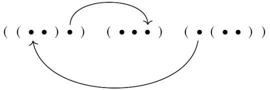
On the Order Theory of Unlabelled Transition Systems

Peter M. Hines — University of York

GT-DAAL Meeting

Rennes – April 2024



Au fond de l'Inconnu pour trouver du nouveau!

Objective of the talk(!)

We revisit some (not-so-recent) Theoretical Computer Science, based on G. Plotkin's *Unlabelled Transition Systems*.

- *"Machine Semantics"* — Theoretical Computer Science (PMH)
- *"From Causality to Computational Models"*
  — Int. J. of Unconventional Comp. (PMH)

Originally developed to resolve an outstanding problem in QM computation

- *"Quantum Circuits for Abstract Machine Computations"*
  — Theoretical Computer Science (PMH)

but ended up being applied more widely; particularly, to explain why we see structures from *Linear Logic* & *Geometry of Interaction* in models of *automata* & *state machines*.

- *"A categorical framework for finite state machines"*
  — Mathematical Structures in Computer Science (PMH)

Motivation :

Currently being revisited for further applications . . . in several C.S. - related topics

The simplest model of computation

An (unlabelled) **Transition System** is simply a set X of **configurations**, together with a discrete notion of causality.

Given a configuration $x \in X$, we can find the *'next configuration(s)'*.

determinism $x \mapsto x'$

partiality $x \mapsto \{\}$

non-determinism $x \mapsto \{x'_1, x'_2, x'_3\}$

> *"Of course, this idea is hardly new, and examples can be found in any book on automata, formal languages, or programming languages "*
> A Structural Approach to Operational Semantics — *G. Plotkin*

S.A.O.S. Aarhus Tech. Report (1981)

"This is too general a notion to produce any useful theory "

The simplest model of computation

An (unlabelled) **Transition System** is simply a set X of **configurations**, together with a discrete notion of causality.

Given a configuration $x \in X$, we can find the *'next configuration(s)'*.

determinism $x \mapsto x'$

partiality $x \mapsto \{\}$

non-determinism $x \mapsto \{x'_1, x'_2, x'_3\}$

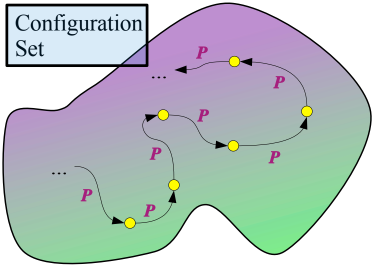
> *"Of course, this idea is hardly new, and examples can be found in any book on automata, formal languages, or programming languages "*
>
> A Structural Approach to Operational Semantics — *G. Plotkin*

S.A.O.S. Published version (2004)

~~"This is too general a notion to produce any useful theory "~~

An inessential simplification

We consider the case with *partial, deterministic* dynamics



A (countable) set X of **configurations**.

A (partial) **transition function**, $\mathcal{P} : X \to X$

(also known as the **primitive evolution**).

Partiality gives, for free, notions of start / halt configurations :

Starting config.s These satisfy $\mathcal{P}^{-1}(s) = \{\}$.

Halting config.s These satisfy $\mathcal{P}(h) = \{\}$.

Another relevant quote . . .

Again, from 'A Structural Approach to Operational Semantics' :

> *"We ignore the fact that transitions only make sense at a certain level. What counts as a* **single transition** *may consist of* **many steps** *when viewed in more detail.*
>
> It is a *matter of experience* to choose the right definition."

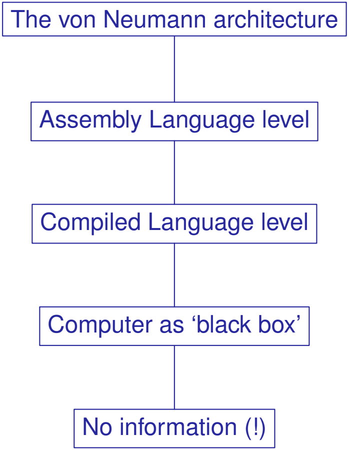
Question :

Given transition systems $(X, \mathcal{P})$ and $(X, \mathcal{Q})$ on the same configuration set, when do they, *"describe the same computation at different levels of detail"* ?

This is what we axiomatise, in order-theoretic terms.

