: H_n(X) \to H_n(Y)$, $n \in \mathbf{N}$. Homotopic maps induce the same homomorphism (homotopy invariance). Functoriality: $(g \circ f)_\# = g_\# \circ f_\#, (g \circ f)_* = g_* \circ f_*$.
- A continuos map $f: X \to Y$ is a homotopy equivalence if there exists a homotopy inverse $g: Y \to X$ satisfying $g \circ f \simeq id_X$ and $f \circ g \simeq id_Y$. Homotopy equivalent spaces have isomorphic homotopy and (co)homology groups.

Technique: Traces – and trace categories
Getting rid of increasing reparametrizations

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 $\varphi \circ \alpha = \psi \circ \beta$.
(Fahrenberg-R., 06): 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 $\vec{T}(f) : \vec{T}(X) \to \vec{T}(Y)$.
Variant: $\vec{R}(X)(x, y)$ consists of regular d-paths (not constant on any non-trivial interval $J \subset I$). The contractible group $Homeo_+(I)$ of increasing homeomorphisms acts on these – freely if $x \neq y$.

Theorem (FR:06)

$\vec{R}(X)(x, y)/_{\simeq} \to \vec{P}(X)(x, y)/_{\simeq}$ is a homeomorphism.

Preorder categories

Getting organised with indexing categories

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

$$x' \rightrightarrows x \xrightarrow{\preceq} y \rightrightarrows y'$$

- 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 $\vec{T}^X : \vec{D}(X) \to Top$

- $\vec{T}^X(x,y) := \vec{T}(X)(x,y)$
- $\vec{T}^X(\sigma_x, \sigma_y)$: $\vec{T}(X)(x,y) \longrightarrow \vec{T}(X)(x',y')$

$$[\sigma] \longmapsto [\sigma_x * \sigma * \sigma_y]$$

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

A persistence point of view

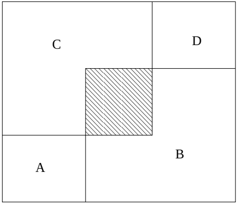
- How much does (the homotopy type of) $\vec{T}^X(x, y)$ depend on (small) changes of x, y?
- Which concatenation maps $\vec{T}^X(\sigma_x, \sigma_y) : \vec{T}^X(x,y) \to \vec{T}^X(x',y'),\ [\sigma] \mapsto [\sigma_x * \sigma * \sigma_y]$ are homotopy equivalences, induce isos on homotopy, homology groups 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?
- $\leadsto$ need a lot of bookkeeping!

Dihomology $\vec{H}_*$

- For every d-space X, there are homology functors

$$\vec{H}_{*+1}(X) = H_* \circ \vec{T}^X_\pi : \vec{D}_\pi(X) \to Ab,\ (x,y) \mapsto H_*(\vec{T}(X)(x,y))$$

  capturing homology of all relevant d-path spaces in X and the effects of the concatenation structure maps.
- A d-map $f : X \to Y$ induces a natural transformation $\vec{H}_{*+1}(f)$ from $\vec{H}_{*+1}(X)$ to $\vec{H}_{*+1}(Y)$.
- Properties? Calculations? Not much known in general. A master's student has studied this topic for X a cubical complex (its geometric realization) by constructing a cubical model for d-path spaces.
- Higher dihomotopy functors $\vec{\pi}_*$: in the same vain, a bit more complicated to define, since they have to reflect choices of base paths.



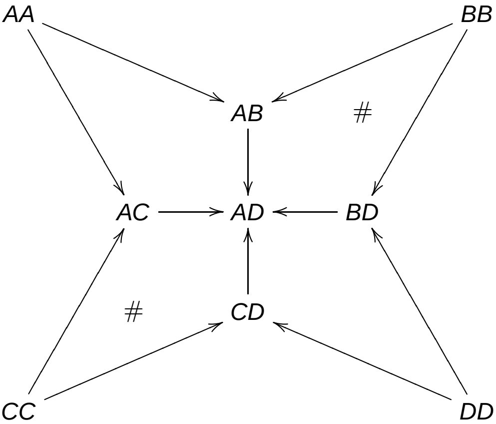
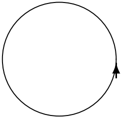


Figure: Standard example and component category

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

Examples of component categories
Oriented circle

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



oriented circle

$\mathcal{C}: \Delta \underset{b}{\overset{a}{\rightleftarrows}} \bar{\Delta}$

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

- 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).

