Homotopy theory and concurrency

Rick Jardine

Dagstuhl Seminar 16282

July 13, 2016
Basic idea (V. Pratt, 1991): represent the simultaneous execution of processors $a$ and $b$ as a picture (2-cell)

Simultaneous action of multiple processors is represented by higher dimensional cubes.

Restrictions on the system arising from shared resources are represented by removing cubical cells of varying dimensions, so one is left with a cubical subcomplex $K \subset \Box^N$ of an $N$-cell, where $N$ is the number of processors.
Higher dimensional automata are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$.

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$. 

*Higher dimensional automata* are cubical subcomplexes $K \subset \square^N$ of the “standard” $N$-cell $\square^N$. 

States are objects (vertices), and “execution paths” are morphisms of the “path category” $P(K)$.

Execution paths are equivalence classes of combinatorial paths through the complex. Executions paths between states $x$ and $y$ are the morphisms $P(K)(x, y)$.

**Basic problem**: Compute $P(K)(x, y)$.
Example: the Swiss flag

\[ B = (a_2, b_2) \]

\[ A = (a_1, b_1) \]
The \textbf{n-cell} $\Box^n$ is the poset
\[ \Box^n = \mathcal{P}(n), \]
the set of subsets of the totally ordered set $n = \{1, 2, \ldots, n\}$. There is a unique poset isomorphism
\[ \phi : \mathcal{P}(n) \xrightarrow{\cong} 1 \times n, \]
where 1 is the 2-element poset $0 \leq 1$. Here,
\[ A \leftrightarrow (\epsilon_1, \ldots, \epsilon_n) \]
where $\epsilon_i = 1$ if and only if $i \in A$. We use the ordering of $n$ to specify the poset isomorphism $\phi$. 

Rick Jardine
Homotopy theory and concurrency
The box category

Suppose that $A \subset B \subset n$. The interval $[A, B] \subset \mathcal{P}(n)$ is defined by

$$[A, B] = \{ C \mid A \subset C \subset B \}.$$

There are canonical poset maps

$$\mathcal{P}(m) \cong \mathcal{P}(B - A) \xrightarrow{\cong} [A, B] \subset \mathcal{P}(n).$$

where $m = |B - A|$. These compositions are the coface maps $d : \square^m \subset \square^n$.

There are also codegeneracy maps $s : \square^n \rightarrow \square^r$, determined by subsets $A \subset n$, where $|A| = r$, and such that $s(B) = B \cap A$.

The cofaces and codegeneracies are the generators for the box category $\square$ consisting of the posets $\square^n$, $n \geq 0$, subject to the cosimplicial identities.
A **cubical set** is a functor $X : \Box^{op} \to \textbf{Set}$. 

$\Box^n \mapsto X_n$, and $X_n$ is the set of $n$-**cells** of $X$. 

The collection of all such functors and natural transformations between them is the category $c\textbf{Set}$ of cubical sets. 

**Examples**

1) The **standard** $n$-**cell** $\Box^n$ is the functor $\text{hom}(\ , \Box^n)$ represented by $\Box^n = P(n)$ on the box category $\Box$. 

The $n$-cells of a cubical set $X$ can be identified with maps $\sigma : \Box^n \to X$. 

2) Deleting the top cell from $\Box^n$ gives the **boundary** $\partial \Box^n$. 

There are 2 maximal faces of $\partial \Box^n$ for each $i \in n$: $[\{i\}, n]$, $[\emptyset, \{1, \ldots, \hat{i}, \ldots, n\}]$. 

3) The **cubical horn** $\nabla^n_{(i, \epsilon)}$ is defined by deleting a top dim. face from $\partial \Box^n$. 

---

Rick Jardine

Homotopy theory and concurrency
A finite cubical complex is a subcomplex $K \subset \square^n$. It is completely determined by cells

$$\square^r \subset K \subset \square^n$$

where the composites are cofaces.