The high-level vs. Low-level comparison

In his book '*Gödel, Escher, Bach*', D. Hofstadter describes what he calls "chunking", with the following example :



We intuitively think of "low-level" vs. "high-level" as a form of ordering!

The same computation, described at different levels of generality.

Basic definitions

Given a *partial function* $\eta : X \to X$, its **transitive closure** is the *relation*

$$\eta^+ \stackrel{def.}{=} \bigcup_{j=1}^{\infty} \eta^j$$

The key definition

Given transition systems (X, η) and (X, μ) we say that

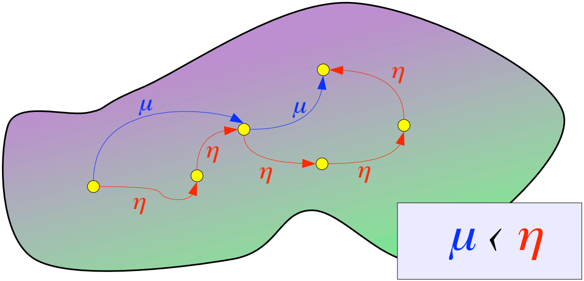
η is a **refinement** of μ when $\mu \subseteq \eta^+ = \bigcup_{j=1}^{\infty} \eta^j$.

Operationally : η is a refinement of μ when, for all configurations $x \in X$,

$$\mu(x) = y \quad \Rightarrow \quad \exists\, N_x > 0 \;\; s.t. \;\; \eta^{N_x}(x) = y$$

Notation We suggestively write this as $\mu \prec \eta$.

Decomposing steps of transition systems



"Every single transition of (X, μ) *may be decomposed into multiple transitions of* (X, η)*."*

Trivially, every partial function is a refinement of the nowhere-defined partial function $\perp(x) = \{\}$.

Different views of the same computation

Given a transition system $(X, \mathcal{P})$, its **machine semantics** is the set of partial functions refined by $\mathcal{P}$.

$$[\mathcal{P}] \stackrel{def.}{=} \{\eta : X \to X \text{ such that } \eta \prec \mathcal{P}\}$$

— all **representations**, or 'partial descriptions' of the behaviour of $(X, \mathcal{P})$.

This is closed under composition of partial functions, and hence a semigroup with a zero $\perp$.

Additional properties

Reflexivity $\eta \prec \eta$

Transitivity $\nu \prec \mu$ and $\mu \prec \eta$ implies $\nu \prec \eta$

Two Questions :

1. Do we have a *partial order* ?
2. What is the relationship between *composition* and *refinement* ?

Ceci n'est pas un poset . . .

We do not have the *symmetry* property

$$\eta < \mu \text{ and } \mu < \eta \text{ does not imply } \eta = \mu$$

The relation $<$ is a preorder, but not a partial order.

Counterexample : the group $\mathbb{Z}_3$

Configuration set $X = \{0, 1, 2\}$

$$\eta(x) = x + 1 \mod 3 \quad , \quad \mu(x) = x + 2 \mod 3$$

The obstacle to $<$ being a partial order is *cyclic behaviour*.

Even worse, the usual preorder $\rightarrow$ partial order quotient commonly causes a collapse to triviality.

Cycle-freeness

Definitions : A transition system (X, η) is **cycle-free** iff, for all $x \in X$

$$\eta^K(x) \neq x \quad \forall K > 0$$

When $|X| < \infty$ this is simply *nilpotency* of η.

For an arbitrary transition system, define its **cycle-free semantics** $[\![\mathcal{P}]\!] \subseteq [\mathcal{P}]$ by

$$[\![\mathcal{P}]\!] \stackrel{def.}{=} \{\eta < \mathcal{P} \ : \ (X, \eta) \text{ is cycle-free}\}$$

— the collection of partial descriptions of $(X, \mathcal{P})$ that, *"do not see cyclic behaviour"*.

Refinement then becomes a partial order

$([\![\mathcal{P}]\!], <)$ is then a poset (but not necessarily a semigroup . . .) with bottom element $\bot$.

The structure of cycle-free semantics

When $(X, \mathcal{P})$ is itself cycle-free, its *cycle-free semantics* $[\![\mathcal{P}]\!]$ has a particularly neat form.