Compression: Generalized congruences and quotient categories

Bednarczyk, Borzyszkowski, Pawlowski, TAC 1999

How to identify morphisms in a category between different objects in an organised manner?
Start with an equivalence relation $\simeq$ on the objects.
A generalized congruence is an equivalence relation on non-empty sequences $\varphi = (f_1 \dots f_n)$ of morphisms with $cod(f_i) \simeq dom(f_{i+1})$ ($\simeq$-paths) satisfying

1. $\varphi \simeq \psi \Rightarrow dom(\varphi) \simeq dom(\psi), codom(\varphi) \simeq codom(\psi)$
2. $a \simeq b \Rightarrow id_a \simeq id_b$
3. $\varphi_1 \simeq \psi_1, \varphi_2 \simeq \psi_2, cod(\varphi_1) \simeq dom(\varphi_2) \Rightarrow \varphi_2\varphi_1 \simeq \psi_2\psi_1$
4. $cod(f) = dom(g) \Rightarrow f \circ g \simeq (fg)$

Quotient category $\mathcal{C}/\simeq$: Equivalence classes of objects and of $\simeq$-paths; composition: $[\varphi] \circ [\psi] = [\varphi\psi]$.

Tool: Homotopy flows
used to define a significant generalized congruence

A d-map $H : X \times \vec{I} \to X$ is called a homotopy flow if

future $H_0 = id_X \xrightarrow{H} f = H_1$

past $H_0 = g \xrightarrow{H} id_X = H_1$

H_t is not a homeomorphism, in general; the flow is irreversible.
H and f are called

automorphic if $\vec{T}(H_t) : \vec{T}(X)(x, y) \to \vec{T}(X)(H_t x, H_t y)$ is a homotopy equivalence for all $x \preceq y, t \in I$.

Automorphisms are closed under composition – concatenation of homotopy flows!
$Aut_+(X), Aut_-(X)$ monoids of automorphisms.
Variations: $\vec{T}(H_t)$ induces isomorphisms on homology groups, homotopy groups....

Automorphic homotopy flows give rise to generalized congruences

Let X be a d-space and $Aut_{\pm}(X)$ the monoid of all (future/past) automorphisms.

"Flow lines" are used to identify objects (pairs of points) and morphisms (classes of dipaths) in an organized manner.

$Aut_{\pm}(X)$ gives rise to a generalized congruence on the (homotopy) preorder category $\vec{D}_{\pi}(X)$ as the symmetric and transitive congruence closure of what will be described on the next slide.

The resulting component category has as its objects the components connected by equivalence classes of dipaths.

Congruences and component categories

- $(x,y) \simeq (x',y'),\ f_+ : (x,y) \leftrightarrow (x',y') : f_-, \qquad f_\pm \in Aut_\pm(X)$
- $$(x,y) \stackrel{(\sigma_1,\sigma_2)}{\longrightarrow} (u,v) \simeq (x',y') \stackrel{(\tau_1,\tau_2)}{\longrightarrow} (u',v'),$$
  $f_+ : (x,y,u,v) \leftrightarrow (x',y',u',v') : f_-, \qquad f_\pm \in Aut_\pm(X)$, and
  $$\begin{array}{ccc} \vec{T}(X)(x',y') & \stackrel{(\tau_1,\tau_2)}{\longrightarrow} & \vec{T}(X)(u',v') \\ \vec{T}(f_+) \uparrow \downarrow \vec{T}(f_-) & & \vec{T}(f_+) \uparrow \downarrow \vec{T}(f_-) \\ \vec{T}(X)(x,y) & \stackrel{(\sigma_1,\sigma_2)}{\longrightarrow} & \vec{T}(X)(u,v) \end{array}$$
  commutes (up to ...).
- $(x,y) \stackrel{(c_x,H_y)}{\longrightarrow} (x,fy) \simeq (fx,fy) \stackrel{(H_x,c_{fy})}{\longrightarrow} (x,fy),\ H : id_X \to f$.
  Likewise for $H : g \to id_X$.

The component category $\vec{D}_\pi(X)/\simeq$ identifies pairs of points on the same “homotopy flow line” and (chains of) morphisms.

Concluding remarks

- Component categories contain the essential information given by (algebraic topological invariants of) path spaces
- Compression as 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)
- Dihomotopy equivalence: Definition uses automorphic homotopy flows to ensure homotopy equivalences

$$\vec{T}(f)(x,y) : \vec{T}(X)(x,y) \to \vec{T}(Y)(fx,fy) \text{ for all } x \preceq y.$$

- Much more theoretical and practical work remains to be done!