# Homotopy in concurrent processes

Stefan Sokołowski[1]

Sopot, 20 April 1998

[1]Polish Academy of Sciences, Institute of Computer Science, Gdańsk Division, ul. Abrahama 18, 81-825 Sopot, Poland, `http://www.ipipan.gda.pl/~stefan/`. Report available electronically at `ftp://ftp.ipipan.gda.pl/stefan/reports/58-homotopy.ps.gz`.

**Abstract**

## Homotopy in concurrent processes

In theories of job scheduling and of distributed computing, there have been many attempts to introduce tools originating from algebraic and combinatorial topology, such as homotopy groups. Informally, the fundamental (or first homotopy) group gives an account of the nature of "holes" in a topological space. In the realm of processes, such holes may correspond to forbidden configurations; e.g. where more than one process is within the same critical region.

However, many topological properties, technically necessary for the construction of the fundamental group, have no counterparts, or only artificial ones, in concurrent processes. The paths corresponding to process executions are not cyclic, because time only flows forwards, and they cannot be as naturally composed as looping paths in topological spaces. The fundamental "group" lacks therefore its group operations.

This paper puts forward a simple remedy for the shortcomings: if you cannot find a useful group operation on your homotopy classes, settle for a less requiring structure, e.g. *homotopy cpo*-s (explained in this paper). As will be shown, the transition from vectors of processes to their homotopy cpo-s is functorial, preserves information about the "holes" and abstracts from inessential details. This renders the approach a potentially useful tool for investigating admissible runs of concurrent processes.

**Streszczenie**

## Homotopie w badaniu procesów współbieżnych

W teoriach szeregowania zadań oraz obliczeń rozproszonych wielokrotnie próbowano stosować narzędzia pochodzące z topologii algebraicznej i kombinatorycznej, takie jak grupy homotopii. Mówiąc nieformalnie grupa podstawowa (czyli pierwsza grupa homotopii) oddaje naturę ,,dziur" w przestrzeni topologicznej. W kontekście procesów takie dziury mogą odpowiadać zabronionym konfiguracjom; np. takim, w których więcej niż jeden proces znajduje się wewnątrz regionu krytycznego.

Jednak szereg własności topologicznych niezbędnych technicznie dla konstrukcji grupy podstawowej nie ma naturalnych odpowiedników w procesach współbieżnych. Ścieżki odpowiadające wykonaniom procesów nie są cykliczne, bo czas płynie tylko w jedną stronę, więc nie mogą być składane w taki sam sposób jak zamknięte ścieżki w przestrzeniach topologicznych. Takiej ,,grupie" podstawowej brakuje więc operacji grupowych.

Niniejsza praca przedstawia prostą drogę wyjścia z tej trudności. Jeśli nie da się znaleźć użytecznej operacji grupowej, należy się zadowolić strukturą mniej wymagającą niż grupa; np. *homotopijnym cpo* (wyjaśnione dokładniej w tej pracy). Pokazujemy, że przejście od wektorów procesów do ich homotopijnych cpo jest funktorialne, zachowuje informację o ,,dziurach" i abstrahuje od nieistotnych szczegółów. W wyniku tego przedstawione podejście może się stać użytecznym narzędziem do badania dopuszczalnych wykonań procesów współbieżnych.

# Contents

# Chapter 0

# Introduction

In theories of job scheduling and of distributed computing, there have been many attempts to introduce tools originating from algebraic and combinatorial topology. To mention just a few, Herlihy and Shavit [8] have reduced task solvability problem to existence of a certain mapping between simplicial complexes representing the task's input and the task's output. Herlihy and Rajsbaum [6] have used chain homotopies and acyclic carrier theorem to derive some lower bounds and impossibility results for distributed computing. Hoest and Shavit [9] have studied asynchronous complexity by the way of chromatic subdivisions of complexes. Tutorial papers in this area are [7] by Herlihy and Rajsbaum, and [5] by Herlihy. In another line of research, Gunawardena [4] has given an intuitively appealing proof, based on investigations of homotopy properties of certain simple topological spaces, that "two phase locking is safe" — an important theorem on database consistency. Probably the most deeply "algebraicized" of all is the contribution by Goubault and Jensen [3] where the geometric view of concurrency (cf. Pratt [11] and van Glabbeek [2]) is provided with a complete construction of homology groups in arbitrary dimensions.

Informally, the fundamental (or first homotopy) group gives an account of the nature of "holes" in a topological space. In the realm of processes, such holes may correspond to forbidden configurations; e.g. where more than one process is within the same critical region.

However, many topological properties, technically necessary for the construction of the fundamental group, have no counterparts, or only artificial ones, in concurrent processes. The paths corresponding to process executions are not cyclic, because time only flows forwards, and they cannot be as naturally composed as looping paths in topological spaces. The fundamental "group" lacks therefore its group operations. The situation is better in homologies, since in this case the group structure is derived from the combinatorial properties of respective simplicial complexes rather than from process executions. Homology groups for task solvability problems are therefore defined. However, in most considerations important for computer science, either the conclusions are drawn from the combinatorial arguments that precede the construction of these groups, or the homologies are used only marginally, the only issue of importance being whether or not they are all trivial, i.e. whether or not the corresponding complex is acyclic. Goubault and Jensen [3] seem to be an exception but even they deal with homologies with "modulo 2" coefficients. This corresponds to simplexes without orientation and reflects the fact that time does not flow around. To sum up this critical paragraph: my feeling is that the payout from homotopy and homology considerations for computer scientists is more limited than the payout from such considerations for topologists.

This paper deals with homotopy rather than homology. It puts forward a simple remedy for the shortcomings: if you cannot find a useful group operation on your homotopy classes, settle for a less requiring structure, e.g. *homotopy cpo*-s (explained in this paper). As will be shown, the transition from vectors of processes to their homotopy cpo-s is functorial, preserves information about the "holes" and abstracts from inessential details. This renders the approach a potentially useful tool for investigating admissible runs of concurrent processes.

The functoriality referred to above seems a novelty in computer science applications of algebraic topology. People have assigned algebraic structures to processes but, to the best of my knowledge, nobody has ever translated maps between processes to homomorphisms of respective structures. That non-functorial approach allows for a classification of processes the way Euler numbers classify manifolds, but it comes short of a subtler analysis of dependencies between processes. As will be shown, the functoriality of fundamental cpo-s may be applied to the problem of implementability of processes using other processes.

**Acknowledgements.** Former versions of this paper have been presented at the **LIRA'97** conference in Novi Sad, Yugoslavia, and at seminars in the **Institute of Computer Science** in Warsaw, Poland, and in the **University of Gdańsk**, Poland. They have also been discussed with **Marek Bednarczyk** and with **Andrzej Borzyszkowski**. All these events have resulted in substantial improvements of the contents.

# Chapter 1

# Preliminary notions

Auxiliary terminology and notations are introduced here.

## 1.1 Set theory

Given sets $X$ and $Y$, a function $f : X \to Y$, a subset $A \subseteq X$ and a subset $B \subseteq Y$, denote the *image of* $A$ *via* $f$ by $f^{\to}A$ and the *coimage of* $B$ *via* $f$ by $f^{\leftarrow}B$:

$$f^{\to}A \stackrel{\text{def}}{=} \left\{ fx \in Y \,\middle|\, x \in A \right\} \quad \text{and} \quad f^{\leftarrow}B \stackrel{\text{def}}{=} \left\{ x \in X \,\middle|\, fx \in B \right\}$$

The *generalized Cartesian product* of a family $\{X_i\}_{i \in I}$ of sets is denoted by $\prod_{i \in I} X_i$. It consists of functions $f$ over the index set $I$ such that $f\,i \in X_i$ for all $i \in I$. In the special case, when $I = \{0, 1\}$, the product is denoted by $X_0 \times X_1$ and its elements by $\langle a_0, a_1 \rangle$ with $a_0 \in X_0$ and $a_1 \in X_1$.

The set of all subsets of a certain $X$ will be denoted by $\mathcal{P}(X)$. The set of all finite subsets of $X$ will be denoted by $\mathcal{P}_{\mathit{fin}}(X)$.

## 1.2 Chain-complete partial orders

*Cpo*, i.e. *chain-complete partial order*, is any triple $\mathcal{X} = \langle X, 0, \sqsubseteq \rangle$ where

- $\langle X, \sqsubseteq \rangle$ is a partial order closed upon least upper bounds (*lub*s) of countable ascending *chains*:

$$x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots \;\Rightarrow\; \text{there exists} \bigsqcup_{n \in Nat} x_n$$

- $0 \in X$ is the bottom element: $\forall_{x \in X}\, 0 \sqsubseteq x$.

By $\max \mathcal{X}$ (or $\max X$ ) will be denoted the set of maximal elements of $\mathcal{X}$:

$$\max \mathcal{X} \stackrel{\text{def}}{=} \left\{ x \in X \,\middle|\, \forall_{y \in X}\, x \sqsubseteq y \Rightarrow x = y \right\}$$

*Closure* of a subset $Y \subseteq X$ in a certain cpo $\mathcal{X} = \langle X, 0, \sqsubseteq \rangle$ is the set of lubs of its chains:

$$\mathrm{cl}_{\mathcal{X}}\, Y \stackrel{\text{def}}{=} \left\{ \textstyle\bigsqcup_{n \in Nat} y_n \,\middle|\, (\forall_{n \in Nat}\, y_n \in Y) \;\&\; y_0 \sqsubseteq y_1 \sqsubseteq y_2 \sqsubseteq \ldots \right\}$$

Subset $Y$ is *dense in* $\mathcal{X}$ whenever $\mathrm{cl}_{\mathcal{X}}\, Y = X$. *Downcone* of a subset $Y \subseteq X$ in a certain cpo $\mathcal{X} = \langle X, 0, \sqsubseteq \rangle$ is the set of elements smaller than some elements of $Y$:

$$\downarrow_{\mathcal{X}} Y \stackrel{\text{def}}{=} \left\{ x \in X \,\middle|\, \exists_{y \in Y}\, x \sqsubseteq y \right\}$$

The $\mathcal{X}$ in $\downarrow_{\mathcal{X}} Y$ will be omitted whenever there arises no doubt as to the cpo.

A subset $Y \subseteq X$ is *downward-closed* in $\mathcal{X}$ if

$$\forall_{y \in Y} \forall_{x \in X}\, x \sqsubseteq y \Rightarrow x \in Y$$

**Proposition 1**
*Downcone $\downarrow_{\mathcal{X}} Y$ is downward-closed.*

Assume $\mathcal{X} = \langle X, 0, \sqsubseteq \rangle$ and $\mathcal{X}' = \langle X', 0', \sqsubseteq' \rangle$ are cpo-s. A function $f : X \to X'$ is *continuous* iff it is monotone and preserves the bottom and the limits:

- $x_1 \sqsubseteq x_2 \Rightarrow f x_1 \sqsubseteq' f x_2$,
- $f 0 = 0'$,
- $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots \;\Rightarrow\; f(\bigsqcup_{n \in Nat} x_n) = \bigsqcup'_{n \in Nat} f x_n$.

**Proposition 2** (cpo-s form a category)
*Chain complete partial orders and continuous mappings form a category.*

The category referred to in Prop. 2 is denoted by $\underline{\underline{Cpo}}$.

Cpo $\mathcal{X}$ is called *straight* iff any two elements, which have a common upper bound, are comparable:

$$\forall_{x,y,z \in X}\, x \sqsubseteq z \;\&\; y \sqsubseteq z \Rightarrow x \sqsubseteq y \vee y \sqsubseteq x$$

**Theorem 1** (on continuous extensions)
*Assume $\mathcal{X} = \langle X, 0, \sqsubseteq \rangle$ and $\mathcal{Y} = \langle Y, 0, \sqsubseteq \rangle$ are cpo-s, $\mathcal{X}$ is straight and a subset $X_0$ is dense in $X$. Then for any monotone function $f : X_0 \to Y$ there exists a unique continuous extension $\bar{f} : X \to Y$ (such that $\bar{f}|_{X_0} = f$ ).*

## 1.3 Words

For any set $A$ (called *alphabet*), $A^*$ denotes the set of finite sequences over $A$ called finite $A$-*words*, $A^\omega$ denotes the set of countably infinite $A$-words and $A^s \stackrel{\text{def}}{=} A^* \cup A^\omega$ denotes the set of finite and infinite $A$-words. For any set $A$, $\mathcal{S}_A = \langle A^s, \varepsilon, \sqsubseteq \rangle$, where $\varepsilon$ is the empty word and $\sqsubseteq$ is the prefix order, denotes the cpo of $A$-words.

**Proposition 3** (prefix order does not fork)
*For any finite word $w \in A^*$ and any words $v_1, v_2, u \in A^s$, if $w \frown v_1 \sqsubseteq u$ and $w \frown v_2 \sqsubseteq u$ then either $v_1 \sqsubseteq v_2$ or $v_2 \sqsubseteq v_1$.*

**Proposition 4**
*$\mathcal{S}_A$ is a straight cpo and $A^*$ is dense in $\mathcal{S}_A$.*

For any finite word $w = a_0 \dots a_{n-1} \in A^*$, let $\sharp w \stackrel{\text{def}}{=} n$ denote the *length of* $w$. For any infinite word $w = a_0 a_1 \dots \in A^\omega$, let $\sharp w \stackrel{\text{def}}{=} \omega$, a special element considered bigger than any natural number. For any finite word $w = a_0 \dots a_{n-1} \in A^*$ and any natural $k < n$, let $w[k]$ denote $a_k$, the $k$-th letter in $w$, with $w[k]$ undefined for $k \geq n$. Similarly, for any infinite word $w = a_0 a_1 \dots \in A^\omega$ and any natural $k$, let $w[k]$ denote $a_k$, the $k$-th letter in $w$. Define the operation $\frown$ of *word concatenation* by

$$(w \frown v)[k] \stackrel{\text{def}}{=} \begin{cases} w[k] & \text{if } k < \sharp w \\ v[k - \sharp w] & \text{if } k \geq \sharp w \end{cases}$$

Thus, $w \frown v = w$ whenever $\sharp w = \omega$.

Given alphabets $A$ and $B$, any function $f : A^* \to B^*$ such that

- $f\varepsilon = \varepsilon$ and
- $f(w \frown v) = f(w) \frown f(v)$ for $w, v \in A^*$

is called a *word-homomorphism*. Word-homomorphisms are often given as extensions of single letter mappings; notably,

**Proposition 5** (on extension of function defined on an alphabet)
*An arbitrary function $f : A \to B^*$ assigning words to single letters has a unique extension to a word-homomorphism $f : A^* \to B^*$.*

The word-homomorphism extending a certain $f : A \to B^*$ in compliance with Prop. 5 will also be denoted by $f$.

**Proposition 6** (on continuous extension of word-homomorphism)
*An arbitrary word-homomorphism $f : A^* \to B^*$ has a unique continuous extension*

$$\sigma f : A^s \to B^s$$

*(such that $(\sigma f)|_{A^*} = f$ ).*

Proof of Proposition 6:

By virtue of Prop. 4 and Thm. 1, it only has to be shown that $f$ is monotone. But this follows directly from the definition of the prefix order in $\mathcal{S}_A$.
□

**Proposition 7**
*For any word-homomorphisms $A^* \xrightarrow{f} B^* \xrightarrow{g} C^*$,*

$$\sigma(g \circ f) = \sigma g \circ \sigma f$$

The extension $\sigma f : A^s \to B^s$ generated by a word-homomorphism discussed above may fail to satisfy the homomorphic law

$$\sigma f(w \frown v) = \sigma f w \frown \sigma f v$$

For a counterexample, consider the word-homomorphism $f : A^* \to A^*$ for $A \stackrel{\text{def}}{=} \{a, b\}$ defined by

$$\begin{cases} f\,a \stackrel{\text{def}}{=} a \\ f\,b \stackrel{\text{def}}{=} \varepsilon \end{cases}$$

(erasing letter $b$) and let

$$w \stackrel{\text{def}}{=} bbb\ldots \quad \text{and} \quad v \stackrel{\text{def}}{=} a$$

Then $w\frown v = w$ and $\sigma f(w\frown v) = \sigma f w = \varepsilon$; on the other hand,

$$\sigma f w \frown \sigma f v = \varepsilon\frown a = a \neq \varepsilon = \sigma f(w\frown v)$$

But still, continuous extensions of word-homomorphisms are monotone wrt the prefix ordering:

**Proposition 8** (monotonicity of word-homomorphisms)
*For any word-homomorphism* $f : A^* \to B^*$ *and any words* $w, v \in A^s$, *if* $w \sqsubseteq v$ *then* $\sigma f w \sqsubseteq \sigma f v$.