- Cycle-free semantics and machine semantics coincide $[\![\mathcal{P}]\!] = [\mathcal{P}]$.
- $[\![\mathcal{P}]\!]$ is a semigroup with a partial order.
  — but order & composition are *not* compatible!
- There is a top element $P : X \to X$
- and a bottom element $\perp : X \to X$
- We have closure under **finite meets** and **arbitrary joins**
  — explicit formulæ to follow!

We arrive at a *locale* or *pointless topology*.

A special form of Heyting algebra, or Lindenbaum-Tarski algebra of an intuitionistic logic

Consider some **finite** subset $\{\eta_j\}_{j\in J} \subseteq [\![\mathcal{P}]\!]$.

Assume (w.l.o.g.) that each of these is defined at some configuration $x \in X$, so $\eta_j(x)$ exists, for all $j \in J$.

For each η_j, there exists a **unique** $N_j > 0$ such that $\eta_j(x) = \mathcal{P}^{N_j}(x)$.

The meet is defined in terms of the **least common multiple** $L_x = lcm(\{N_j\}_{j\in J})$ of these unique natural numbers:

$$\left(\bigwedge_{j\in J} \eta_j \right)(x) \; = \; \mathcal{P}^{L_x}(x)$$

As the set $\{N_i\}_{j\in J} \subseteq \mathbb{N}$ is **finite**, this least common multiple exists.

Joins in the cycle-free semantics

Alternatively, consider an **arbitrary** subset $\{\mu_k\}_{k\in K} \subseteq [\![\mathcal{P}]\!]$.

Consider some configuration $x \in X$ such that (again w.l.o.g.) $\mu_k(x)$ exists for all $k \in K$.
For each μ_k, there exists a **unique** $M_k \in \mathbb{N}$ satisfying $\mu_k(x) = \mathcal{P}^{M_k}(x)$.

The join is defined in terms of the *greatest common divisor* of
the countable set $G_x = gcd(\{M_k\}_{k\in K})$

$$\left(\bigvee_{k\in K} \mu_k\right)(x) = \mathcal{P}^{G_x}(x)$$

The greatest common divisor of $\{M_k\} \subseteq \mathbb{N}$ always exists.

What about the general case ?

Consider the cycle-free semantics of an arbitrary transition system $(X, \mathcal{P})$.

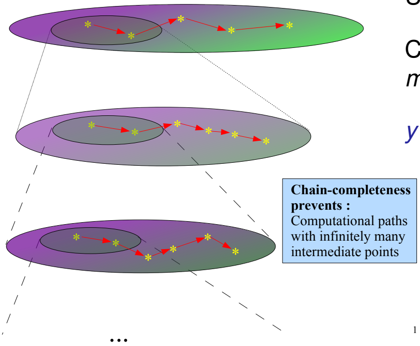
1. $((\llbracket \mathcal{P} \rrbracket, \prec)$ is a poset, with bottom element the nowhere-defined function $\perp$
2. The down-closure of every element is a locale.
3. It does not, in general, contain $\mathcal{P}$ itself — there is no top element.
4. It is **directed-complete**

   – at least, assuming the axiom of choice.
5. It has **compact** or **finitary** elements
   - these are the partial functions with **finite support**.
6. Every element is the supremum of a chain of finitary elements.

Directed-completeness as Zeno's paradox ?

Consider an infinite chain $\eta_0 \prec \eta_1 \prec \eta_2 \prec \eta_3 \prec \dots$

This must have a supremum within $[\![\mathcal{P}]\!]$.



Unbounded infinite chains imply :

Computational paths with *infinitely many* distinct steps.

$y\left(\mathcal{P}^{+}\right) x$ but $\mathcal{P}^{N}(x) \neq y \ \forall N > 0 \in \mathbb{N}$

Thus $([\![\mathcal{P}]\!], \prec)$ is **chain-complete**.

The key property we want is **directed-completeness** :

- A **directed set** $D \subseteq [\![\mathcal{P}]\!]$ is one for which any $d, d' \in D$ there exists some $e \in D$ with $d, d' \prec e$.
- **Iwamura's Lemma** is a key tool of domain theory :

  "Chain-Completeness implies Directed-Completeness,"

  – a Lemma on Directed Sets (Tsurane Iwamura 1944)

So, finally ...

$([\![\mathcal{M}]\!], \prec)$ is a Scott Domain.

Computing suprema : how & why we should do this.