Equivalently, $K$ is a set of intervals $[A, B] \subset \mathcal{P}(n)$ which is closed under taking subintervals.

A cell (interval) is maximal if $r = |B - A|$ is maximal wrt these constraints.

Finite cubical complexes are higher dimensional automata.
There is a **triangulation functor**

\[ | \cdot | : \text{cSet} \to \text{sSet}, \]

with \[|\Box^n| := B(\mathcal{P}(n)) \cong B(1 \times n) \cong (\Delta^1)^n.\]

\(B(C)\) is the **nerve** of a category \(C\): \(B(C)_n\) is the set

\[a_0 \to a_1 \to \cdots \to a_n\]

The **triangulation** \(|K|\) is defined by

\[|K| = \lim_{\Box^n \to K} |\Box^n| = \lim_{\Box^n \to K} B(1 \times n) \cong \lim_{[A,B] \in K} B([A,B]).\]

NB: \(K, L\) simplicial complexes, then \(K \times L\) is a simplicial complex, by finding a (compatible) total ordering on the vertices \(K_0 \times L_0\).
Examples

1) $|\square^2| = B(1^2) = B(P(2))$:

\[
\begin{align*}
(0, 1) & \rightarrow (1, 1) & \{2\} & \rightarrow \{1, 2\} \\
\uparrow & \quad \quad \quad \uparrow & \quad \quad \quad \uparrow & \quad \quad \quad \uparrow \\
(0, 0) & \rightarrow (1, 0) & \emptyset & \rightarrow \{1\}
\end{align*}
\]

2) $|\square^1 \times \square^1|$ has 1-skeleton

\[
\begin{align*}
(0, 1) & \rightarrow (1, 1) \\
\uparrow & \quad \quad \quad \uparrow \\
(0, 0) & \rightarrow (1, 0)
\end{align*}
\]

with non-degenerate 2-cells $P(2) \rightarrow P(1) \times P(1)$ given by the canonical isomorphism $P(2) \cong P(1) \times P(1)$ and its twist. Thus

$$|\square^1 \times \square^1| \simeq S^2 \vee S^1$$
Say that a monomorphism of cubical sets is a \textbf{cofibration}. A map $X \to Y$ of cubical sets is a \textbf{weak equivalence} if the induced map $f_* : |X| \to |Y|$ is a weak equivalence of simplicial sets.

\textbf{Fibrations} of cubical sets are defined by a right lifting property with respect to all trivial cofibrations.

**Theorem 1.**

1) With these definitions the category $c\text{Set}$ has the structure of a proper, closed (cubical) model category.

2) The adjoint functors

$$| \cdot | : c\text{Set} \leftrightarrow s\text{Set} : S$$

define a Quillen equivalence.

The right adjoint $S$ is the singular functor.
The nerve functor $B : \text{cat} \rightarrow s\text{Set}$ has a left adjoint

$$P : s\text{Set} \rightarrow \text{cat},$$

called the **path category** functor. Most often see $\tau_1(X) = P(X)$. $P(X)$ is the category generated by the 1-skeleton $\text{sk}_1(X)$ (a graph), subject to the relations:

1) $s_0(x)$ is the identity morphism for all vertices $x \in X_0$,  
2) the triangle

$$\begin{array}{ccc}
X_0 & \xrightarrow{d_2(\sigma)} & X_1 \\
& \searrow^{d_1(\sigma)} & \\
& & X_2 \\
& \downarrow^{d_0(\sigma)} & \\
&&
\end{array}$$

commutes for all 2-simplices $\sigma : \Delta^2 \rightarrow X$. 

Rick Jardine

Homotopy theory and concurrency
Suppose that $K \subset \Box^n$ is an HDA, with states (vertices) $x, y$. Then

$$P(\lvert K \rvert)(x, y)$$

is the set of **execution paths** from $x$ to $y$.

$P(K) := P(\lvert K \rvert)$ is the **path category** of the cubical complex $K$.