Assume $A_1 \subseteq A$ is a subalphabet; $A_1$-*projection* is the unique continuous extension $\sigma\pi_{A_1} : A^s \to A_1^s$ of the unique word homomorphism $\pi_{A_1} : A^* \to A_1^*$ given by

$$\pi_{A_1} a \stackrel{\text{def}}{=} \begin{cases} a & \text{if } a \in A_1 \\ \varepsilon & \text{if } a \notin A_1 \end{cases}$$

The $\sigma$ in $\sigma\pi_{A_1}$ will be omitted.

**Proposition 9**
$\pi_{A_1} \circ \pi_{A_2} = \pi_{A_1 \cap A_2}$.

For any letter $a \in A$, denote the *number of occurrences of* $a$ in a word

$$\sharp_a : A^s \to \mathit{Nat} \cup \{\omega\} \quad \text{by}$$
$$\sharp_a \stackrel{\text{def}}{=} \sharp \circ \pi_{\{a\}}$$

For any word $w$ and any natural $k \in \mathit{Nat}$, $w^k$ is the word $w$ repeated $k$ times:

$$\sharp w^k \stackrel{\text{def}}{=} k \cdot \sharp w$$
$$w^k[n] \stackrel{\text{def}}{=} w[n \bmod \sharp w]$$

where $n \bmod \omega \stackrel{\text{def}}{=} n$. Analogously, $w^\omega$ is the word $w$ repeated infinitely many times:

$$\sharp w^\omega \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } w = \varepsilon \\ \omega & \text{if } w \neq \varepsilon \end{cases}$$
$$w^\omega[n] \stackrel{\text{def}}{=} w[n \bmod \sharp w]$$

Contiguous subwords will be denoted as follows:

$$w[k\,..\,k+i] \stackrel{\text{def}}{=} w[k]\ldots w[k+i]$$
$$w[k\,..\,] \stackrel{\text{def}}{=} w[k]\, w[k+1]\ldots$$

For any word $w \in A^s$, let

$$\mathit{Alph}\, w \stackrel{\text{def}}{=} \left\{ a \in A \,\middle|\, \sharp_a w > 0 \right\}$$

be called the *alphabet of word* $w$; likewise, for any set $W$ of $A$-words let

$$\mathit{Alph}\, W \stackrel{\text{def}}{=} \bigcup_{w \in W} \mathit{Alph}\, w$$

be called the *alphabet of set* $W$. The set

$$\mathit{Pref}\, w \stackrel{\text{def}}{=} \downarrow\{w\}$$

(cf. the definition of downcone in Sec. 1.2 on p. 9) is the *set of prefixes* of word $w$.

**Proposition 10**
*For any subalphabet* $A_1 \subseteq A$ *and any word* $w \in A^s$, $\mathit{Alph}\,(\pi_{A_1} w) \subseteq \mathit{Alph}\, w$.

A set of words will be referred to as *disjoint* if the alphabets of its words are pairwise disjoint; let $\mathcal{P}_d(A^s)$ denote the family of disjoint sets of $A$-words, then

$$W \in \mathcal{P}_d(A^s) \stackrel{\text{def}}{\iff}$$
$$\iff W \subseteq A^s \;\&\; (\forall_{w_1,w_2 \in W}\, \mathit{Alph}\, w_1 \cap \mathit{Alph}\, w_2 \neq \emptyset \Rightarrow w_1 = w_2)$$

A word-homomorphism $g : A^* \to B^*$ is *non-gluing for* a disjoint set $W$ of $A$-words if

$$\forall_{w_1,w_2 \in W}\, \mathit{Alph}\,(\sigma g w_1) \cap \mathit{Alph}\,(\sigma g w_2) \neq \emptyset \Rightarrow w_1 = w_2$$

i.e. if it preserves the disjointness of the alphabets of the words in $W$.

**Proposition 11**
*If a word-homomorphism* $g : A^* \to B^*$ *is non-gluing for a certain disjoint set* $W$ *of* $A$*-words, then*

$$(\pi_{\mathit{Alph}\,(\sigma g w)} \circ \sigma g)|_{(\mathit{Alph}\, W)^s} = (\sigma g \circ \pi_{\mathit{Alph}\, w})|_{(\mathit{Alph}\, W)^s}$$

*for any* $w \in W$*; i.e. the following diagram commutes:*

$$\begin{array}{ccc} (\mathit{Alph}\, W)^s & \xrightarrow{\sigma g} & B^s \\ \downarrow{\scriptstyle \pi_{\mathit{Alph}\, w}} & & \downarrow{\scriptstyle \pi_{\mathit{Alph}\,(\sigma g w)}} \\ (\mathit{Alph}\, w)^s & \xrightarrow[\sigma g]{} & (\mathit{Alph}\,(\sigma g w))^s \end{array}$$

Proof of Proposition 11:

First consider single-letter words $a \in \mathit{Alph}\, W$. If $a \in \mathit{Alph}\, w$ then

$$\sigma g a \in (\mathit{Alph}\,(\sigma g w))^*$$

and

$$\pi_{\mathit{Alph}\,(\sigma g w)}(\sigma g a) = \sigma g a = \sigma g(\pi_{\mathit{Alph}\, w} a)$$

If, in contrast, $a \in \mathit{Alph}\, W \setminus \mathit{Alph}\, w$ then $a \in \mathit{Alph}\, w'$ for a certain other $w' \in W$ and, as proved above,

$$\pi_{\mathit{Alph}\,(\sigma g w')}(\sigma g a) = \sigma g(\pi_{\mathit{Alph}\, w'} a) = \sigma g a$$

By the non-gluing property of $g$ for $W$, no letters of $\sigma g a$ belong to $\mathit{Alph}\,(\sigma g w)$; hence

$$\pi_{\mathit{Alph}\,(\sigma g w)}(\sigma g a) = \varepsilon$$

and also

$$\sigma g(\pi_{\mathit{Alph}\, w} a) = \sigma g \varepsilon = \varepsilon$$

To complete the proof, extend it by continuity to multi-letter words from $(\mathit{Alph}\, W)^s$.
□

# Chapter 2

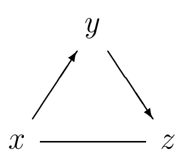
# Consistent equivalences and contractions

The model of concurrency discussed in the sequel leads to quite special partial orders. Some useful properties may however be discussed in whole generality for any partial orders. This is done here, prior to the introduction of particular partial orders related to concurrency.

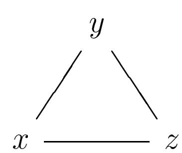
## 2.1 Factorization of partial orders

Assume $\sqsubseteq$ is a partial order and $\sim$ is an equivalence relation in a certain set $X$. Equivalence $\sim$ is *consistent with* partial order $\sqsubseteq$ if the following two conditions are satisfied:

1. $\forall_{x,y,z}\; x \sqsubseteq y \sqsubseteq z \;\&\; x \sim z \Rightarrow x \sim y$ , i.e. the following diagram

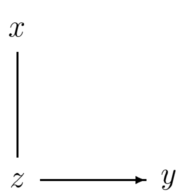



implies the diagram

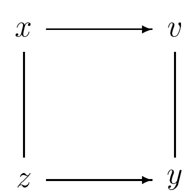



(where arrows stand for $\sqsubseteq$ and lines without arrowheads stand for $\sim$).

2. $\forall_{x,y,z}\; x \sim z \sqsubseteq y \Rightarrow \exists_v\, x \sqsubseteq v \sim y$ , i.e. the following diagram

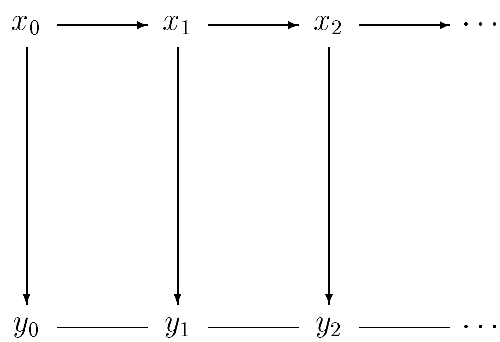



may be completed to

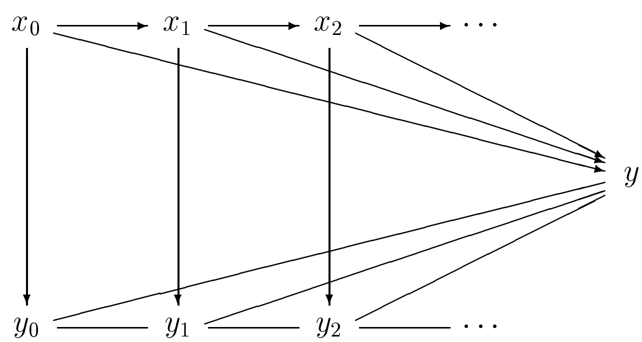



Equivalence $\sim$ is $\omega$*-consistent with* partial order $\sqsubseteq$ if it is consistent with $\sqsubseteq$ and satisfies the following additional condition called $\omega$-rule:

3. $\forall_{x_0,x_1,x_2,\ldots}\ \forall_{y_0,y_1,y_2,\ldots}$
$x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots\ \&\ y_0 \sim y_1 \sim y_2 \sim \ldots\ \&\ (\forall_{i \in Nat}\, x_i \sqsubseteq y_i)\ \Rightarrow$
$\exists_y\, \forall_{i \in Nat}\, x_i \sqsubseteq y \sim y_i$
i.e., every infinite diagram



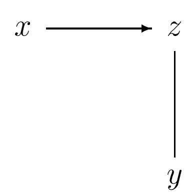

may be completed to




**Proposition 12**
*Equality is $\omega$-consistent with any partial order.*

Whenever an equivalence $\sim$ is consistent with a partial order $\sqsubseteq$, the quotient set of $X$ by $\sim$ has a natural partial order inherited from $\langle X, \sqsubseteq \rangle$:

$$\sqsubseteq\ \subseteq\ (X/_\sim) \times (X/_\sim)$$
$$[x]_\sim \sqsubseteq [y]_\sim \overset{\text{def}}{\Longleftrightarrow} \exists_z\, x \sqsubseteq z \sim y$$




**Proposition 13**
*The definition above is correct, i.e. it does not depend on a selection of $x$ and $y$ from their equivalence classes.*

Proof of Proposition 13:
This follows from condition 2 in the definition of consistency of $\sim$ with $\sqsubseteq$.
□

**Proposition 14**
*The above defined relation $\sqsubseteq\ \subseteq (X/_\sim) \times (X/_\sim)$ is a partial order.*

Proof of Proposition 14:

Reflexivity is obvious, transitivity follows from condition 2 and weak antisymmetry follows from conditions 2 and 1 in the definition of consistency of $\sim$ with $\sqsubseteq$.

□

**Proposition 15**

*Assume that an equivalence* $\sim$ *on* $X$ *is consistent with a partial order* $\sqsubseteq \subseteq X \times X$ *and that an equivalence* $\sim'$ *on* $X'$ *is consistent with a partial order* $\sqsubseteq' \subseteq X' \times X'$. *Assume futher that a certain monotone map* $f : X \to X'$ *is given, which preserves the equivalences:*

$$\forall_{x,y}\, x \sim y \Rightarrow fx \sim' fy \tag{2.1}$$

*Then the quotient map*

$$\begin{cases} \bar{f} : X/_\sim \to X'/_{\sim'} \\ \bar{f}\,[x]_\sim \stackrel{\text{def}}{=} [fx]_{\sim'} \end{cases}$$

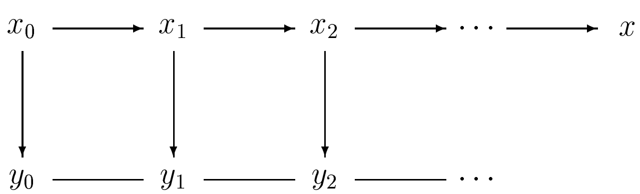
*is monotone wrt natural quotient partial orders.*

Proof of Proposition 15:

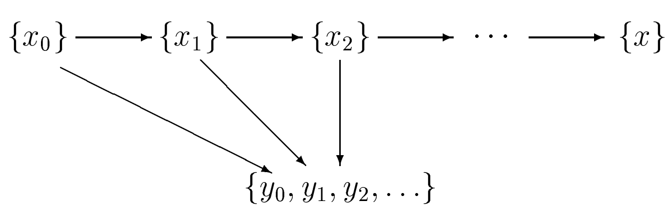
Condition (2.1) guarantees the well-definedness of the quotient map $\bar{f}$. Its monotonicity is straightforward.

□

If $\langle X, 0, \sqsubseteq\rangle$ is a cpo and an equivalence $\sim \subseteq X \times X$ is consistent with $\sqsubseteq$, then the quotient structure $\langle X/_\sim\,, [0]_\sim\,, \sqsubseteq\rangle$, while being a partial order by virtue of Prop. 14, may fail to be a cpo. For instance, if $X$ is




and the equivalence $\sim$ relates the $y_i$-s (as depicted), then the quotient is




The ascending chain $\{x_0\} \sqsubseteq \{x_1\} \sqsubseteq \{x_2\} \sqsubseteq \ldots$ has therefore two incomparable minimal upper bounds, hence it is not a cpo.

**Proposition 16**

*If* $\langle X, 0, \sqsubseteq\rangle$ *is a cpo and an equivalence* $\sim \subseteq X \times X$ *is* $\omega$*-consistent with* $\sqsubseteq$, *then* $\langle X/_\sim\,, [0]_\sim\,, \sqsubseteq\rangle$ *is a cpo too. Moreover, the natural quotient mapping*

$$\begin{cases} q : X \to X/_\sim \\ q\,x \stackrel{\text{def}}{=} [x]_\sim \end{cases}$$

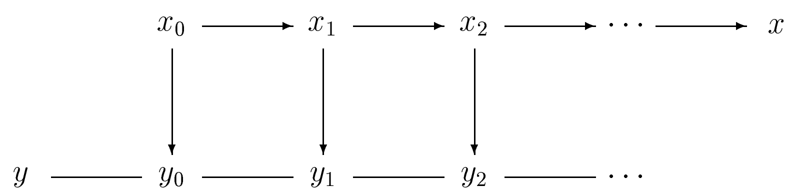
*is continuous.*

Proof of Proposition 16:

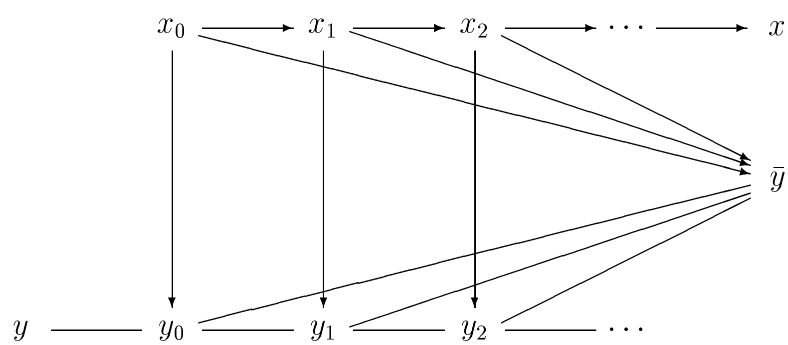
Assume $[x_0]_\sim \sqsubseteq [x_1]_\sim \sqsubseteq [x_2]_\sim \sqsubseteq \ldots$ is an ascending chain; select the representatives $x_0, x_1, x_2, \ldots$ from their $\sim$-classes so as to form an ascending chain in $X$; let $x$ be its lub:

$$x_0 \longrightarrow x_1 \longrightarrow x_2 \longrightarrow \cdots \longrightarrow x$$

Obviously, $[x]_\sim$ is an upper bound on $[x_i]_\sim$-s. We only have to demonstrate that it is *the least* upper bound. Assume $[y]_\sim$ is another upper bound on $[x_i]_\sim$-s; by the definition of the quotient order, the situation is as follows:




The $\omega$-rule now implies the existence of an $\bar{y}$ such that




Since $\bar{y}$ is an upper bound on $x_i$-s, $x \sqsubseteq \bar{y} \sim y$ and hence $[x]_\sim \sqsubseteq [y]_\sim$.
□

**Proposition 17**
*Assume* $\langle X, 0, \sqsubseteq \rangle$ *is a cpo and an equivalence* $\sim$ *on* $X$ *is* $\omega$*-consistent with* $\sqsubseteq$*. Assume further that*

$$x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots \quad \text{and}$$
$$y_0 \sqsubseteq y_1 \sqsubseteq y_2 \sqsubseteq \ldots$$

