Invariants of directed spaces and persistence

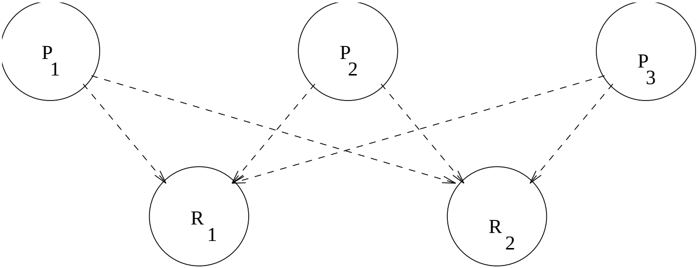
Martin Raussen

Department of mathematical sciences
Aalborg Universitet

MSRI-workshop, 5.10.2006

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.

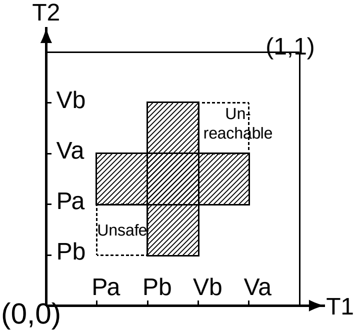
Semaphores!

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

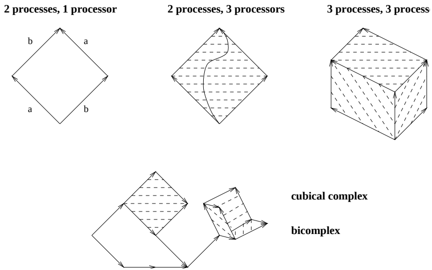


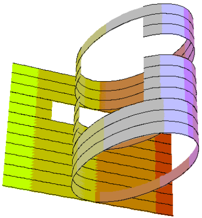
$P_1 : P_a P_b V_b V_a \qquad P_2 : P_b P_a V_a V_b$

Executions are directed paths avoiding a forbidden region (shaded).
Dipaths that are dihomotopic (homotopy through dipaths) correspond to equivalent executions.

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

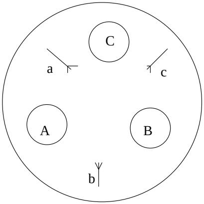
Vaughan Pratt, Rob van Glabbeek, Eric Goubault...



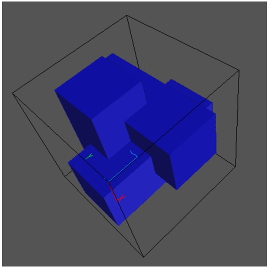


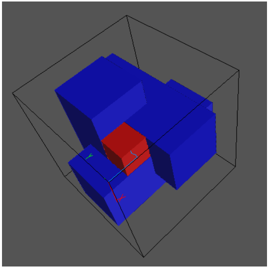
with preferred directions!

Higher dimensional automata
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.

Framework: d-spaces, M. Grandis (03)

X a topological space. $\vec{P}(X) \subseteq X^I$ a set of d-paths ("directed" paths $\leftrightarrow$ executions) satisfying

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

$(X, \vec{P}(X))$ is called a d-space.
Example: HDA with directed execution paths. Light cones (relativity)
A d-space is called saturated if furthermore

- $\varphi \in X^I, \alpha \in I^I$ nondecreasing and surjective (homeo), $\varphi \circ \alpha \in \vec{P}(X) \Rightarrow \varphi \in \vec{P}(X)$
  i.e., if $\vec{P}(X)$ is closed under reparametrization equivalence.

$\vec{P}(X)$ is in general not closed under reversal – $\alpha(t) = 1 - t$.

Dihomotopy, d-homotopy

Morphisms: d-maps $f : X \to Y$ satisfying

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

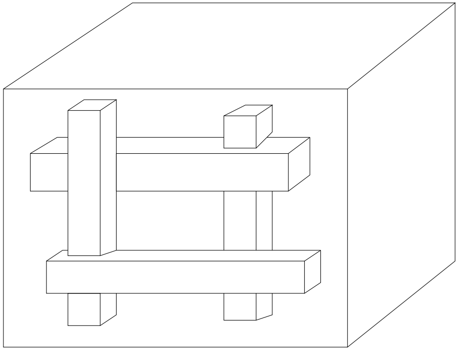
in particular: $\vec{P}(I) = \{\sigma \in I^I | \sigma \text{ nondecreasing}\}$
$\vec{I} = (I, \vec{P}(I)) \Rightarrow \vec{P}(X) =$d-maps from $\vec{I}$ to X.