$P(K)$ can be defined directly for $K$: it is generated by the graph $sk_1(K)$, subject to the relations given by $s_0(x) = 1_x$ for vertices $x$, and by forcing the commutativity of

![Diagram](attachment:diagram.png)

for each 2-cell $\sigma : \Box^2 \subset K$ of $K$. 

Rick Jardine

Homotopy theory and concurrency
Lemma 2.

\[ \text{sk}_2(X) \subset X \text{ induces } P(\text{sk}_2(X)) \cong P(X) \text{ for simplicial sets (or cubical complexes) } X. \]

Lemma 3.

\[ \epsilon : P(BC) \rightarrow C \text{ is an isomorphism for all small categories } C. \]

Lemma 4.

There is an isomorphism \( G(P(X)) \cong \pi(X) \) for all simplicial sets \( X \).

\( G(P(X)) \) is the free groupoid on the category \( P(X) \).
The path 2-category

\( L = \) finite simplicial complex: \( P(L) \) is the path component category of a 2-category \( P_2(L) \).

\( P_2(L) \) consists of categories \( P_2(L)(x, y) \), \( x, y \in L \).
The objects (1-cells) are paths of non-deg. 1-simplices

\[
x = x_0 \to x_1 \to \cdots \to x_n = y
\]
of \( L \). The morphisms of \( P_2(L)(x, y) \) are composites of the pictures

where the displayed triangle bounds a non-deg. 2-simplex.

Compositions are functors

\[
P_2(L)(x, y) \times P_2(L)(y, z) \to P_2(L)(x, z)
\]
defined by concatenation of paths.
Theorem 5.

Suppose that $L$ is a finite simplicial complex. Then there is an isomorphism

$$\pi_0 P_2(L) \cong P(L).$$

$\pi_0 P_2(L)$ is the path component category of the 2-category $P_2(L)$. Its objects are the vertices of $L$, and

$$\pi_0 P_2(L)(x, y) = \pi_0 (BP_2(L)(x, y)).$$

Slogan: $P_2(L)$ is a “resolution” of the path category $P(L)$. 
The algorithm

Here’s an algorithm for computing $P(L)$ for $L \subset \Delta^N$:

1) Find the 2-skeleton $sk_2(L)$ of $L$ (vertices, 1-simplices, 2-simplices).

2) Find all paths (strings of 1-simplices)

$$\omega : v_0 \stackrel{\sigma_1}{\rightarrow} v_1 \stackrel{\sigma_2}{\rightarrow} \ldots \stackrel{\sigma_k}{\rightarrow} v_k$$

in $L$.

3) Find all morphisms in the category $P_2(L)(v, w)$ for all vertices $v < w$ in $L$ (ordering in $\Delta^N$).

4) Find the path components of all $P_2(L)(v, w)$, by approximating path components by full connected subcategories, starting with a fixed path $\omega$.

Code: Graham Denham (Macaulay 2), Mike Misamore (C).
Let $L \subset \Delta^{40}$ be the subcomplex

```
0 \rightarrow 2 \rightarrow 4 \rightarrow \ldots \rightarrow 38 \rightarrow 40
```

This is 20 copies of the complex $\partial \Delta^2$ glued together. There are $2^{20}$ morphisms in $P(L)(0, 40)$.

The listing of morphisms of $P(L)$ consumes 2 GB of disk.

**Moral:** The size of the path category $P(L)$ can grow exponentially with $L$. 

**Example: the necklace**
Suppose that $L \subseteq K \subseteq \Delta^N$.

$L$ is a **full subcomplex** of $K$ if the following hold:

1) $L$ is path-closed in $K$, in the sense that, if there is a path

$$v = v_0 \to v_1 \to \cdots \to v_n = v'$$

in $K$ between vertices $v, v'$ of $L$, then all $v_i \in L$,

2) if all the vertices of a simplex $\sigma \in K$ are in $L$ then the simplex $\sigma$ is in $L$.