*are ascending chains. If* $x_n \sim y_n$ *for any* $n \in Nat$ *then*

$$\left(\bigsqcup_{n \in Nat} x_n\right) \sim \left(\bigsqcup_{n \in Nat} y_n\right)$$

Proof of Proposition 17:

Since, by virtue of Prop. 16, the natural quotient mapping

$$\begin{cases} q : X \to X/_\sim \\ q\,x \stackrel{\text{def}}{=} [x]_\sim \end{cases}$$

is continuous, we have

$$\left[\bigsqcup_{n\in Nat} x_n\right]_\sim = \bigsqcup_{n\in Nat} [x_n]_\sim = \bigsqcup_{n\in Nat} [y_n]_\sim = \left[\bigsqcup_{n\in Nat} y_n\right]_\sim$$

□

**Proposition 18**
*Let* $\langle X, 0, \sqsubseteq\rangle$ *and* $\langle X', 0', \sqsubseteq'\rangle$ *be cpos. Assume that an equivalence* $\sim$ *on* $X$ *is* $\omega$*-consistent with the partial order* $\sqsubseteq \subseteq X \times X$ *and that an equivalence* $\sim'$ *on* $X'$ *is* $\omega$*-consistent with the partial order* $\sqsubseteq' \subseteq X' \times X'$. *Assume futher that a certain continuous map* $f : X \to X'$ *is given, which preserves the equivalences:*

$$\forall_{x,y}\, x \sim y \Rightarrow fx \sim' fy$$

*Then the quotient map*

$$\begin{cases} \bar{f} : X/_\sim \to X'/_{\sim'} \\ \bar{f}\,[x]_\sim \stackrel{\text{def}}{=} [fx]_{\sim'} \end{cases}$$

*is continuous wrt natural quotient partial orders.*

Proof of Proposition 18:
The monotonicity of $\bar{f}$ follows from Prop. 15 on page 15. To prove that $\bar{f}$ preserves least upper bounds of chains, use the diagrams from the proof of Prop. 16.
□

## 2.2 Contractions

Assume now that $\sqsubseteq$ is a partial order in a set $X$. A *contraction*[1] is any relation $C \subseteq X \times X$ such that $x_1\, C\, x_2$ implies

1. $\exists_y\, x_1 \sqsubseteq y \;\&\; x_2 \sqsubseteq y$, i.e. $C$-related points have a common upper bound,
2. $\forall_{y_1}\, x_1 \sqsubseteq y_1 \Rightarrow \exists_{y_2}\, x_2 \sqsubseteq y_2 \;\&\; y_1\, C\, y_2$, i.e. going up from $x_1$ may be mimicked by going up from $x_2$ to a $C$-related point,
3. $\forall_{y_2}\, x_2 \sqsubseteq y_2 \Rightarrow \exists_{y_1}\, x_1 \sqsubseteq y_1 \;\&\; y_1\, C\, y_2$, i.e. the same, symmetrically.

(this is similar to bisimulations). As will be seen, contractions, under some extra conditions, serve for identifying elements in ascending chains, thus “contracting” these chains, without loosing the partial order properties.

**Proposition 19**
*Equality is a contraction.*

**Proposition 20**
*If* $C$ *is a contraction, then so is its transposition* $C^T$ *defined by*

$$x\, C^T\, y \stackrel{\text{def}}{\Longleftrightarrow} y\, C\, x$$

[1]Marek Bednarczyk prefers another term “collapse”.

**Proposition 21**
*If* $C_1$ *and* $C_2$ *are contractions, then so is their composition* $C_1;C_2$ *defined by*

$$x\,(C_1;C_2)\,y \stackrel{\text{def}}{\Longleftrightarrow} \exists_z\, x\, C_1\, z \;\&\; z\, C_2\, y$$

Contractions are not necessarily equivalence relations, e.g. the empty relation is not, but the unique greatest contraction is an equivalence, as implied by the following theorem:

**Theorem 2**
*The set-theoretical union* $C_0$ *of all contractions in* $X$ *is a contraction itself and an eqivalence in* $X$*.*

Proof of Theorem 2:

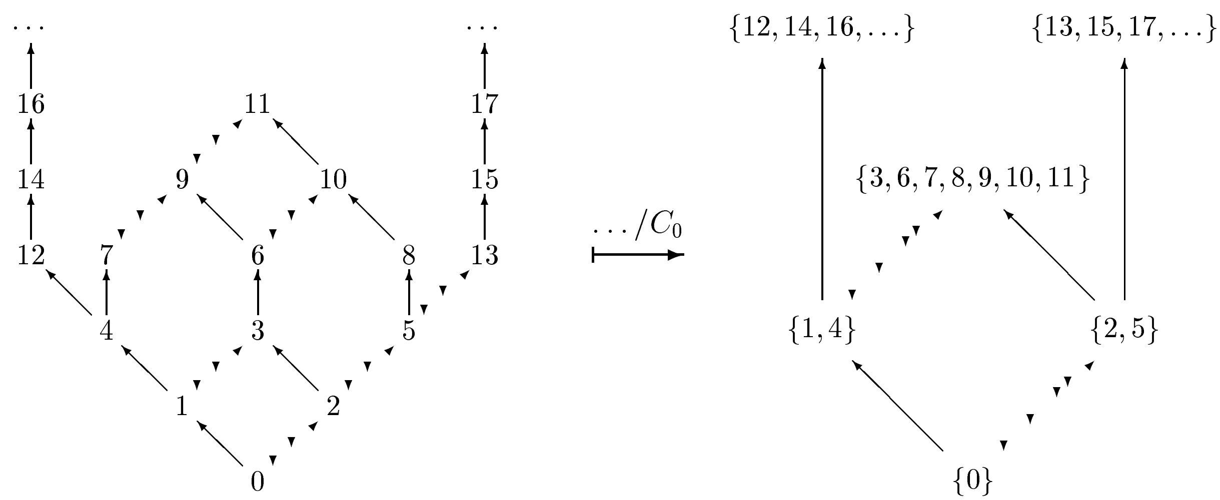
Prop. 19 implies that $C_0$ is reflexive; Prop. 20 implies that $C_0$ is symmetric; Prop. 21 implies that $C_0$ is transitive. Therefore, $C_0$ is an equivalence relation.

It is straightforward that $C_0$ is a contraction.

□

The set-theoretical union $C_0$ of all contractions in $X$ is called the *maximal contraction* in $X$. Informally speaking, $C_0$ relates the elements from which the reachability of the maximal points and the reachability of infinite ascending chains in $X$ are the same. In particular, it does not relate distinct maximal points; neither it relates any such points $x_1$ and $x_2$ that $x_1 \sqsubseteq y$ and $x_n \not\sqsubseteq y$ where $y$ is a maximal point; neither it relates a beginning of an ascending chain with a point that does not begin any infinite ascending chain.

Consider an example partial order and its factorization by the maximal contraction:

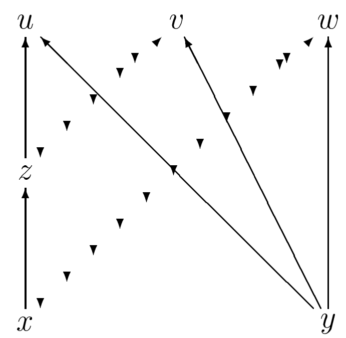



As you can see, the infinite paths 12-14-16-... and 13-15-17-..., but also the finite paths 7-9-11, 3-6-9-11 etc. are redundant from the point of view of reachability of maximal elements and for this reason they have been contracted to single points. One may interpret the equivalence classes of $C_0$ from the point of view of “decisions”. For instance, in points 3, 6, 7, etc. the decision has already been taken to go to 11 rather then to engage in one of the infinite paths. Point 3 has therefore been identified with all other points from which one may only go to 11. On the other hand, in 2 and in 5 it has been decided *either* to go to 11 *or* to engage in the odd-numbered infinite path and not to engage in the even-numbered infinite path. Point 0 is still different: no decisions have yet been taken.

The reader should, however, be warned against getting too hooked on this kind of informal explanations. In particular, it is not true that $C_0$ glues together exactly those points, which are smaller from exactly the same maximal elements. The relation $M$ which does this, namely

$$x \, M \, y \overset{\text{def}}{\iff} (\forall_{z \in \max X} \; x \sqsubseteq z \iff y \sqsubseteq z)$$

is not necessarily a contraction at all, let alone the maximal contraction $C_0$. Indeed, in




the points $x$ and $y$ are $M$-related but they are not $C_0$-related. If they were, then $z$ would have to be $C_0$-related to one of $u$, $v$, $w$ or $y$, which is not the case.

**Theorem 3**
*The maximal contraction in $X$ is consistent with the partial order $\sqsubseteq$ in $X$.*

Proof of Theorem 3:
CONDITION 1 from the definition of consistency in Sec. 2.1 (page 13):

Assume $x \sqsubseteq y \sqsubseteq z$ and $x \, C_0 \, z$. As the maximal contraction, $C_0$ includes equality (cf. Prop. 19):

$$\left\{ \langle x, x \rangle \,\middle|\, x \in X \right\} \subseteq C_0$$

Define a new relation $\bar{C}$ by

$$\bar{C} \overset{\text{def}}{=} C_0 \cup \{\langle x, y \rangle\}$$

Since $C_0$ is the union of *all* contractions, we only have to prove that $\bar{C}$ is a contraction. To do this, we have to verify conditions 1–3 from the definition of contraction for the new pair $\langle x, y \rangle$. Cond. 1 is obvious, because $y$ is an upper bound on $x$ and $y$.

Assume $x \sqsubseteq v$ for a certain $v \in X$. Since $C_0$ is a contraction and $x \, C_0 \, z$, there exists a $w \in X$ such that $z \sqsubseteq w \, C_0 \, v$. But then also $y \sqsubseteq z \sqsubseteq w$, hence the pair $\langle x, y \rangle$ satisfies cond. 2.

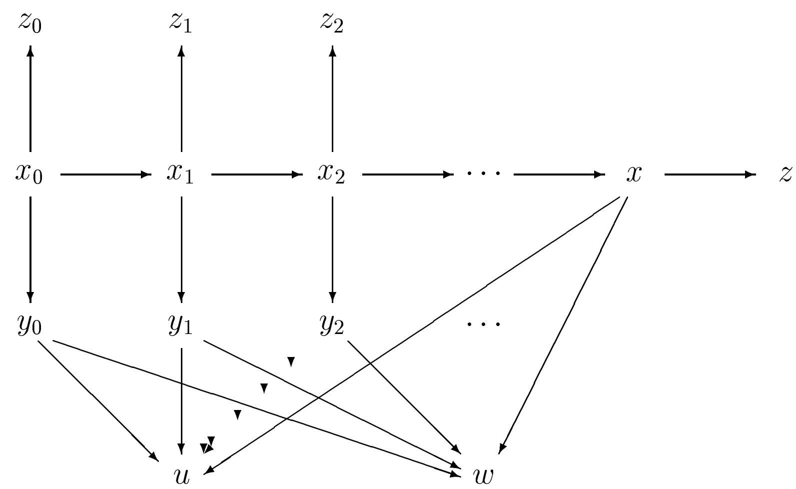
Now assume $y \sqsubseteq v$ for a certain $v \in X$. Then $x \sqsubseteq y \sqsubseteq v$ and $v \, C_0 \, v$ (because $C_0$ includes equality); hence the pair $\langle x, y \rangle$ satisfies cond. 3.

CONDITION 2 from the definition of consistency in Sec. 2.1 (page 13):

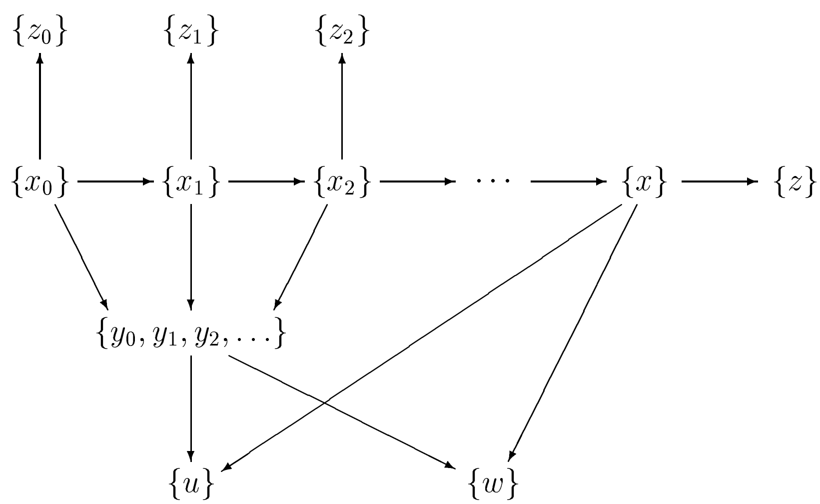
Assume $z \sqsubseteq y$ and $x \, C_0 \, z$. Since $C_0$ is a contraction, there exists a $v \in X$ such that $x \sqsubseteq v$ and $v \, C_0 \, y$ (cond. 3 from the definition of contraction).
□

By virtue of Prop. 14 on page 14, the quotient of a partially ordered set by its maximal contraction is a partially ordered set again. Moreover, the transition to quotients is functorial, i.e. the quotients of monotone mappings are monotone. The factorization by the maximal contraction may be viewed as a standard simplification procedure for partial orders, shortening ascending paths (hence the term "contraction"), and "sliming" the partially ordered bushes.

Unfortunately, even if $\langle X, 0, \sqsubseteq \rangle$ is a cpo, the maximal contraction $C_0$ is not necessarily $\omega$-consistent with $\sqsubseteq$ and $\langle X/C_0\,,\ [0]_{C_0}\,,\ \sqsubseteq \rangle$ may fail to be a cpo. Indeed, consider the following example:




This is surely a cpo with $x = \bigsqcup_{i \in Nat} x_i$. Most $C_0$-classes are one-element with just one exception: $y_i$-s are $C_0$-related. The quotient order is thus:




Now, both $\{x\}$ and $\{y_0, y_1, y_2, \ldots\}$ are incomparable minimal upper bounds on $\{x_i\}$-s, hence the lub does not exist.

Further in this paper we will see (Thm. 7 on page 41) that the maximal contractions in the particular case of the cpo-s, which emerge in the investigation of concurrent processes, are $\omega$-consistent and therefore the respective quotients are cpo-s.

# Chapter 3

# Concurrent processes

Vectors of processes, introduced here, are the main object of our investigations. Later, the tools for such investigations based on homotopy are discussed.

## 3.1 Vectors of processes

*Vector of processes*[1] is any quadruple

$$\mathcal{V} = \langle P, A, L, I \rangle$$

where

- $P$ is an arbitrary set of *processes*.
- $A = \{A_p\}_{p \in P}$ is a mutually disjoint family of sets of *actions*. Sometimes the letter $A$ will also be used to denote the set of all actions, $A = \bigcup_{p \in P} A_p$.
- $L : \prod_{p \in P} A_p^s$ is the *software for* vector $\mathcal{V}$ [2]; it consists of *programs for* particular processes, $Lp \in A_p^s$ for $p \in P$. Programs for processes are assumed to involve all actions in their respective alphabets $A_p$:

  $$Alph\,(Lp) = A_p \tag{3.1}$$

  for $p \in P$.
- $I \subseteq Conf_f\, \mathcal{V}$ is the set of *admissible configurations* (where $Conf\, \mathcal{V} \stackrel{\text{def}}{=} \prod_{p \in P} Pref\,(Lp)$ [3] is the set of *configurations* and $Conf_f\, \mathcal{V} \stackrel{\text{def}}{=} \prod_{p \in P} (A_p^* \cap Pref\,(Lp))$ is the set of *finitary configurations*); the finitary configurations which are not admissible, those in $O \stackrel{\text{def}}{=} Conf_f\, \mathcal{V} \setminus I$, are called *excluded configurations* or *obstacles*.

Informally, a program $Lp$ is the sequence of actions that process $p$ "wants" to engage in, but it may be delayed or prevented from doing so by an interference from other processes. A configuration is a collection of prefixes of programs and this collection characterizes the state of execution of particular processses. The set $Conf\, \mathcal{V}$ of configurations has a natural cpo structure given by the component-wise prefix ordering. Some finitary configurations,

[1] Wiesław Pawłowski has proposed the term "multiprocess".

[2] Recall form Sec. 1.1 on page 8 that $\prod$ denotes the generalized Cartesian product.

[3] Recall from Sec. 1.3 on page 12 that $Pref\, w$ is the set of prefixes of word $w$.

for instance the ones in which the requirement of an exclusive access to a scarce resource is violated, may be forbidden (excluded configurations).