Some motivation

Given a transition system $(X, \mathcal{P})$, recall the start & halt configurations :

$$X_{start} = \{s : \mathcal{P}^{-1}(s)\} \text{ empty}\} \quad , \quad X_{halt} = \{s : \mathcal{P}(s) \text{ empty}\}$$

Define the **boundary** to be $\Delta = X_{start} \cup X_{halt} \subseteq X$

and the **interior** to be its compliment, $\mathcal{O} = X \backslash \Delta$.

We wish to move from :

- The most detailed step-by-step description, $\mathcal{P} : X \to X$ to
- The most detailed start-halt description

  The 'largest' element of $[\![\mathcal{P}]\!]$ defined solely on the boundary Δ.

Matrix form of transitions

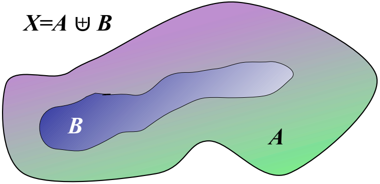
Taking an *arbitrary* split of the configurations :

Let us divide the configuration set as disjoint union, $X = A \uplus B$:

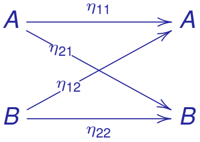
Every $\eta \in [\mathcal{P}]$ can be written as a **matrix**

$$\eta = \begin{pmatrix} \eta_{11} & \eta_{12} \\ \eta_{21} & \eta_{22} \end{pmatrix}$$

Where entries in distinct columns have disjoint support.

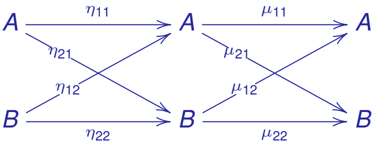


In digraph form :

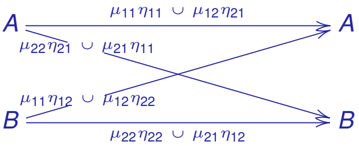


Composition as ‘summing over paths’

Composition of partial functions (or relations, or partial injective functions, etc.) is given by the usual ‘summing over paths’ matrix formula :

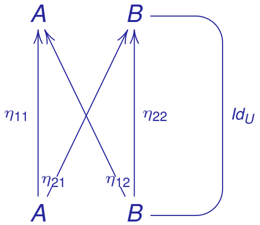


$=$

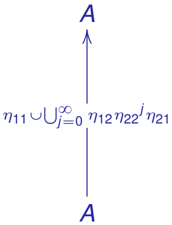


The trace on partial functions

The **‘particle-style’ categorical trace** : another “summing over paths” construction



giving :



A *partial function* $Tr^B \begin{pmatrix} \eta_{11} & \eta_{12} \\ \eta_{21} & \eta_{22} \end{pmatrix} = \eta_{11} \cup \bigcup_{j=0}^{\infty} \eta_{12}{\eta_{22}}^j \eta_{21}$

Some background

This is an example of a **categorical trace** :

- Abstract cat. definition introduced by A. Joyal, R. Street, D. Verity (1996)
  (See also S. Abramsky (1996) & M. Hyland (unpublished))
  — giving the particle-style trace on Relations with Disjoint Union as an example.
- Extended to other categories by various authors
  1. Partial functions (E. Haghverdi 2001)
  2. Partial Injective functions (PMH 1997)
     — prefigured by Lutz & Derby's JANUS programming language (1982)
  3. Fixed-point operators (Hyland, Benton 2002)
  4. Stochastic Relations & Markov processes (P. Panangaden 2007, 2009)
  5. . . . and *many others*.
- Seen in *slightly disguised form* as Girard's Resolution Formula,
  from his first three Geometry of Interaction papers (1989-94)

The trace, and refinement

Consider a transition system $(A \uplus B, \mathcal{P})$ and a representation $\eta \in [\mathcal{P}]$

The trace $Tr^B(\eta)$ satisfies

- "Representations refine their traces" : $Tr^B(\eta) \prec \eta$.

