Full abstraction for fair testing in CCS

Tom Hirschowitz
CNRS and Université de Savoie

CALCO 2013

Motivation

Reasoning on programming languages:

- until now, mostly methods,
- we would like a theory.

We would like to be able to say:

> "By Theorem T, the morphism f from language L to language L′ preserves and reflects such observational equivalence".

Leads to stupid questions like:

- What is a programming language?
- What is an observational equivalence?
- What is a compilation?

Motivation : a theory of programming languages

Other attempts

- Plotkin,Turi, et al. Categorical approach to operational semantics.
- Montanari et al. Tile model: double-categorical approach.
- Plotkin and Power. Lawvere theories.
- Ciancia. dialgebras.
- Hirscho. 2-categorical approach to higher-order rewriting.
- ... (I thought of two of the above only yesterday, guess which) ?

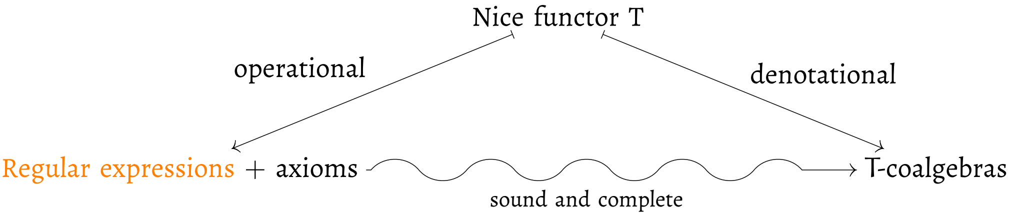
A starting point: Kleene coalgebra

- Most other attempts organise syntax and reductions into some algebraic structure.
- Idea from automata theory:

Kleene coalgebra [Bonchi,Bonsangue,Rutten,Silva,...]

Start from a nice functor, and derive syntax and axioms.

- The functor encapsulates the `rule of the game'.



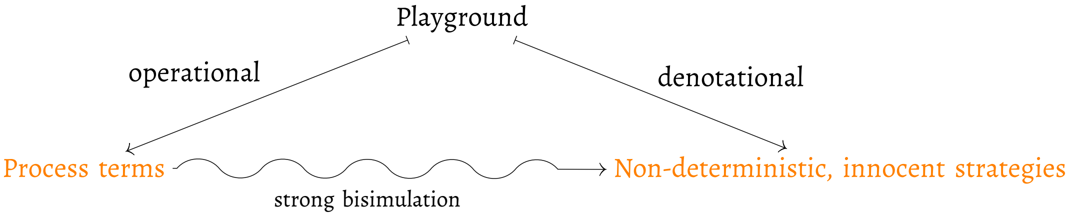
A starting point: Kleene coalgebra

This work may be seen as an attempt to adapt Kleene coalgebra to the world of programming languages.

What is missing?

In game semantical terms, Kleene coalgebra seems to only account for `one-player' games.
$\rightsquigarrow$ Replace the functor with something else.

Rule of the game = playground



Innocent, non-deterministic strategies

- In game semantics, they are known to be problematic (Harmer).
- Solution from presheaf semantics (Joyal, Nielsen, Winskel):

Change definition of strategies:

- prefix-closed sets of plays;
- functors $\mathrm{Plays}^{op} \to 2$, where 2 is the poset $0 \leqslant 1$;
- functors $\mathrm{Plays}^{op} \to$ sets.

- Then incorporate innocence.

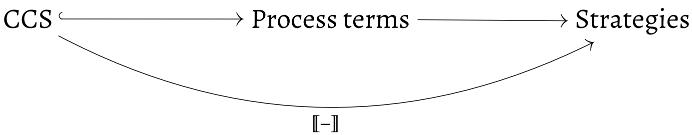
Slogan

Innocent, non-deterministic strategies = innocent presheaves!

We'll see what that means in a moment.

Application

- A playground for Milner's CCS.
- Simple categorical tools $\rightsquigarrow$ fair testing, denotationally: $S \sim T$.
- Translation of CCS processes:



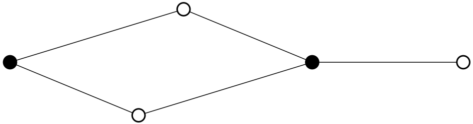
Theorem.
$P \sim_s Q$ iff $[\![P]\!] \sim [\![Q]\!]$, where $\sim_s$ is standard fair testing equivalence.

Open question: can any of this be derived in the general setting?

This talk

1. The playground for CCS.
2. Innocent, non-deterministic strategies.
3. Semantic fair testing.
4. Idea of the translation CCS $\longrightarrow$ Strategies.

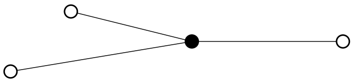
Positions



- ● ≈ player.
- ○ ≈ channel.
- Edges : `player knows channel'.

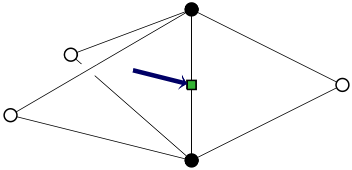
Example move: input

Initial and final positions are the same, e.g.