A vector of processes $\mathcal{V} = \langle P, A, L, I\rangle$ as above generates sequences of actions which are consistent with the vector's software and admissible, in the following sense. Define the function *endpt* that yields the projections of a given finite $A$-word to $A_p$-s for $p \in P$:

$$\begin{cases} endpt \ : A^* \to \prod_{p\in P} A_p^* \\ endpt\, w\, p \stackrel{\text{def}}{=} \pi_{A_p} w \end{cases}$$

A word $w \in A^s$ is said to be *consistent with* the software $L$ of vector $\mathcal{V}$ if the endpoints of its finite prefixes are in $Conf\,\mathcal{V}$:

$$\forall_{v\in A^*}\ v \sqsubseteq w \Rightarrow endpt\, v \in Conf\,\mathcal{V}$$

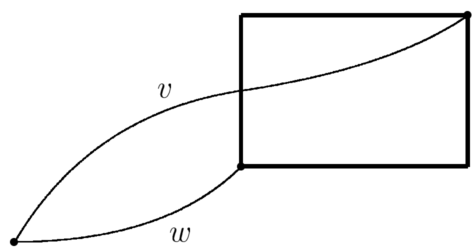
**Proposition 22**
*A word $w \in A^s$ is consistent with $L$ if and only if*

$$\forall_{p\in P}\, \pi_{A_p} w \sqsubseteq Lp$$

Informally, a word is consistent with $L$ if it is a "zip-join" of prefixes of programs. The set of words which are consistent with the software of a vector $\mathcal{V}$ will be denoted by $Cons\,\mathcal{V}$.

Any two consistent words define a *cube* describing the configurations contained between their endpoints:

$$\begin{cases} \square : Cons\,\mathcal{V} \times Cons\,\mathcal{V} \to \mathcal{P}(Conf\,\mathcal{V}) \\ \square\,\langle w, v\rangle \stackrel{\text{def}}{=} \left\{c \,\middle|\, endpt\, w \sqsubseteq c \sqsubseteq endpt\, v\right\} \end{cases}$$




**Proposition 23**
*For any finite word $w \in A^*$ and any action $a \in A$, such that $w^\frown a \in Cons\,\mathcal{V}$, the cube between $w$ and $w^\frown a$ consists of exactly two configurations:*

$$\square\,\langle w\,,\, w^\frown a\rangle = \{endpt\, w\,,\, endpt\,(w^\frown a)\}$$

Let $\mathcal{V} = \langle P, A, L, I\rangle$ and $\mathcal{V}' = \langle P', A', L', I'\rangle$ be vectors of processes and let $g : A^* \to A'^*$ be a consistency-preserving word-homomorphism; i.e. $(\sigma g)^{\to}(Cons\,\mathcal{V}) \subseteq Cons\,\mathcal{V}'$. Define the *g-closeness* relation

$$\ll_g \subseteq\ Cons\,\mathcal{V}' \times Cons\,\mathcal{V}$$

as follows:

$$\begin{aligned} v' \ll_g w \stackrel{\text{def}}{\Longleftrightarrow}\ & (v' = \varepsilon \ \&\ w = \varepsilon) \vee \\ & (\exists_{w_0 \in Cons\,\mathcal{V}\,\cap A^*} \exists_{a\in A}\, w = w_0{}^\frown a \ \&\ \sigma g w_0 \sqsubsetneq v' \sqsubseteq \sigma g w) \vee \\ & (w \in Cons\,\mathcal{V} \setminus A^* \ \&\ v' = \sigma g w) \end{aligned}$$

($\sqsubsetneq$ is the proper-prefix relation: $w_1 \sqsubsetneq w_2 \stackrel{\text{def}}{\Longleftrightarrow} w_1 \sqsubseteq w_2 \ \&\ w_1 \neq w_2$). Informally, $v'$ is $g$-close to $w$ if it is a prefix of $\sigma g w$ and lies within a single $g$-step from $\sigma g w$. This notion will be used in the definition of vector morphism in Sec. 3.2.

**Proposition 24**
*Assume* $\mathcal{V} = \langle P, A, L, I\rangle$ *and* $\mathcal{V}' = \langle P', A', L', I'\rangle$ *are vectors of processes and* $g : A^* \to A'^*$ *is a consistency-preserving word-homomorphism. Then* $\sigma g w \ll_g w$ *for any* $w \in \mathit{Cons}\,\mathcal{V}$.

**Proposition 25**
*Assume* $\mathcal{V} = \langle P, A, L, I\rangle$ *and* $\mathcal{V}' = \langle P', A', L', I'\rangle$ *are vectors of processes and* $g : A^* \to A'^*$ *is a consistency-preserving word-homomorphism. If* $v' \sqsubseteq \sigma g w$ *for certain* $w \in \mathit{Cons}\,\mathcal{V}$ *then there exists a* $w_0 \in \mathit{Cons}\,\mathcal{V}$ *such that* $v' \ll_g w_0 \sqsubseteq w$.

**Proposition 26**
*Assume* $\mathcal{V} = \langle P, A, L, I\rangle$, $\mathcal{V}' = \langle P', A', L', I'\rangle$ *and* $\mathcal{V}'' = \langle P'', A'', L'', I''\rangle$ *are vectors of processes and* $A^* \xrightarrow{g'} A'^* \xrightarrow{g''} A''^*$ *are consistency-preserving word-homomorphisms. Then*

$$\ll_{g''\circ g'} = \ll_{g''} ; \ll_{g'}$$

*i.e., for any words* $w \in \mathit{Cons}\,\mathcal{V}$ *and* $v'' \in \mathit{Cons}\,\mathcal{V}''$,

$$v'' \ll_{g''\circ g'} w \iff \exists_{u' \in \mathit{Cons}\,\mathcal{V}'}\; v'' \ll_{g''} u' \;\&\; u' \ll_{g'} w$$

A word $w \in A^s$ is *admissible* (or *avoids obstacles*) in $\mathcal{V}$ if the endpoints of its all prefixes are in the set $I$ of admissible configurations:

$$\forall_{v \in A^*}\; v \sqsubseteq w \Rightarrow \mathit{endpt}\, v \in I$$

Since $I \subseteq \mathit{Conf}\,\mathcal{V}$, the admissibility is a stronger property than the consistency. An admissible word will be called a *path generated* by vector $\mathcal{V}$:

$$\mathit{Pth}\,\mathcal{V} \stackrel{\text{def}}{=} \left\{ w \in A^s \,\middle|\, \forall_{v \in A^*}\; v \sqsubseteq w \Rightarrow \mathit{endpt}\, v \in I \right\}$$

## 3.2 Vector morphisms

Vector morphisms, discussed in this section, may be informally thought of as translating higher-level vectors of processes to lower-level ones. A vector morphism consists of two components. The first one, $f$, contravariantly retrieves from any lower-level process $p'$ the higher-level process $fp'$ in whose implementation $p'$ participates. For any higher-level process $p$, the coimage $f^{\leftarrow}\{p\} \stackrel{\text{def}}{=} \left\{ p' \in P' \,\middle|\, fp' = p \right\}$ may be thought of as consisting of processes which cooperate on a lower-level to achieve the goals originally achieved by $p$. The other component, $g$, translates sequences of high-level actions to sequences of low-level actions. The definition of vector morphism that follows contains a number of conditions to guarantee a certain kind of consistency between $f$ and $g$ and between the both components and the structure of vectors of processes involved.

But more than this is needed for a pragmatically sound notion of implementation; the vector morphism defined and discussed below is just a "raw material" for this. Implementations are studied in Sec. 3.3 (page 26).

Given two vectors $\mathcal{V} = \langle P, A, L, I\rangle$ and $\mathcal{V}' = \langle P', A', L', I'\rangle$, a *vector morphism* from $\mathcal{V}$ to $\mathcal{V}'$ is any pair $\langle f, g\rangle : \mathcal{V} \to \mathcal{V}'$ with

- $f : P' \to P$ — (contravariant! [4]) assigning processes to processes,

[4] In earlier versions of this report, this read covariantly $f : P \to \mathcal{P}(P')$, so nothing really changed.

- $g : A^* \to A'^*$ — word-homomorphism on actions

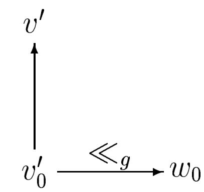
satisfying the following conditions:

1. $\pi_{A'_{p'}}(\sigma g(Lp)) = \begin{cases} L'p' & \text{if } p = fp' \\ \varepsilon & \text{if } p \neq fp' \end{cases}$ for $p \in P$ and $p' \in P'$.
   Informally, the image via $g$ of the program for a process $fp' \in P'$ is a "zip-join" of programs $L'p''$ for all processes $p'' \in P''$ that implement $f'p'$, i.e. satisfy $f'p'' = f'p'$.

2. $\Box \langle w, v \rangle \subseteq I$ implies $\Box \langle \sigma g w, \sigma g v \rangle \subseteq I'$ for $w, v \in Cons\, \mathcal{V}$.
   Informally, $g$ avoids obstacles in the sense that no obstacles may appear between words $\sigma g w$ and $\sigma g v$ unless they have already been present between $w$ and $v$.

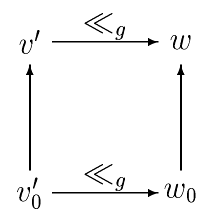
3. For any $\mathcal{V}$-path $w_0$ and any $\mathcal{V}'$-paths $v'_0, v'$, if

   $$v'_0 \sqsubseteq v' \;\&\; v'_0 \ll_g w_0$$

   (see the definition of $g$-closeness $\ll_g$ on page 22) then there exists a $\mathcal{V}$-path $w$ such that

   $$w_0 \sqsubseteq w \;\&\; v' \ll_g w$$

Note that Condition 3 refers to the notion of paths generated (cf. page 23). This notion implicitly involves the softwares $L$ and $L'$ of both vectors $\mathcal{V}$ and $\mathcal{V}'$ and also their sets $I$ and $I'$ of admissible configurations. For Cond. 3 to make sense, it is indispensable that $g$ be path-preserving. This is proved in Prop. 33 on page 35. Until then, no propositions nor theorems make use of Cond. 3.

**Proposition 27**
*If $\langle f, g \rangle : \mathcal{V} \to \mathcal{V}'$ is a vector morphism then*

$$Alph\,(g^{\to} A_p) = \bigcup \left\{ A'_{p'} \,\middle|\, fp' = p \right\}$$

*for any process $p \in P$.*

Proof of Proposition 27:

This follows from (3.1) on page 21 and from condition 1 in the definition of vector morphism.
□

**Proposition 28**
*If $\langle f, g \rangle : \mathcal{V} \to \mathcal{V}'$ is a vector morphism then $g$ is non-gluing for $L^{\to} P$ (cf. the definition of "non-gluing" on p. 12).*

**Proposition 29**
*If* $\langle f, g\rangle : \mathcal{V} \to \mathcal{V}'$ *is a vector morphism then* $g$ *is consistency-preserving.*

Given two vector morphisms $\mathcal{V} \stackrel{\langle f', g'\rangle}{\longrightarrow} \mathcal{V}' \stackrel{\langle f'', g''\rangle}{\longrightarrow} \mathcal{V}''$, their *composition*

$$\langle f, g\rangle = \langle f'', g''\rangle \circ \langle f', g'\rangle : \mathcal{V} \to \mathcal{V}''$$

is defined in a natural way:

$$\begin{cases} f : P'' \to P \\ f \stackrel{\text{def}}{=} f' \circ f'' \end{cases} \qquad \begin{cases} g : A \to A'' \\ g \stackrel{\text{def}}{=} g'' \circ g' \end{cases}$$

**Proposition 30**
*Pair* $\langle f, g\rangle$ *defined as above is a vector morphism.*

Proof of Proposition 30:

Take an arbitrary process $p' \in P'$ and denote the union $\bigcup \left\{ A''_{p''} \,\middle|\, f''p'' = p' \right\}$ by $Q_{p'}$. Since, by virtue of (3.1) on page 21, the set $\left\{ L'p' \,\middle|\, p' \in P' \right\}$ is disjoint and $g''$ is non-gluing (cf. Prop. 28), Prop. 11 on page 12 implies that

$$\begin{aligned} &\sigma g'' \circ \pi_{A'_{p'}} = \\ &= \pi_{Alph\,(\sigma g''(L'p'))} \circ \sigma g'' = \\ &= \pi_{Alph\,(g''^{\to}(Alph\,(L'p')))} \circ \sigma g'' = \\ &= \pi_{Alph\,(g''^{\to} A'_{p'})} \circ \sigma g'' = \qquad \text{(by Prop. 27)} \\ &= \pi_{Q_{p'}} \circ \sigma g'' \end{aligned}$$

Assume $p''_0 \in P''$ and apply the above equality to $f''p''_0$ for $p'$:

$$\begin{aligned} &\pi_{A''_{p''_0}} \circ \sigma g'' \circ \pi_{A'_{f''p''_0}} = \\ &= \pi_{A''_{p''_0}} \circ \pi_{Q_{p'}} \circ \sigma g'' = \quad \text{(by Prop. 9, since } A''_{p''_0} \subseteq Q_{p'} \text{)} \\ &= \pi_{A''_{p''_0}} \circ \sigma g'' \end{aligned}$$

Therefore,

$$\begin{aligned} &L''p''_0 = && \left(\begin{matrix}\text{since } \langle f'', g''\rangle \text{ satisfies Cond. 1 from} \\ \text{the definition of vector morphism}\end{matrix}\right) \\ &= \pi_{A''_{p''_0}}(\sigma g''(L'(f''p''_0))) = && \left(\begin{matrix}\text{since } \langle f', g'\rangle \text{ satisfies Cond. 1 from} \\ \text{the definition of vector morphism}\end{matrix}\right) \\ &= \pi_{A''_{p''_0}}(\sigma g''(\pi_{A'_{f''p''_0}}(\sigma g'(L(f'(f''p''_0)))))) = \\ &= \pi_{A''_{p''_0}}(\sigma g''(\sigma g'(L(f'(f''p''_0))))) = && \text{(by Prop. 7 on page 10)} \\ &= \pi_{A''_{p''_0}}(\sigma g(L(fp''_0))) \end{aligned}$$

This proves that $\langle f, g\rangle$ satisfies Cond. 1.

The verification of Cond. 2 is straightforward. The verification of Cond. 3 requires Prop. 26 from page 23 and is left to the reader as well.

□

For any vector $\mathcal{V}$ of processes define the *identity morphism* $I_{\mathcal{V}} = \langle f_0, g_0 \rangle : \mathcal{V} \to \mathcal{V}$ by

$$\begin{cases} f_0\, p \stackrel{\text{def}}{=} p \\ g_0\, a \stackrel{\text{def}}{=} a \end{cases}$$

**Proposition 31**
*The identity morphism defined above is a vector morphism.*

**Theorem 4**
*Vectors of processes with vector morphisms, composition and identity, as defined above, form a category.*

Proof of Theorem 4:
By the direct verification of

- associativity of composition of vector morphisms:

$$(\langle f'', g'' \rangle \circ \langle f', g' \rangle) \circ \langle f, g \rangle = \langle f'', g'' \rangle \circ (\langle f', g' \rangle \circ \langle f, g \rangle)$$

- neutrality of identity wrt composition:

$$\langle f, g \rangle \circ I_{\mathcal{V}} = I_{\mathcal{V}'} \circ \langle f, g \rangle = \langle f, g \rangle$$

  for $\langle f, g \rangle : \mathcal{V} \to \mathcal{V}'$.

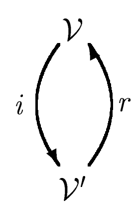
□

The category referred to in Thm. 4 is denoted by *Vect*.

## 3.3 Implementations

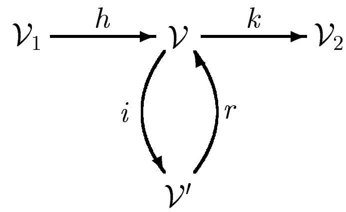
As has already been informally explained in Sec. 3.2, morphisms may serve to implementing vectors of processes in other vectors of processes. But for a pragmatically useful notion of implementation, one needs morphisms going both ways: the one from higher to lower level serving as the proper implementation, i.e. translating higher-level notions to lower-level ones; and the other from lower to higher level retrieving the high-level behaviour from its low-level elements.