**Lemma 6.**

Suppose that $L$ is a full subcomplex of $K$. Then the functor $P(L) \to P(K)$ is fully faithful.

ie. if $x, y \in L$ then $P(L)(x, y) = P(K)(x, y)$. 
Examples

- \( \partial \Delta^2 \subset \Lambda_0^3 \) and \( \partial \Delta^2 \subset \Lambda_3^3 \) are full subcomplexes.

- Suppose that \( i \leq j \) in \( \mathbb{N} \). \( K[i, j] \) is the subcomplex of \( K \) such that \( \sigma \in K[i, j] \) if and only if all vertices of \( \sigma \) are in the interval \([i, j]\) of vertices \( v \) such that \( i \leq v \leq j \). \( K[i, j] \) is a full subcomplex of \( K \).

- Suppose that \( v \leq w \) are vertices of \( K \). Let \( K(v, w) \) be the subcomplex of \( K \) consisting of simplices whose vertices appear on a path from \( v \) to \( w \). \( K(v, w) \) is a full subcomplex of \( K \).

Construct \( K(v, w) \) from \( K[v, w] \) by deleting sources and sinks.

A vertex \( v \) is a source of \( K \) if there are no 1-simplices \( u \to v \) in \( K \). (0 is a source of \( \Lambda_0^3 \))

A vertex \( z \) is a sink if there are no 1-simplices \( z \to w \) in \( K \). (3 is a sink of \( \Lambda_3^3 \))
Suppose that $K \subset \square^n$ is a cubical complex. Say that a vertex $x$ is a **corner** of $K$ if it belongs to only one maximal cell.

**Lemma 7 (Misamore).**

Suppose that $x$ is a corner of $K$, and let $K_x$ be the subcomplex of cells which do not have $x$ as a vertex. Then the induced functor

$$P(K_x) \to P(K)$$

is fully faithful.
1) The cubical horn \((0, 1) \rightarrow (1, 1)\) has a sink but no corners.

\[
\begin{array}{c}
(0, 0) \\
\uparrow \\
(1, 0)
\end{array}
\]

2) The Swiss flag has 6 corners, 1 sink, 1 source.

\[
\begin{array}{c}
\bullet \\
\downarrow \\
\star \\
\downarrow \\
\bullet
\end{array}
\]

\[
\begin{array}{c}
\bullet \\
\downarrow \\
\star \\
\downarrow \\
\bullet
\end{array}
\]

\[
\begin{array}{c}
\bullet \\
\downarrow \\
\star \\
\downarrow \\
\bullet
\end{array}
\]
1) The algorithm depends on having an entire HDA in storage, on a computer system that is powerful enough to analyze it.

2) We want local to global methods to study large (aka. “infinite”) models with patching techniques.

3) Nobody has any idea of what higher homotopy invariants should mean for higher dimensional automata, or even what the appropriate homotopy theory should be.

One suggestion: Joyal’s theory of quasi-categories.

$K \mapsto P(K)$ is a quasi-category invariant, but beware:

If $K \to L$ is a quasi-category weak equivalence then $P(K) \to P(L)$ is an equivalence of categories.
The size parameter

The assignment $A \mapsto |A|$ defines a poset map $\mathcal{P}(\mathbb{N}) \to \mathbb{N}$.

If $K \subset \square^N$ then the composite $t$

$$|K| \to B(\mathcal{P}(\mathbb{N})) \to B\mathbb{N}$$

$|A|$ is the number of steps the system took to reach state $A$, by whatever path. $t$ is the “size parameter”.

Suppose that $m < n$ are natural numbers. $K(m, n)$ is the subcomplex of $K$ whose cells have vertices $A$ such that $m \leq |A| \leq n$.

**Fact**: $K(m, n)$ is a full subcomplex of $K$, so $P(K(m, n)) \to P(K)$ is fully faithful by Lemma 6.

The complexes $K(m, n)$ define a coarse “topology” on $K$. 