- 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

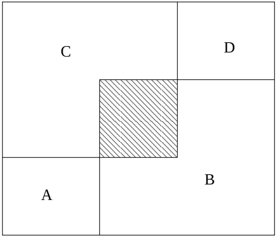


All dipaths from minimum to maximum are homotopic.
A dipath through the “hole” is not dihomotopic to a dipath on the boundary.

The fundamental category: favourite gadget so far

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

- objects points in X
- morphisms d- or dihomotopy classes of d-paths in X



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” only for loopfree categories

Outline

- Better bookkeeping: A zoo of categories and functors associated to a directed space – with a lot more animals than just the fundamental category
- Directed homotopy equivalences – more than just the obvious generalization of the classical notion
  Definition? Automorphic homotopy flows! Properties?
- Localization of categories with respect to invariant functors – ”components”, compressing information, making calculations feasible
- More general: “Bisimulation”(?) equivalence of categories with respect to a functor (over a fixed category)

Technique: Traces – and trace categories

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.

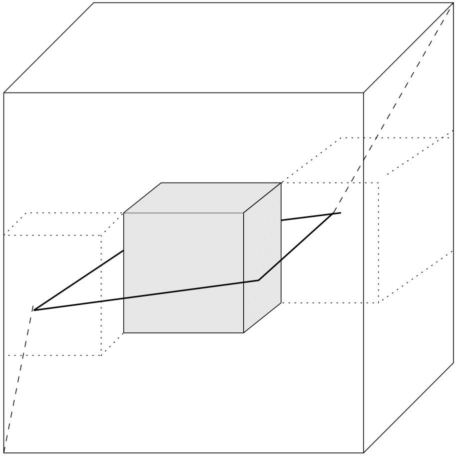
Sensitivity with respect to variations of end points
A persistence point of view

Questions: 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?
$\rightsquigarrow$ need a lot of bookkeeping!

Birth and death of dihomotopy
by example



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 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 $\vec{T}^X_\pi : \vec{D}_\pi(X) \to Ho-Top$.

Dihomology $\vec{H}_*$

For every d-space X, there are homology functors

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

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.

Factorization categories and higher homotopy

Indexing category = Factorization category $F\vec{T}(X)$ [Baues] with

- objects: $\sigma_{xy} \in \vec{T}(X)(x,y)$
- morphisms: $F\vec{T}(X)(\sigma_{xy}, \sigma'_{x'y'}) := \{(\varphi_{x'x}, \varphi_{yy'}) \in \vec{T}(X)(x',x) \times \vec{T}(X)(y,y') \mid \sigma'_{x'y'} = \varphi_{yy'} \circ \sigma_{xy} \circ \varphi_{x'x}\}$.

and functor $F\vec{T}^X : F\vec{T}(X) \to Top_*,\ \sigma_{xy} \mapsto (\vec{T}(X)(x,y), \sigma_{xy})$ – and induced pointed maps.
Compose with homotopy functors to get
$\vec{\pi}_{n+1}(X) : F\vec{T}(X) \to Grps$, resp. Ab,
$\vec{\pi}_{n+1}(X)(\sigma_{xy}) = \pi_n(\vec{T}(X)(x,y); \sigma_{xy})$
and maps induced by concatenation on the homotopy groups.

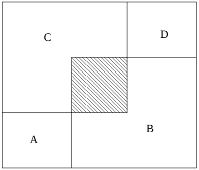
Dihomotopy equivalence – a naive definition

Definition

A d-map $f : X \to Y$ is a dihomotopy equivalence if there exists a d-map $g : Y \to X$ such that $g \circ f \simeq id_X$ and $f \circ g \simeq id_Y$.

But this does not imply an obvious property wanted for:
A dihomotopy equivalence $f : X \to Y$ should induce (ordinary) homotopy equivalences

$$\vec{T}(f) : \vec{T}(X)(x, y) \to \vec{T}(Y)(fx, fy)!$$



A map d-homotopic to the identity does not preserve homotopy types of trace spaces? Need to be more restrictive!

Homotopy flows

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

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

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

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....

Dihomotopy equivalences again

Definition

A d-map $f : X \to Y$ is called a future dihomotopy equivalence if there are maps $f_+ : X \to Y, g_+ : Y \to X$ with $f \to f_+$ and automorphic homotopy flows $id_X \to g_+ \circ f_+, id_Y \to f_+ \circ g_+$.
Property of dihomotopy class!

likewise: past dihomotopy equivalence $f_- \to f, g_- \to g$
dihomotopy equivalence = both future and past dhe (g_-, g_+ are then d-homotopic).