Formally, *implementation of* vector $\mathcal{V}$ in vector $\mathcal{V}'$ is any pair of morphisms

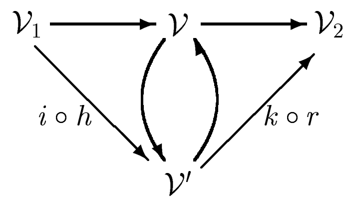



such that $r \circ i = I_{\mathcal{V}}$. Such a pair of morphisms is often called *retraction*. Morphism $i$ serves to translate from high-level $\mathcal{V}$ to low-level $\mathcal{V}'$; morphism $r$ reads the low-level $\mathcal{V}'$-behaviour back in high-level terms.

Given a "pipeline" of morphisms and an implementation $\langle i, r \rangle$ of $\mathcal{V}$ in $\mathcal{V}'$:




the morphisms $h$ and $k$ may be refined to a lower-level pipe:




with no externally visible difference of behaviour, since

$$(k \circ r) \circ (i \circ h) = k \circ (r \circ i) \circ h = k \circ I_{\mathcal{V}} \circ h = k \circ h$$

# Chapter 4

# Examples of vectors of processes

## 4.1 Reader and writer

Consider the vector

$$\mathcal{V}^{rw} = \langle P^{rw}, A^{rw}, L^{rw}, I^{rw} \rangle$$

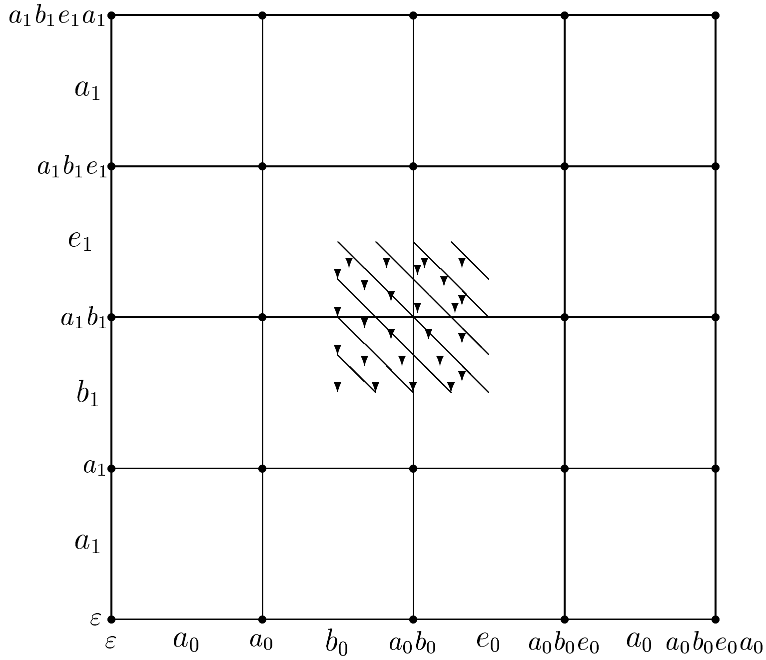
consisting of processes named 0 and 1 performing actions $b_0$ (beginning of reading), $e_0$ (end of reading), $b_1$ (beginning of writing), $e_1$ (end of writing) and $a_0, a_1$ (anything else):

$$P^{rw} \stackrel{\text{def}}{=} \{0, 1\}$$
$$L^{rw}\,0 \stackrel{\text{def}}{=} a_0 b_0 e_0 a_0 \quad L^{rw}\,1 \stackrel{\text{def}}{=} a_1 b_1 e_1 a_1$$

This means process 0 performs the neutral action $a_0$, then reads $b_0 e_0$, then runs $a_0$ again; and process 1 goes $a_1$, then writes $b_1 e_1$, then $a_1$ again; $A^{rw}$ is thus implicitely defined as $\{a_0, b_0, e_0, a_0, a_1, b_1, e_1, a_1\}$. The set of obstacles consists of the configuration $\langle a_0 b_0, a_1 b_1 \rangle$ corresponding to the situation when the reader and the writer are simultaneously using the resource:

$$I^{rw} \stackrel{\text{def}}{=} \mathit{Conf}_f\, \mathcal{V}^{rw} \setminus \{\langle a_0 b_0, a_1 b_1 \rangle\}$$

The paths generated by $\mathcal{V}^{rw}$ are the finite ones beginning in $\langle \varepsilon, \varepsilon \rangle$ and going from left to right and from bottom upwards in the configuration space $\mathit{Conf}\, \mathcal{V}^{rw}$ and not intersecting the shaded area, i.e. the obstacle, at the following picture:




## 4.2 Complete sequentialization

Consider the vector

$$\mathcal{V}^{cs} = \langle P^{cs}, A^{cs}, L^{cs}, I^{cs} \rangle$$

consisting of processes named 0 and 1 indefinitely performing actions $a_0$ and $a_1$, respectively:

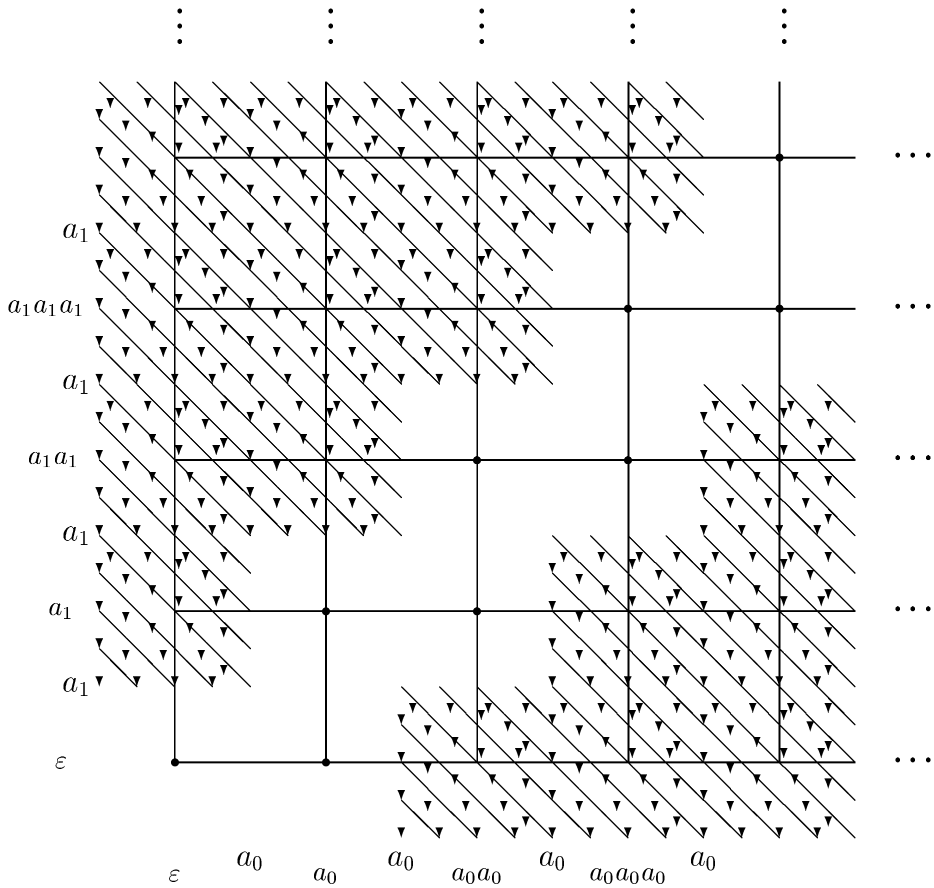
$$P^{cs} \stackrel{\text{def}}{=} \{0, 1\}$$
$$A^{cs} \stackrel{\text{def}}{=} \{a_0, a_1\}$$
$$L^{cs}\,0 \stackrel{\text{def}}{=} (a_0)^\omega \quad L^{cs}\,1 \stackrel{\text{def}}{=} (a_1)^\omega$$

with admissible configurations:

$$I^{cs} \stackrel{\text{def}}{=} \left\{ \left\langle a_0^n, a_1^k \right\rangle \middle| k \leq n \leq k+1 \right\}$$

This means that the admissible paths begin with $a_0$ and no two adjacent actions are equal. Thus, any admissible path is a prefix of $(a_0a_1)^\omega$.

The paths generated by $\mathcal{V}^{cs}$ are the finite and infinite ones beginning in $\langle \varepsilon, \varepsilon \rangle$ and going from left to right and from bottom upwards in the configuration space *Conf* $\mathcal{V}^{cs}$ depicted below and not intersecting the shaded area:




## 4.3 Producer and consumer with a 1-place buffer

Consider the vector

$$\mathcal{V}^{pc1} = \left\langle P^{pc1}, A^{pc1}, L^{pc1}, I^{pc1} \right\rangle$$

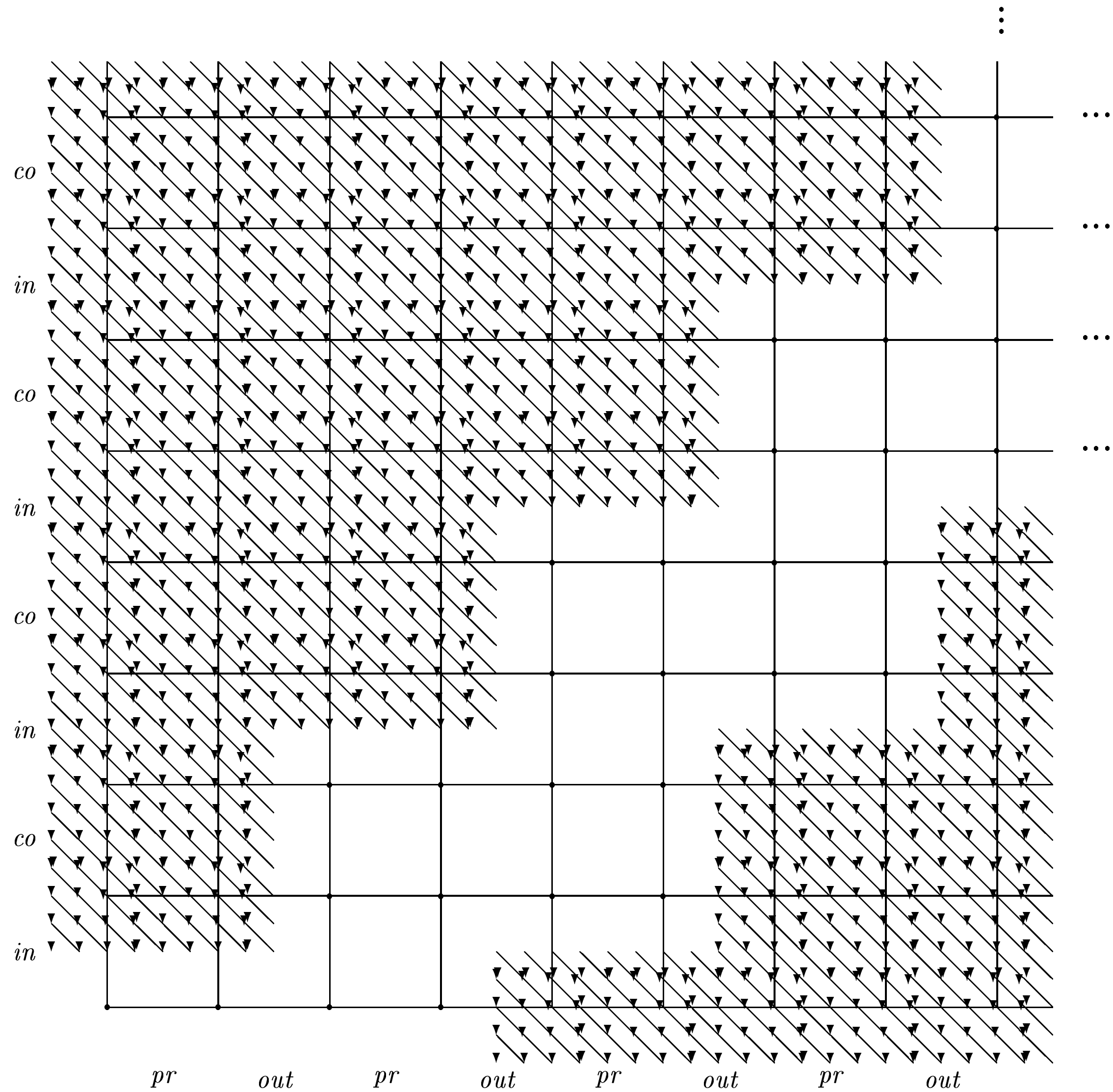
consisting of processes named 0 and 1 indefinitely performing actions *pr* (produce), *out* (output to buffer), *in* (input from buffer) and *co* (consume):

$$P^{pc1} \stackrel{\text{def}}{=} \{0, 1\}$$
$$L^{pc1}\, 0 \stackrel{\text{def}}{=} (pr\ out)^{\omega} \quad L^{pc1}\, 1 \stackrel{\text{def}}{=} (in\ co)^{\omega}$$

with the obvious admissible configurations:

$$I^{pc1} \stackrel{\text{def}}{=} \left\{ \langle w, v \rangle \sqsubseteq \langle (pr\ out)^{\omega}, (in\ co)^{\omega} \rangle \,\middle|\, \sharp_{in} w \leq \sharp_{out} w \leq \sharp_{in} w + 1 \right\}$$

The paths generated by $\mathcal{V}^{pc1}$ are the ones that go from left to right and from bottom upwards and do not intersect the shaded area at the following picture:




## 4.4 Implementation of $\mathcal{V}^{cs}$ in $\mathcal{V}^{pc1}$

Define mappings:

$$\begin{cases} f_i : P^{pc1} \to P^{cs} \\ f_i\, p \stackrel{\text{def}}{=} p \end{cases} \qquad \begin{cases} g_i : (A^{cs})^* \to (A^{pc1})^* \\ g_i\, a_0 \stackrel{\text{def}}{=} pr\ out \\ g_i\, a_1 \stackrel{\text{def}}{=} in\ co \\ g_i\, w \text{ — extended homomorphically} \end{cases}$$

and

$$\begin{cases} f_r : P^{cs} \to P^{pc1} \\ f_r\, p \stackrel{\text{def}}{=} \{p\} \end{cases} \qquad \begin{cases} g_r : (A^{pc1})^* \to (A^{cs})^* \\ g_r\, out \stackrel{\text{def}}{=} a_0 \\ g_r\, in \stackrel{\text{def}}{=} a_1 \\ g_r\, a \stackrel{\text{def}}{=} \varepsilon \quad \text{for } a \in \{pr, co\} \\ g_r\, w \text{ — extended homomorphically} \end{cases}$$

It is easy to check that

$$i = \langle f_i, g_i \rangle : \mathcal{V}^{cs} \to \mathcal{V}^{pc1} \quad \text{and} \quad r = \langle f_r, g_r \rangle : \mathcal{V}^{pc1} \to \mathcal{V}^{cs}$$

are vector morphisms. Furthermore,

$$f_i(f_r p) = p$$

and

$$g_r(g_i\, a_0) = g_r(pr\ out) = a_0$$
$$g_r(g_i\, a_1) = g_r(in\ co) = a_1$$

Therefore, $r \circ i = I_{\mathcal{V}^{cs}}$ and pair $\langle i, r \rangle$ is an implementation of vector $\mathcal{V}^{cs}$ in vector $\mathcal{V}^{pc1}$.