Rick Jardine  
Homotopy theory and concurrency
Suppose that the poset map $\alpha : \mathcal{P}(n) \to \mathcal{P}(N)$ is a monomorphism that preserves meets and joins.

$[A, B]$ in $\mathcal{P}(n)$ determines an interval $[\alpha(A), \alpha(B)]$ in $\mathcal{P}(N)$, and $\alpha$ restricts to a poset monomorphism $\alpha : [A, B] \to [\alpha(A), \alpha(B)]$ (not a coface).

$\alpha$ preserves inclusion and intersections of intervals.

The subcomplex $K_\alpha \subset \mathcal{P}(N)$ is generated by the intervals $[\alpha(A), \alpha(B)]$, and there is a diagram of simp. set maps

$$
\begin{array}{ccc}
|K| & \xrightarrow{\alpha_*} & |K_\alpha| \\
\downarrow & & \downarrow \\
BP(n) & \xrightarrow{\alpha} & BP(N)
\end{array}
$$

$K_\alpha$ is a refinement of $K$.

**Lemma 8.**

*The functor $\alpha_* : P(K) \to P(K_\alpha)$ is fully faithful.*
$K \subset \square^N$, finite cubical complex.

$K_0 = A \sqcup B$, with $t(v) < t(w)$ for all $v \in A$ and $w \in B$. Let $A$ and $B$ be the corr. full subcomplexes of $K$. Suppose that $P(A)$ and $P(B)$ are computed.

The **frontier subcomplex** $L$ (of $|K|$) is generated by 1-cells and 2-cells which have vertices in both $A$ and $B$.

Example: if $\sigma : x \to y$ has $x \in A$ and $y \in B$, then $\sigma \in L$.

Composition with $\sigma$ defines a map

$$\sigma_* : P(A)(u, x) \times P(B)(y, v) \to P(K)(u, v).$$
An example

Black: $t(F) \leq 7$, Red: $t(F) \geq 8$. Total $t$: 14.
Suppose that $\omega : \Delta^1 \times \Delta^1 \rightarrow L$ is defined by 2-simplices $\omega_0$ and $\omega_1$ such that $d_1\omega_0 = d_1\omega_1$ and $d_2(\omega_0) \in A$ and $d_0(\omega_1) \in B$. One of the 2-simplices $\omega_0$ or $\omega_1$ could be degenerate.

There are induced maps

$$\omega_0 : P(A)(u, \sigma(0, 0)) \times P(B)(\sigma(1, 1), v) \rightarrow P(A)(u, \sigma(0, 0)) \times P(B)(\sigma(1, 0), v)$$

and

$$\omega_1 : P(A)(u, \sigma(0, 0)) \times P(B)(\sigma(1, 1), v) \rightarrow P(A)(u, \sigma(0, 1)) \times P(B)(\sigma(1, 1), v)$$
\[ \omega_0, \omega_1 \text{ define the parallel pair of arrows of a coequalizer diagram} \]

\[
\bigsqcup_{\omega: \Delta^1 \times \Delta^1 \to L} P(A)(u, \sigma(0, 0)) \times P(B)(\sigma(1, 1), v) \Rightarrow \bigsqcup_{\sigma:x \to y \in L} P(A)(u, x) \times P(B)(y, v) \to P(K)(u, v).
\]

ie. \( P(K)(u, v) \) is a set of equivalence classes for a relation on the set

\[
\bigsqcup_{\sigma:x \to y \in L} P(A)(u, x) \times P(B)(y, v),
\]

defined in a very specific way.

**Conclusion:** \( P(K) \) is computable from \( P(A) \) and \( P(B) \), together with incidence relations defined by the frontier subcomplex \( L \).
References

J. F. Jardine.
Path categories and resolutions.

J. F. Jardine.
Path categories and quasi-categories.

Michael D. Misamore.
Computing path categories of finite directed cubical complexes.