  As a (trivial) consequence

  $$\eta \prec \mu \;\Rightarrow\; Tr^B(\eta) \prec \mu$$

- "Traces preserve cycle-freeness"

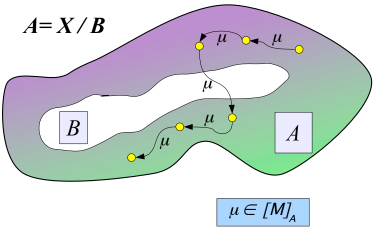
  $$\eta \in [\![\mathcal{P}]\!] \quad \Rightarrow \quad Tr^B(\eta) \in [\![\mathcal{P}]\!]$$

- Given some $\nu \prec \eta$ that is *only defined on A* then $\nu \prec Tr^B(\eta)$.

The categorical trace as a supremum

- Given $X = A \uplus B$,
- Let us denote the representations '*only defined on A*' by $[\mathcal{M}]_A =$

$$\{\mu \ : \ dom(\mu) \subseteq A \supseteq im(\mu)\} \subseteq [\mathcal{P}]$$



Whenever $[\mathcal{M}]_A$ is partially ordered :

$$Tr^B(\eta) = Sup\{\mu \in [\mathcal{M}]_A \ : \ \mu < \eta\}$$

A significant special case

Let us split the configuration set into

The boundary Δ – the start-halt states.

The Interior $\mathcal{O}$ – the 'intermediate' states.

Definition The **black box semantics** $(\!|\mathcal{P}|\!) \subseteq [\![\mathcal{P}]\!] \subseteq [\mathcal{P}]$ is the set of representations only defined on the boundary.

Partial functions that take start states to halt states

The most general of these is of course of interest!

- On $(\!|\mathcal{P}|\!)$, the refinement $\prec$ becomes the *inclusion ordering* on partial functions:

$$f \prec g \text{ exactly when } f(x) = y \Rightarrow g(x) = y$$

- $((\!|\mathcal{P}|\!), \prec)$ is a lattice-enriched semigroup, since $\eta\mu = \bot$, for all $\eta, \mu \in (\!|\mathcal{P}|\!)$.
  - **Bottom element :** the nowhere-defined partial function $\bot$.
  - **Top element :** the trace $Tr^{\mathcal{O}}(\mathcal{P})$.

Confluence of refinement ?

The categorical trace allows us to move from

- the most general **step-by-step** description of dynamics, to
- the most general **start-halt** description of the same system

in a single jump . . . but sometimes, a single jump is too much.

Consider 'hiding the behaviour' one configuration at a time

Given a subset of configurations $\{x_0, x_1, x_2\} \subseteq X$. How does the chain

$$Tr^{\{x_0, x_1 x_2\}}(\eta) \prec Tr^{\{x_0, x_1\}}(\eta) \prec Tr^{\{x_0\}}(\eta) \prec \eta$$

relate to

$$Tr^{\{x_2\}}\left(Tr^{\{x_1\}}\left(Tr^{\{x_0\}}(\eta)\right)\right) \prec Tr^{\{x_1\}}\left(Tr^{\{x_0\}}(\eta)\right) \prec Tr^{\{x_0\}}(\eta) \prec \eta$$

Or indeed **re-orderings** of either of these ?

A key property of traces

A key property of traces is that they are **confluent**.

From Joyal, Street , & Verity (1996)

The **vanishing II axiom** states[a]

$$Tr^A\left(Tr^B(\_)\right) \;=\; Tr^B\left(Tr^A(\_)\right) \;=\; Tr^{A\uplus B}(\_)$$

[a] In our setting, up to symmetry & inclusion maps.

The process of moving from :

- operational, or step-by-step descriptions of a transition system, to
- denotational or start-halt descriptions of the same system,

is *confluent* and can be carried out in a fine-grained manner.

A property equally at home in logic, lambda calculus, and automata theory :)

Future directions / ongoing work

- **Logic / Category Theory** What are the properties of traces on monoids (i.e. categories with only one object) ?

  "On strict extensional reflexivity on compact closed categories "
  — *Outstanding Contributions in Logic (PMH 2013)*

- **Automata Theory** Is there a notion of trace that preserves properties such as *planarity*? – a property already studied in automata theory & logic.

- **Information Theory** Refinement describes the notion of "hiding information" about a computation.

  Less info. $\eta \prec \mu$ more info.

  Can we quantify how much, in information-theoretic terms?

- **Number theory / Foundations of computation** J. Conway's proof of undecidability / computational universality in elementary arithmetic is based on :
  - iterative problems on simple arithmetic operators,
  - a translation into Universal Register Machines.

  Can we describe this in *logical / categorical* terms instead?