## 4.5 Producer and consumer with a 2-place buffer

A 2-place buffer is assisted by an additional process which shifts elements from the first to the second place in the buffer:

$$\mathcal{V}^{pc2} = \left\langle P^{pc2}, A^{pc2}, L^{pc2}, I^{pc2} \right\rangle$$

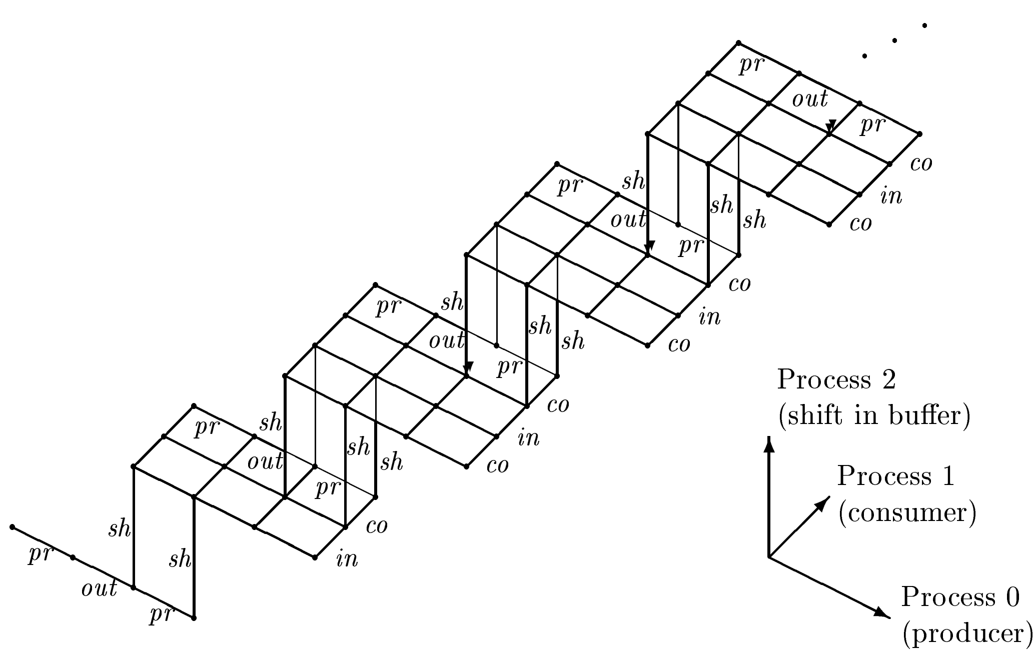
where

$$P^{pc2} \stackrel{\text{def}}{=} \{0, 1, 2\}$$
$$L^{pc2}0 \stackrel{\text{def}}{=} (pr\ out)^\omega \quad L^{pc2}1 \stackrel{\text{def}}{=} (in\ co)^\omega \quad L^{pc2}2 \stackrel{\text{def}}{=} sh^\omega$$

($sh$ — shifting) with the obvious admissible configurations:

$$I^{pc2} \stackrel{\text{def}}{=}$$
$$\left\{ \langle w, v, u \rangle \sqsubseteq \langle (pr\ out)^\omega, (in\ co)^\omega, sh^\omega \rangle \,\middle|\, \begin{array}{l} \sharp_{sh} u \le \sharp_{out} w \le \sharp_{sh} u + 1\ \& \\ \sharp_{in} v \le \sharp_{sh} u \le \sharp_{in} v + 1 \end{array} \right\}$$

In this case, the set of admissible configurations is 3-dimensional:



## 4.6 Two dining philosophers

This is a simplified version of Dijkstra's famous problem of 5 dining philosophers. In this version, there are only two of them facing each other; each other's left fork is his companion's right fork, and each other's right fork is his companion's left fork.

$$\mathcal{V}^{ph2} = \left\langle P^{ph2}, A^{ph2}, L^{ph2}, I^{ph2} \right\rangle$$

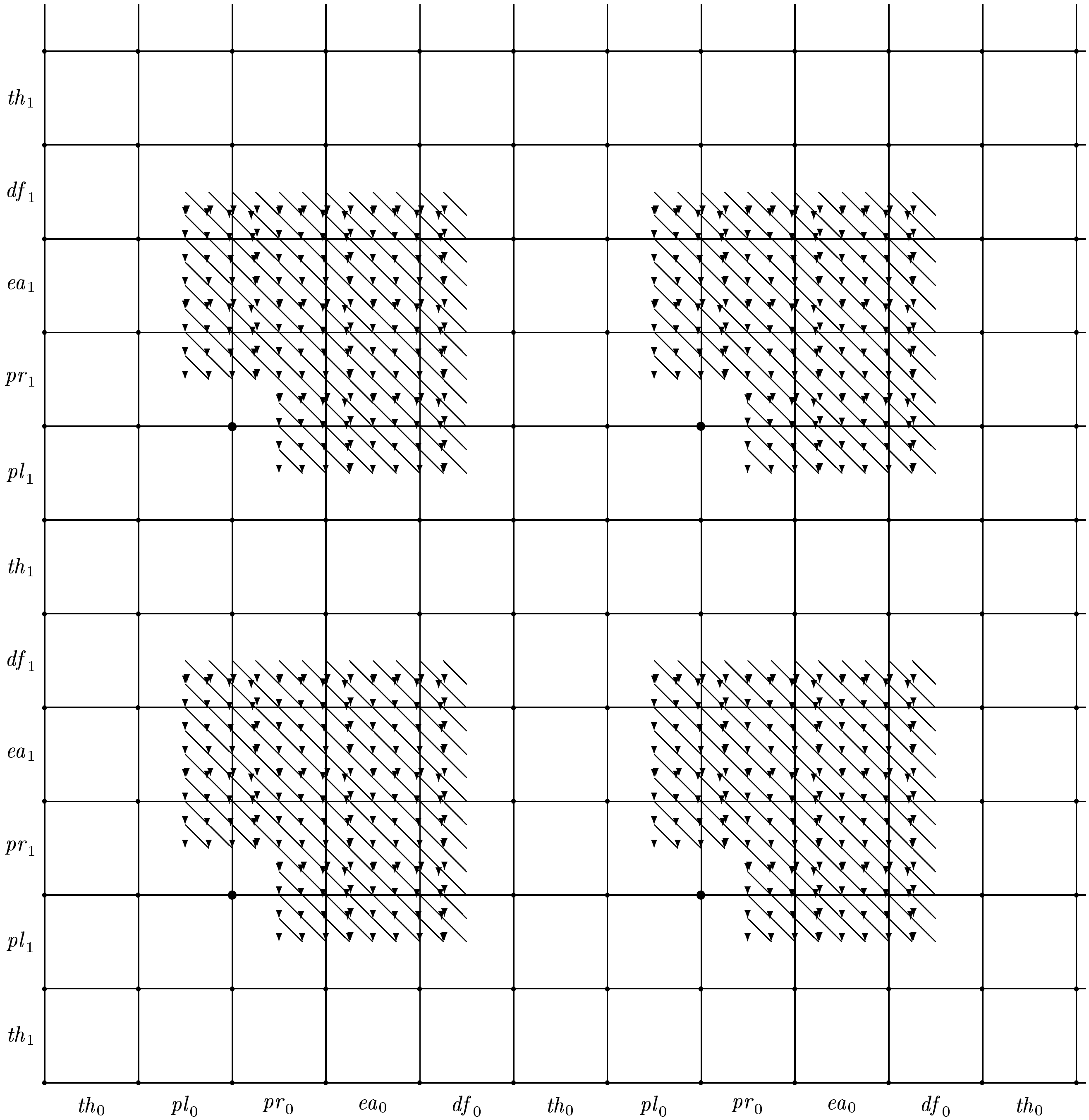
where

$$P^{ph2} \stackrel{\text{def}}{=} \{0,1\}$$
$$L^{ph2} p \stackrel{\text{def}}{=} v_p^{\omega} \quad \text{with} \quad v_p \stackrel{\text{def}}{=} th_p \, pl_p \, pr_p \, ea_p \, df_p \quad \text{for } p \in \{0,1\}$$

($th$ — thinking, $pl$ — picking left fork, $pr$ — picking right fork, $ea$ — eating, $df$ — dropping both forks) with the following admissible configurations:

$$\begin{aligned} I^{ph2} \stackrel{\text{def}}{=} & \\ & Conf_f \, \mathcal{V}^{ph2} \\ & \setminus \bigcup_{n,k \in Nat} \Box \left\langle v_0^n \, v_1^k \, th_0 \, pl_0 \, th_1 \, pl_1 \, , \; v_0^n \, v_1^k \, th_0 \, pl_0 \, pr_0 \, ea_0 \, th_1 \, pl_1 \, pr_1 \, ea_1 \right\rangle \\ & \cup \left\{ \left\langle v_0^n \, th_0 \, pl_0 \, , \; v_1^k \, th_1 \, pl_1 \right\rangle \;\middle|\; n, k \in Nat \right\} \end{aligned}$$

This means that once philosopher $p$ has picked his left fork, he has to drop it before philosopher $1 - p$ may pick his right fork; and the other way round.




Note the fat points at the picture which correspond to deadlocks — if both philosophers have picked their left forks, both will indefinitely wait for right forks.

# Chapter 5

# Homotopy cpo

The homotopy cpo for a given network of processes $\mathcal{V} = \langle P, A, L, I \rangle$ is constructed gradually, as a sequence of functors from category $\underline{\underline{Vect}}$ to category $\underline{\underline{Cpo}}$, each abstracting from a bigger number of secondary details and more cleanly extracting the "genuine nature" of the vector of processes in question.

## 5.1 Vector of processes as path generator

Recall from Sec. 3.1 on page 23 the definition of paths generated by a vector of processes $\mathcal{V} = \langle P, A, L, I \rangle$:

$$Pth\,\mathcal{V} \stackrel{\text{def}}{=} \left\{ w \in A^s \,\middle|\, \forall_{v \in A^*}\, v \sqsubseteq w \Rightarrow endpt\, v \in I \right\}$$

In Sec. 4, paths generated by example vectors were carefully analysed. Now we are going to study the structure of the sets of paths.

**Proposition 32**
*For any vector* $\mathcal{V} = \langle P, A, L, I \rangle$ *of processes, the set* $Pth\,\mathcal{V}$ *with the prefix ordering is a cpo.*

For any vector morphism $\langle f, g \rangle : \mathcal{V} \to \mathcal{V}'$, define the derived mapping on respective path sets:

$$Pth\,\langle f, g \rangle : Pth\,\mathcal{V} \to Pth\,\mathcal{V}'$$
$$Pth\,\langle f, g \rangle \stackrel{\text{def}}{=} \sigma g|_{Pth\,\mathcal{V}}$$

**Proposition 33**
*The definition of* $Pth\,\langle f, g \rangle$ *is correct, i.e.* $\sigma g w \in Pth\,\mathcal{V}'$ *for any* $w \in Pth\,\mathcal{V}$.

Proof of Proposition 33:
This follows from Prop. 25 on page 23, Prop. 23 on page 22 and Cond. 2 from the definition of morphism (see p. 24).
□

As announced, Prop. 33 justifies Cond. 3 in the definition of morphism (page 24).

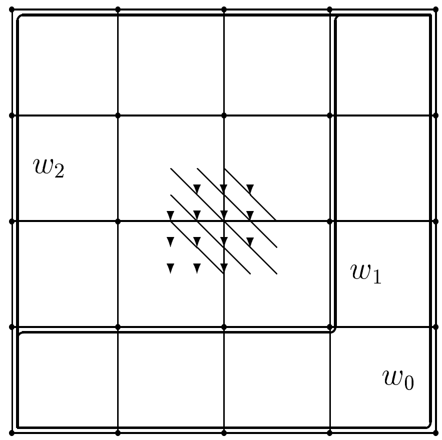
**Proposition 34**
$Pth\,\langle f, g \rangle$ *is a continuous mapping of cpo-s* $Pth\,\mathcal{V}$ *and* $Pth\,\mathcal{V}'$.

**Theorem 5**
$Pth$ *is a functor from category* $\underline{\underline{Vect}}$ *to category* $\underline{\underline{Cpo}}$.

$Pth$ is the first from the promised sequence of vectors.

## 5.2 Homotopy between paths

We want to distinguish between paths joining the same points of the configuration space but going at different sides around the shaded areas at the pictures from Sec. 4 (page 28); on the other hand, we want to identify the paths joining the same points and in the same way avoiding the shaded areas. Recall that the shaded areas are determined by the set of excluded paths of the vector. To explain the intended identifications of paths, let us consider the reader-and-writer example from Sec. 4.1 (page 28):




There are no "holes" between $w_0$ and $w_1$, therefore these two paths will be identified. On the other hand, $w_2$ will stay separate since it avoids the shaded area in a different way.

More precisely, *path homotopy* in a vector $\mathcal{V} = \langle P, A, L, I\rangle$ of processes is the least equivalence relation $H \subseteq \mathit{Pth}\,\mathcal{V} \times \mathit{Pth}\,\mathcal{V}$ such that:

- if

  $$\begin{aligned} w_1 &= w' \frown a_1 a_2 \frown w'' \in \mathit{Pth}\,\mathcal{V} \quad \text{and} \\ w_2 &= w' \frown a_2 a_1 \frown w'' \in \mathit{Pth}\,\mathcal{V} \end{aligned}$$

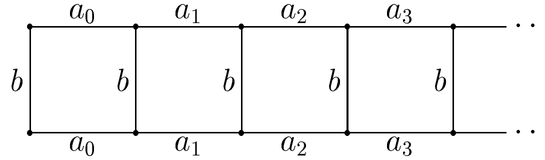
  (i.e. $w_1$ and $w_2$ differ by a transposition of two adjacent actions) then $w_1 \, H \, w_2$;

- the $\omega$-rule holds; i.e. (recall from Sec. 2.1 on p. 13) if $w_0 \sqsubseteq w_1 \sqsubseteq w_2 \sqsubseteq \ldots$ is an ascending chain of paths and $v_0 \, H \, v_1 \, H \, v_2 \, H \ldots$ is a sequence of $H$-related paths such that $w_i \sqsubseteq v_i$ for $i \in \mathit{Nat}$, then there exists a path $u$ such that $w_i \sqsubseteq u$ and $v_i \, H \, u$ for any $i \in \mathit{Nat}$.

The first item in the above definition accounts for gluing together a path with any other path to which it may be brought by a finite number of transpositions of adjacent actions without ever leaving the admissible set $I$. For instance, it implies that the paths

$ba_0a_1a_2\ldots$
$a_0ba_1a_2\ldots$
$a_0a_1ba_2\ldots$
$a_0a_1a_2b\ldots$

in




are $H$-related. The second item in the definition implies that the path

$$a_0a_1a_2\ldots$$

is also $H$-related to them. To see this, set $w_i$ and $v_i$ as follows:

$$w_i \stackrel{\text{def}}{=} a_0 a_1 \ldots a_{i-1} \quad \text{and}$$
$$v_i \stackrel{\text{def}}{=} a_0 a_1 \ldots a_{i-1} b a_{i+1} a_{i+2} \ldots$$

for $i \in \mathit{Nat}$ and check that $w_i \sqsubseteq w_{i+1}$, $w_i \in v_i$ and $v_i \, H \, v_{i+1}$.

In algebraic topology, homotopy is a continuous deformation of mappings; in particular, two paths are homotopic if one can be continuously deformed into the other. This may be perceived as the limit of the above defined notion when the actions are replaced by greater and greater numbers of smaller and smaller actions.

**Proposition 35**

*Equivalence $H$ is $\omega$-consistent with the prefix partial order $\sqsubseteq$ (cf. Sec. 2.1 on page 13).*

**Proposition 36**

*Let $\mathcal{V} = \langle P, A, L, I \rangle$ be a vector of processes and assume that $w, v \in A^*$ are finite words such that*

- *$w, w \frown v \in \mathit{Pth}\,\mathcal{V}$, and*
- *$\Box \langle w, w \frown v \rangle \subseteq I$, i.e. the cube between $w$ and $v$ does not contain any obstacles.*

*Let $u \in A^*$ be another finite word such that $\mathit{endpt}\, u = \mathit{endpt}\, v$. Then $w \frown u \in \mathit{Pth}\,\mathcal{V}$ and*

$$(w \frown u) \, H \, (w \frown v)$$

Denote by $\mathit{Htp}\,\mathcal{V}$ the quotient set of paths generated by the path homotopy $H$:

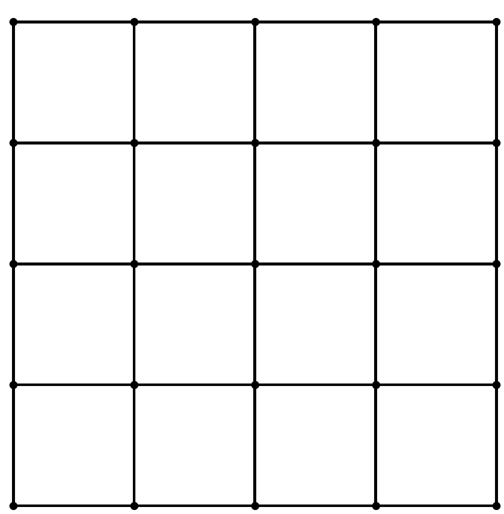
$$\mathit{Htp}\,\mathcal{V} \stackrel{\text{def}}{=} \mathit{Pth}\,\mathcal{V}/H$$

and let $\sqsubseteq$ denote the relation in $\mathit{Htp}\,\mathcal{V}$ defined by

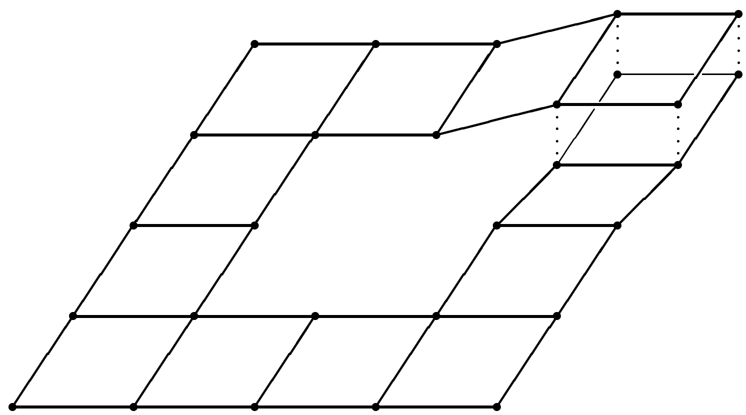
$$[w_1]_H \sqsubseteq [w_2]_H \stackrel{\text{def}}{\Longleftrightarrow} \exists_v \, w_1 \sqsubseteq v \, H \, w_2$$

By Prop. 35 and Prop. 16 (page 15), $\langle \mathit{Htp}\,\mathcal{V}, [\varepsilon]_H, \sqsubseteq \rangle$ is a cpo. It is called *path homotopy cpo* of vector of processes $\mathcal{V}$.