Theorem

A (future/past) dihomotopy equivalence $f : X \to Y$ induces homotopy equivalences

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

Moreover: (All sorts of) Dihomotopy equivalences are closed under composition.

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: $[\varphi] \circ [\psi] = [\varphi\psi]$.

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:

Congruences and component categories

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

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

Examples of component categories
Standard example

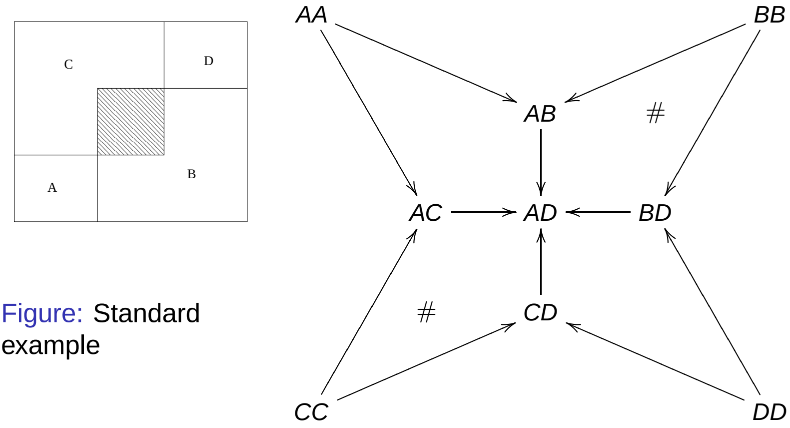
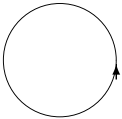


Figure: Standard example



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

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

It is essential to use an indexing category taking care of pairs (source, target).

A categorical generalization

Bisimulation(?) for categories over a category

Framework: Small categories over a fixed category $\mathcal{D}$.
Let $F:\mathcal{C}\to\mathcal{D}$ denote a functor (e.g., homology of trace spaces). Consider

- an equivalence relation $\equiv$ on the objects of $\mathcal{C}$ such that
- for every $x\equiv x'$, there is a subset $\emptyset \neq I(F(x),F(x'))\subset Iso(F(x),F(x'))$ such that $I(F(x),F(x'))=\varphi\circ I(F(x),F(x))$ for every $\varphi\in I(F(x),F(x'))$;
- for every $x\equiv x',\varphi\in I(F(x),F(x')),\sigma\in Mor_{\mathcal{C}}(x,y)$, there exists $y\equiv y',\varphi'\in I(F(y),F(y'))$ and $\sigma'\in Mor_{\mathcal{C}}(x',y')$ s.t.

$$\begin{array}{ccc} F(x) & \xrightarrow{F(\sigma)} & F(y) \\ \varphi\downarrow & & \downarrow\varphi' \\ F(x') & \xrightarrow{F(\sigma')} & F(y') \end{array}$$

commutes. Likewise

$$\begin{array}{ccc} F(x) & \xrightarrow{F(\tau')} & F(y) \\ \psi'\downarrow & & \downarrow\psi \\ F(x') & \xrightarrow{F(\tau)} & F(y') \end{array}$$

F- bisimulation equivalent categories

This relation generates a generalized congruence on $\mathcal{C}$ and a quotient functor $T:\mathcal{C}\to\mathcal{C}/_{\equiv}$. $\mathcal{C}$ and $\mathcal{C}/_{\equiv}$ are considered as equivalent categories over $F:\mathcal{C}\to\mathcal{D}$. Consider the transitive symmetric closure of this relation coming from zig-zags

$$\mathcal{C}_1\to\mathcal{C}_1/_{\equiv_1}\simeq\mathcal{C}_2/_{\equiv_2}\leftarrow\mathcal{C}_2\to\cdots$$

Gives rise to $F:\mathcal{C}\to\mathcal{D}$-(bisimulation) equivalent categories. In the (previous) examples, the equivalence relation on the objects was generated by the automorphic past and future homotopy flows. These do not always identify ”enough” objects. Example: $X=\vec{I}^2\setminus\vec{J}^2$. Then $\vec{H}_2(X)=H_1$ of trace spaces is trivial between arbitrary pairs of points, but automorphic flows cannot identify all points with each other.
This is instead achieved by the bisimulation construction above – trivial component category with respect to $\vec{H}_2$!