In the uninteresting case, when the vector of processes does not contain any “holes”, i.e. when $I = \mathit{Conf}_f\,\mathcal{V}$, the paths that end in the same point are glued together. For instance, if $\mathcal{V}^{rw0}$ denotes the reader-and-writer example from Sec. 4.1 on page 28 but with $I^{rw0} \stackrel{\text{def}}{=} \mathit{Conf}_f\,\mathcal{V}^{rw0} = \mathit{Conf}\,\mathcal{V}^{rw0}$, then $\mathit{Htp}\,\mathcal{V}^{rw0}$ is isomorphic to the configuration space $\mathit{Conf}\,\mathcal{V}^{rw0}$:



with the ordering from left to right and from bottom upwards. A similar picture corresponds to $Htp\,\mathcal{V}$ in the case of non-trivial $I$, only it splits behind the holes. So $Htp\,\mathcal{V}^{rw}$ is:



With more complex structure of excluded regions, the splitting becomes more complex too.

Mapping $Pth\,\langle f, g\rangle : Pth\,\mathcal{V} \to Pth\,\mathcal{V}'$ induces a corresponding mapping on respective path homotopy cpo-s:

$$Htp\,\langle f, g\rangle : Htp\,\mathcal{V} \to Htp\,\mathcal{V}'$$
$$Htp\,\langle f, g\rangle [w]_H \stackrel{\text{def}}{=} [Pth\,\langle f, g\rangle w]_H$$

**Proposition 37**
*Mapping* $Htp\,\langle f, g\rangle$ *is well-defined, i.e.*

$$w\,H\,v \Rightarrow (Pth\,\langle f, g\rangle w)\,H\,(Pth\,\langle f, g\rangle v)$$

Proof of Proposition 37:
This follows from Prop. 36 (page 37) and from Cond. 2 in the definition of vector homomorphism (page 24).
□

**Proposition 38**
$Htp\,\langle f, g\rangle : Htp\,\mathcal{V} \to Htp\,\mathcal{V}'$ *is a continuous mapping of cpo-s.*

Proof of Proposition 38:
From Prop. 35 on page 37, Prop. 37, and Prop. 18 on page 17.
□

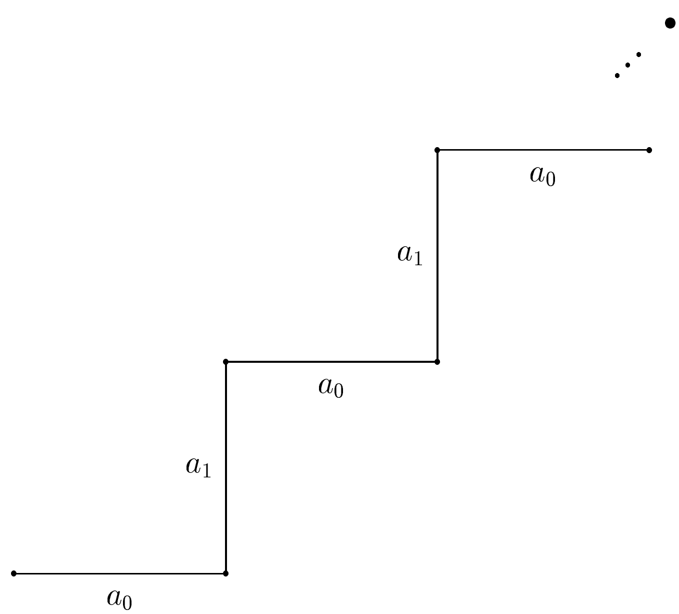
**Theorem 6**
*Htp* *is a functor from category* $\underline{\underline{Vect}}$ *to category* $\underline{\underline{Cpo}}$.

*Htp* is the second functor promised.

## 5.3 Path homotopy cpo-s for some example vectors of processes

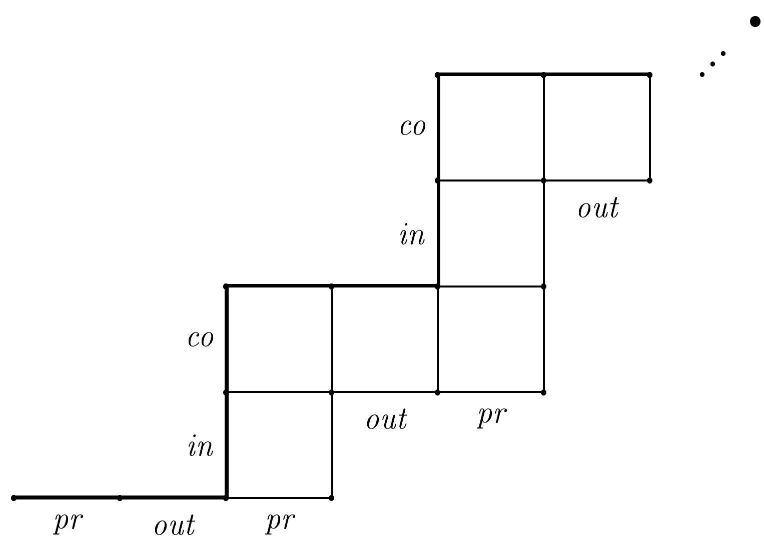
The path homotopy cpo for the reader-and-writer example from Sec. 4.1 (page 28) has already been presented in Sec. 5.1.

For the complete-sequentialization example from Sec. 4.2 (page 29), *Htp* $\mathcal{V}^{cs}$ is linear with an extra point corresponding to the infinite path:



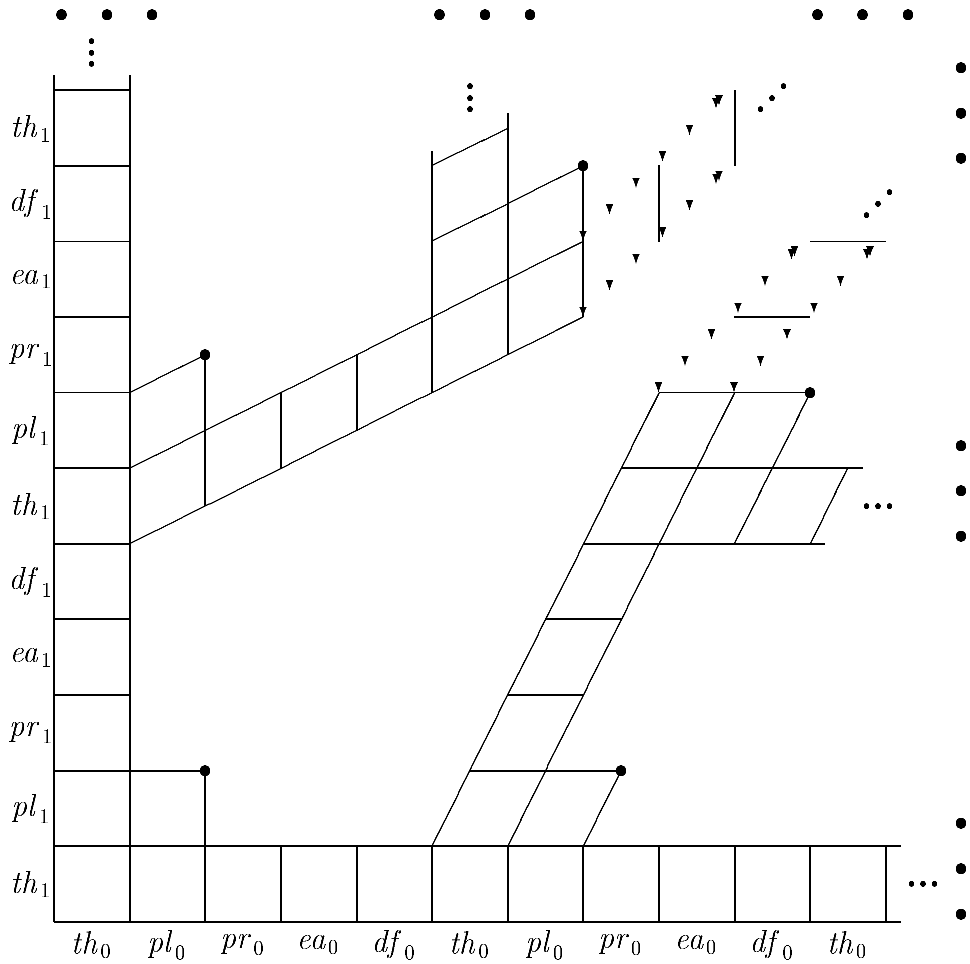

(ordered, as usual, from left to right and from bottom upwards).

The path homotopy cpo *Htp* $\mathcal{V}^{pc1}$ for producer and consumer with a 1-place buffer from Sec. 4.3 (page 30) is very similar:




The thick line on the picture above corresponds to the image of *Htp* $\mathcal{V}^{cs}$ in *Htp* $\mathcal{V}^{pc1}$ via the derived mapping *Htp* $\langle f_i, g_i \rangle$, where $\langle f_i, g_i \rangle$ implements $\mathcal{V}^{cs}$ in $\mathcal{V}^{pc1}$ as described in Sec. 4.4 (page 31). Nothing unexpected so far.

In the infinitary philosophers example from Sec. 4.6 (page 33), the picture will be split in infinitely many branches, behind every "hole". Thus, $Htp\,\mathcal{V}^{ph2}$ is:




Note the maximal elements corresponding to finite paths — evidently, the deadlock points have been preserved by functor $Htp$. Note also the boundary elements corresponding to infinite paths.

## 5.4 Homotopy between grid points

Recall from Sec. 5.3 (page 39) the similarity between $Htp\,\mathcal{V}^{cs}$ — the path homotopy cpo for the complete sequentialization example — and $Htp\,\mathcal{V}^{pc1}$ — the path homotopy cpo for the producer and consumer example. The latter cpo has more elements because:

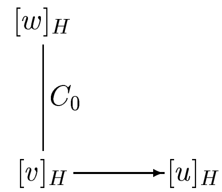
- any action in $\mathcal{V}^{cs}$ has been implemented as a pair of (auxiliary) actions in $\mathcal{V}^{pc1}$, and
- there are extra configurations in $\mathcal{V}^{pc1}$ due to the possibility of inessential transpositions of auxiliary actions.

This difference is factored out in the quotient of $Htp\,\mathcal{V}$ by the maximal contraction (cf. Sec. 2.2, page 18). As demonstrated by the example on page 20, such a quotient may, in general, fail to be a cpo. Therefore, we begin by showing that the maximal contractions in our particular path homotopy cpo-s are $\omega$-consistent with the partial order. This will clear the way for the application of Prop. 16 (page 15) and for guaranteeing the cpo properties for the quotient.

In accordance with Sec. 2.2 (page 18), denote the maximal contraction in *Htp* $\mathcal{V}$ by $C_0$.

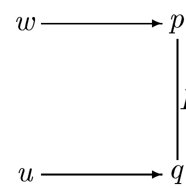
**Proposition 39**
*Assume equivalence classes* $[w]_H, [v]_H, [u]_H \in Htp\,\mathcal{V}$ *satisfy the condition*

$[w]_H \, C_0 \, [v]_H$ and $v \sqsubseteq u$



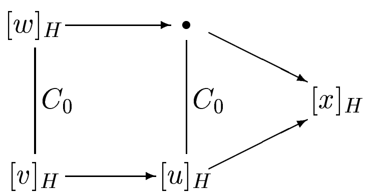
*Then there exist* $p, q \in Pth\,\mathcal{V}$ *such that*

$w \sqsubseteq p$, $u \sqsubseteq q$ *and* $p \, H \, q$

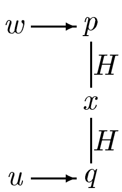


Proof of Proposition 39:

By Cond. 3 and Cond. 1 from the definition of contraction (page 17), a following completion of the initial diagram may be performed:



By the definition of the partial order in *Htp* $\mathcal{V}$, see page 37, the inequalities $[w]_H \sqsubseteq [x]_H$ and $[u]_H \sqsubseteq [x]_H$ translate to

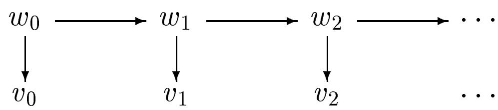


for some $p, q \in Pth\,\mathcal{V}$.
□

**Theorem 7**
*The maximal contraction* $C_0$ *in* *Htp* $\mathcal{V}$ *is* $\omega$*-consistent with its partial order* $\sqsubseteq$.

Proof of Theorem 7:

Assume $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots$ is an ascending chain in $\mathit{Htp}\,\mathcal{V}$ and $y_0 \, C_0 \, y_1 \, C_0 \, y_2 \, C_0 \, \ldots$ is a sequence of $C_0$-related elements of $\mathit{Htp}\,\mathcal{V}$ such that $x_i \sqsubseteq y_i$ for $i \in \mathit{Nat}$. Pick such respective $w_0 \sqsubseteq w_1 \sqsubseteq w_2 \sqsubseteq \ldots$ and $v_0 \sqsubseteq v_1 \sqsubseteq v_2 \sqsubseteq \ldots$ in $\mathit{Pth}\,\mathcal{V}$ that $[w_i]_H = x_i$ and $[v_i]_H = y_i$ for $i \in \mathit{Nat}$ and



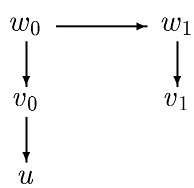
Let $\bar{w} \stackrel{\text{def}}{=} \bigsqcup_{i \in \mathit{Nat}} w_i$. The only case of interest is when $\bar{w}$ is an infinite path: $\sharp\bar{w} = \omega$.

Once we have proven that $[v_0]_H \, C_0 \, [\bar{w}]_H$, the $\omega$-consistency will follow (recall that classes $[v_i]_H$ are $C_0$-related to each other). Define a new relation in $\mathit{Htp}\,\mathcal{V}$:

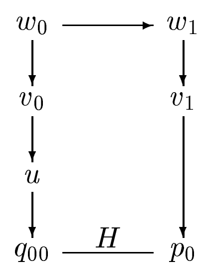
$$\bar{C} \stackrel{\text{def}}{=} \left\{ \langle [u]_H, [\bar{w}]_H \rangle \,\middle|\, v_0 \sqsubseteq u \right\}$$

As usual, we are going to demonstrate that $\bar{C}$ is a contraction. Take an arbitrary $u$ such that $v_0 \sqsubseteq u$.

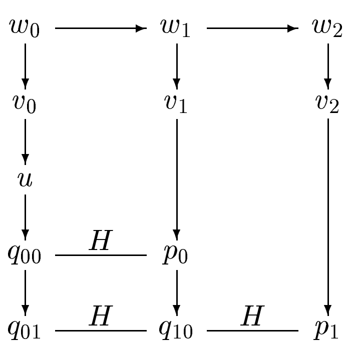
Since $[v_1]_H$ and $[v_0]_H$ are $C_0$-related, by virtue of Prop. 39 the picture



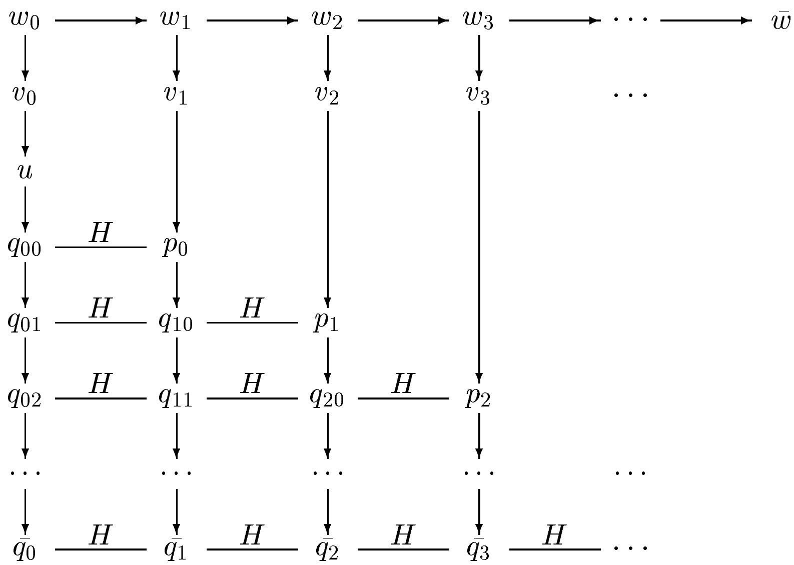
may be completed to



Since $[v_2]_H$ and $[v_1]_H$ are $C_0$-related, by virtue of Prop. 39 and by the consistency of $H$ with $\sqsubseteq$, it may further be completed to:



Applying the same procedure again and again, we get the infinite picture




where $\bar{q}_i \stackrel{\text{def}}{=} \bigsqcup_{j \in Nat} q_{ij}$ for $i \in Nat$. This way, every $v_i$ gives rise to an ascending chain $q_{i0} \sqsubseteq q_{i1} \sqsubseteq q_{i2} \sqsubseteq \ldots$ (vertical sequences at the picture). The lubs $\bar{q}_i$ of these chains are $H$-related to each other (cf. Prop. 17 on page 16), therefore, by the definition of homotopy on page 36, $\bar{w}\, H\, \bar{q}_0$. This implies $[u]_H \sqsubseteq [\bar{w}]_H$ and hence Conditions 1 and 3 from the definition of contraction on page 17 have been verified for $\bar{C}$.

To verify Condition 2, assume $u \sqsubseteq t$. The whole construction of ascending chains over $v_i$-s may now be repeated with $t$ for $u$ resulting in $[t]_H \sqsubseteq [\bar{w}]_H$. Since $[t]_H\, \bar{C}\, [\bar{w}]_H$, Cond. 2 is satisfied. Thus $\bar{C}$ is a contraction and, as such, it is contained in $C_0$; since $[v_0]_H\, \bar{C}\, [\bar{w}]_H$ then also $[v_0]_H\, C_0\, [\bar{w}]_H$. This completes the proof.
□

**Proposition 40**
*The quotient* $Htp\, \mathcal{V}/C_0$ *has a natural cpo structure inherited from* $Htp\, \mathcal{V}$*.*

This quotient

$$Fdm\, \mathcal{V} \stackrel{\text{def}}{=} Htp\, \mathcal{V}/C_0$$

is called *point homotopy cpo*, or *fundamental cpo*, of $\mathcal{V}$.

Mapping $Htp\, \langle f, g\rangle : Htp\, \mathcal{V} \to Htp\, \mathcal{V}'$ induces a corresponding mapping on respective point homotopy cpo-s:

$$Fdm\, \langle f, g\rangle : Fdm\, \mathcal{V} \to Fdm\, \mathcal{V}'$$
$$Fdm\, \langle f, g\rangle [[w]_H]_{C_0} \stackrel{\text{def}}{=} [Htp\, \langle f, g\rangle [w]_H]_{C_0'}$$

**Proposition 41**
*Mapping* $Fdm\, \langle f, g\rangle$ *is well defined, i.e.*

$$[w_1]_H\, C_0\, [w_2]_H \;\Rightarrow\; (Htp\, \langle f, g\rangle [w_1]_H)\, C_0'\, (Htp\, \langle f, g\rangle [w_2]_H)$$

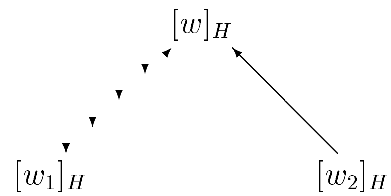
Proof of Proposition 41:

Define the following relation in $Htp\, \mathcal{V}'$:

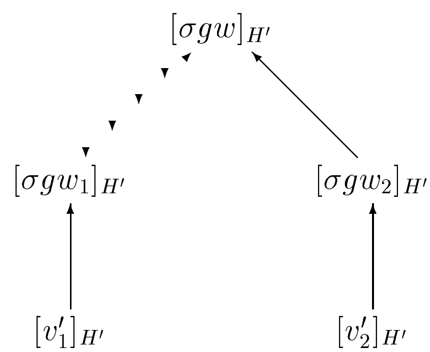
$$\bar{C}' \stackrel{\text{def}}{=} \left\{ \langle [v_1']_{H'}, [v_2']_{H'} \rangle \;\middle|\; \exists_{w_1, w_2 \in Pth\, \mathcal{V}}\ [w_1]_H\, C_0\, [w_2]_H \ \&\ v_1' \ll_g w_1 \ \&\ v_2' \ll_g w_2 \right\}$$

(see the definition of $g$-closeness relation $\ll_g$ on page 22). Thus, $\bar{C}'$ relates the classes of those $g$-images of paths from $Pth\,\mathcal{V}$ which are $C_0$-related; and also the classes, which are $g$-close to them.

Take two arbitrary $\bar{C}'$-related classes $[v_1']_{H'}$ and $[v_2']_{H'}$ and their respective $w_1$ and $w_2$ as in the definition of $\bar{C}'$. Since $[w_1]_H\, C_0\, [w_2]_H$ and $C_0$ is a contraction, there exists a common upper bound

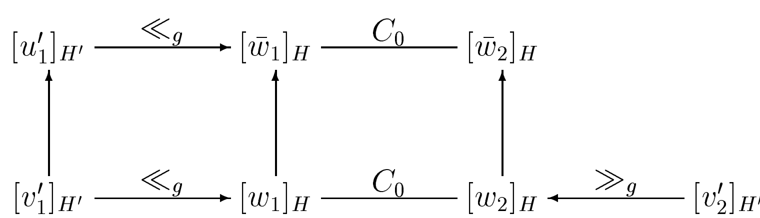



By the monotonicity of $Htp\,\langle f, g\rangle$ (Prop. 38 on page 38), this implies




which demonstrates that any two $\bar{C}'$-related classes have a common upper bound.

Now assume $v_1' \sqsubseteq u_1'$ for a certain $u_1' \in Pth\,\mathcal{V}'$ and try to find an upper bound $[u_2']_{H'}$ of $[v_2']_{H'}$ which is $\bar{C}'$-related to $[u_1']_{H'}$. By virtue of Cond. 3 in the definition of vector morphism (page 24), there exists a $\mathcal{V}$-path $\bar{w}_1$ such that $w_1 \sqsubseteq \bar{w}_1$ and $u_1' \ll_g \bar{w}_1$. On the other hand, since $[w_1]_H$ is $C_0$-related to $[w_2]_H$, there exists an upper bound $[\bar{w}_2]_H$ of $[w_2]_H$ which is $C_0$-related to $[\bar{w}_1]_H$:




Therefore

$[u_1']_{H'}\, \bar{C}'\, [\sigma g \bar{w}_2]_{H'}$ (from Prop. 24 on p. 23) and
$[v_2']_{H'} \sqsubseteq [\sigma g \bar{w}_2]_{H'}$

It has thus been proven that $\bar{C}'$ is a contraction. As such, it is contained in $C_0'$. Therefore, $[w_1]_H\, C_0\, [w_2]_H$ implies

$$\Big\langle Htp\,\langle f, g\rangle [w_1]_H\,,\; Htp\,\langle f, g\rangle [w_2]_H \Big\rangle \;\in\; \bar{C}' \;\subseteq\; C_0'$$

and this completes the proof.
□

**Proposition 42**
*$Fdm\,\langle f,g\rangle : Fdm\,\mathcal{V} \to Fdm\,\mathcal{V}'$ is a continuous mapping of cpo-s.*

Proof of Proposition 42:
From Thm. 7 on page 41, Prop. 41 above, and Prop. 18 on page 17.
□

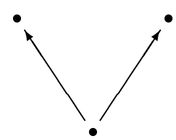
**Theorem 8**
*$Fdm$ is a functor from category $\underline{\underline{Vect}}$ to category $\underline{\underline{Cpo}}$.*

$Fdm$ is the third promised functor in the row.

## 5.5 Point homotopy cpo-s for example vectors of processes

Point homotopy cpo-s for the example vectors of processes from Sec. 4 (page 28) are as follows:

- $Fdm\,\mathcal{V}^{rw}$ :

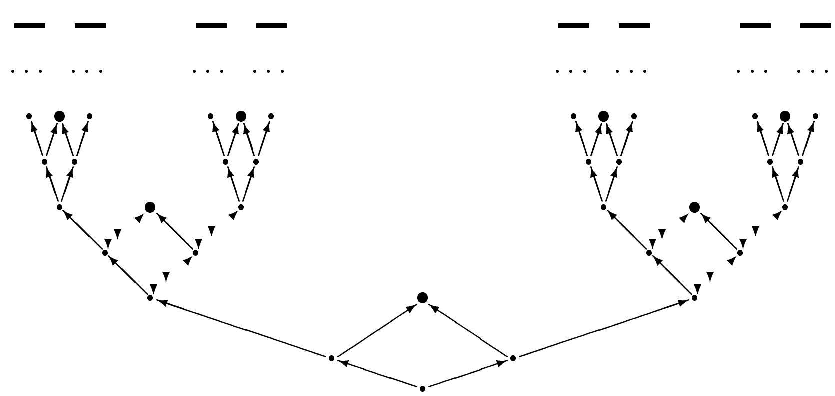
  

  $Htp\,\mathcal{V}^{rw}$ (page 38) has two maximal points; and each of them is $C_0$-related to all points on the branches leading to them. There remain a couple of points, from which both maximal points still can be reached — they are $C_0$-related to each other.

- $Fdm\,\mathcal{V}^{cs}$, $Fdm\,\mathcal{V}^{pc1}$, $Fdm\,\mathcal{V}^{pc2}$ :

  — a single point.

- $Fdm\,\mathcal{V}^{ph2}$ :

  — infinite binary directed acyclic graph with finitary maximal elements corresponding to deadlocks (fat points) and with the Cantor set of infinitary maximal elements[1].

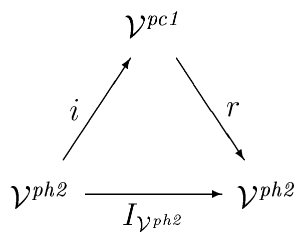
[1]Indeed, every infinite path may be described by an infinite $\{0,2\}$-word with 0 meaning “go up” and 2 meaning “go right”; and this word may serve as the fraction part of a ternary real number whose integer part is 0 (this means: write 0, write dot, write the infinite word, and interpret the whole as a real positional number in base 3) — this is exactly the Cantor set.

## 5.6 Application: non-existence of an implementation

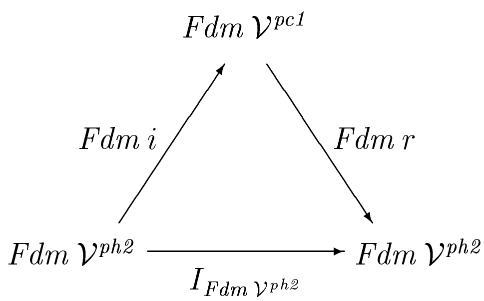
The above considerations allow us to provide the negative answer to the question:

> Does there exist an implementation of $\mathcal{V}^{ph2}$ in $\mathcal{V}^{pc1}$?

Indeed, assume that




is an implementation (cf. Sec. 3.3, page 26). By the functoriality of $Fdm$ (Thm. 8), this commutative diagram of vectors of processes gives rise to the following commutative diagram of cpo-s:




But this means inserting a whole complex cpo $Fdm\,\mathcal{V}^{ph2}$ via $Fdm\,i$ into $Fdm\,\mathcal{V}^{pc1}$ which is a single point and then processing this single point by $Fdm\,r$ to obtain the whole complex cpo $Fdm\,\mathcal{V}^{ph2}$ again. This is not possible.

# Chapter 6

# Future research

## Other models of concurrency

The definition of vector of processes from Sec. 3.1 on page 21 may seem too restrictive. The software $L$ allowed consists of unstructured straight line programs $Lp$ only, without as much as conditional branching, let alone loops or nondeterminism. Đura Paunić has pointed out that imposing regularity requirements on programs might result in interesting configuration spaces of varying shapes. This observation certainly deserves a second thought.

Marek Bednarczyk has told me how he thinks the approach might be generalized to *Place/Transition* Petri nets with multiple dots in states. These nets may be embedded in (a generalisation of) the category of asynchronous transition systems, and, as argued in [1], *Mazurkiewicz traces* [10] may be regarded as their computations. Marek's idea is that the path homotopy classes generalize trace equivalence of paths as defined by Mazurkiewicz. After some (not too many) technical discussions, we both agreed that this direction should be pursued.

No effort has yet been undertaken to compare my underlying model of concurrency with those used in the database community; for instance, with Pratt's [11], with Goubault and Jensen's [3] or with the one used by Herlihy and Rajsbaum [6] in their applications of simplicial complexes. Surely, this has to be done.

So far I have only told you what has to be checked before the approach is brought in order and before it may be considered robust and mature. Whereas some points of this may prove difficult and require ingenuity, one can still think of the outcome as having but a clerical importance. Now the time has come to let free spirit fly high in figuring glorious prospects.

## Exploitation of the technique

One profit of the homotopy considerations is the classification of processes: two processes, whose fundamental cpo-s are different, differ. This reminds of a classification of topological manifolds by the way of Euler numbers, or of homotopy groups, or of homology groups.

In this paper, I have however shown a subtler application of the approach: the non-existence of a certain implementation (cf. Sec. 5.6 p. 46). This reminds of a classical proof using fundamental groups and their homomorphisms, that there is no retraction of a solid circle onto its circumference. In analogy with that topological proof, I have made full use of the functoriality of the fundamental cpo.

Somewhere, in handbooks of algebraic topology, rests a wealth of other similar proof techniques. A researcher's conscience cannot be clear before their applicability to processes is examined.

## Composition of homotopy classes of paths

Marek Bednarczyk thinks, there may after all exist a way to compose homotopy classes. This will not provide the fundamental cpo with as much as a group structure, rather with that of a small category. The idea is to consider *localized* homotopies — the family of homotopy relations one for each pair $\langle w, w^\frown v\rangle$ of paths in $\mathit{Pth}\,\mathcal{V}$.

## Higher-dimensional homotopy structures

A server with two printers attached may satisfy at most two printing requests in the same time. More generally, one might want to study a generalization of critical regions, accessible to at most $k$ processes at once. The "holes" in the configuration space are then higher, $(k+1)$-dimensional and require $k$-dimensional homotopy structures (groups? cpo-s? something else?) for investigation. Please, let me know if you have any results on this.

# Bibliography


[1] M.A. Bednarczyk. *Categories of Asynchronous Systems*. Ph.D. thesis, Sussex University, CST 1/88, 1988.

[2] R. van Glabbeek. Bisimulations for higher dimensional automata. E-mail communication available from `http://theory.stanford.edu/~rvg/hda`, Stanford, 1991.

[3] É. Goubault, T. Jensen. Homology of higher-dimensional automata. *CONCUR 92 — Proc. of the 3rd International Conference on Concurrency Theory, Stonybrook, New York*, 1992.

[4] J. Gunawardena. Homotopy and concurrency. *Bulletin of the EATCS*, Vol. 54, pp. 184–193, October 1994.

[5] M. Herlihy. *A Tutorial on Algebraic Topology and Distributed Computation*, Brown University, Providence, USA, Version of October 1997.

[6] M. Herlihy, S. Rajsbaum. Algebraic spans. *Proc. of the 14th Annual ACM Symp. on Principles of Distributed Systems*, Ottawa, pp. 90–99, August 1995.

[7] M. Herlihy, S. Rajsbaum. Algebraic topology and distributed computing. A primer. *Computer Science Today: Recent Trends and Developments*, LNCS 1000, pp. 203–217, 1995.

[8] M. Herlihy, N. Shavit. *The Topological Structure of Asynchronous Computability*. Brown University, Providence, USA, January 1996.

[9] G. Hoest, N. Shavit. Towards a topological characterization of asynchronous complexity. *Proc. of the 16th ACM Symp. on Principles of Distributed Computing*, Santa Barbara, California, USA, August 1997.

[10] A. Mazurkiewicz. Concurrent program schemes and their interpretations. Report DAIMI PB-78, Dept. of Computer Science, University of Århus, 1977.

[11] V. Pratt. Modeling concurrency with geometry. *Proc. of the 18th Annual ACM Symp. on Principles of Programming Languages*, pp. 311–322, January 1991.