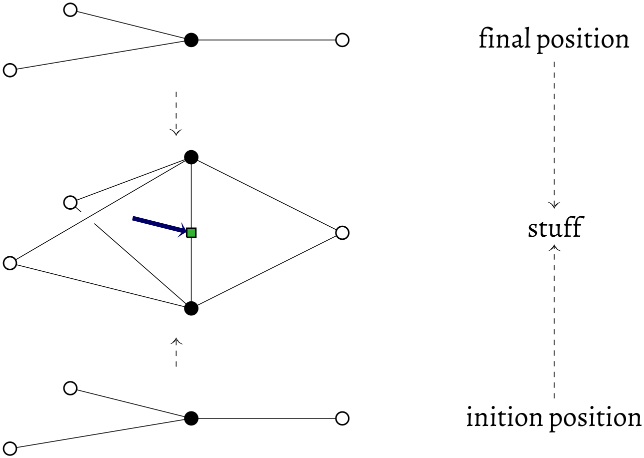
- But: moves are not a mere binary relation (initial position , final position).
- Instead: cospans initial $\rightarrow$ stuff $\leftarrow$ final.
- What stuff? A kind of higher-dimensional graph.

Higher-dimensional graph for the input move

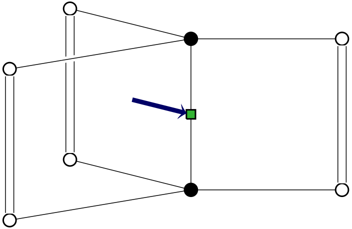


- The arrow indicates on which channel the input occurs.
- One such graph for each arity (here 3).
- Formal definition: see (long version of) paper.

The input move

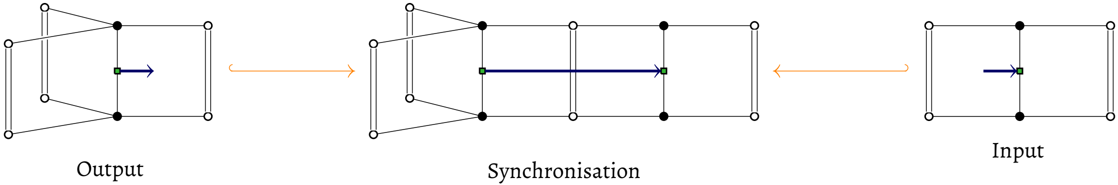


drawn for conciseness as:



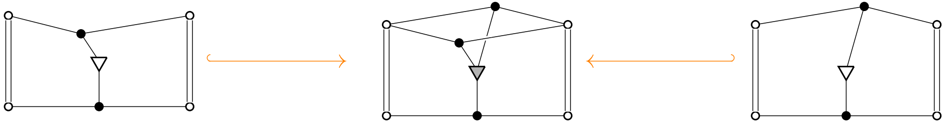
Moves: input/output

Using the previous convention:

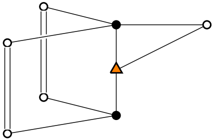


Orange arrows: cospan morphisms.

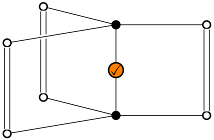
Moves, continued



Left fork — Fork — Right fork



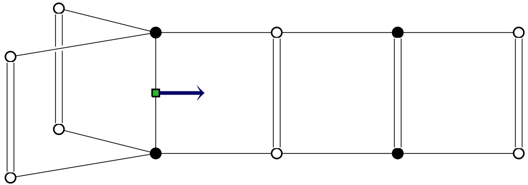
Channel creation



Tick

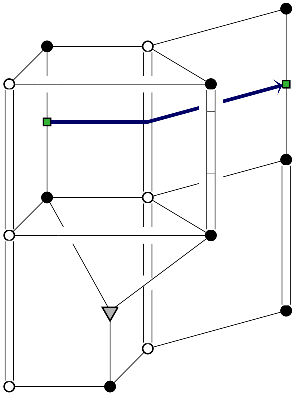
Local vs. global moves

- Until now, moves were local: only involved players were shown.
- Global moves obtained by embedding into larger positions.
- E.g.:



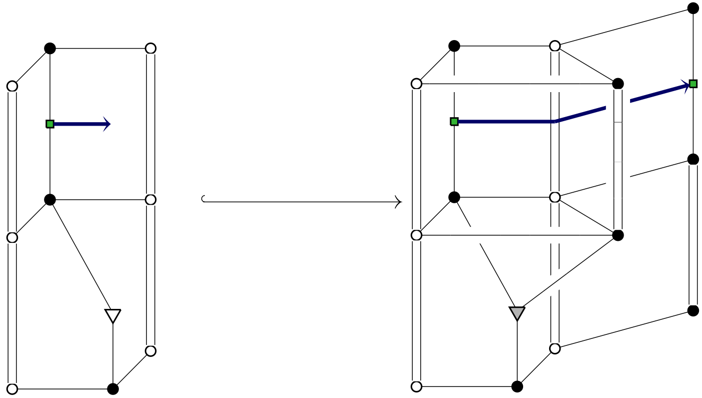
Plays

Obtained by piling up global moves:



Feature a certain amount of concurrency.

Category of plays over position X: P_X



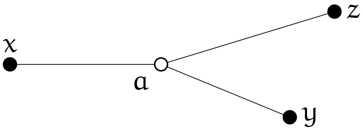
- Plus prefix inclusion.
- Possibly several morphisms between two plays.
- Otherwise, close to configuration posets of event structures.

Naive, non-deterministic strategies over position X

Definition 1. Strategy over X

Presheaf $\mathbb{P}_X^{op} \to$ sets.

Too general: consider the position



and the naive strategy

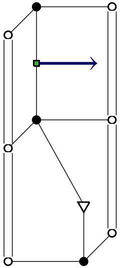
- accepting $x \to y$,
- accepting outside $\to z$,
- but refusing $x \to z$.

Players x and z should not be allowed to choose with whom they synchronise.

Non-deterministic, innocent strategies

Views: let $V_X \subseteq P_X$ consist of histories of exactly one player.

Example:



Definition 2. Innocent strategies

Presheaf $V_X^{op} \longrightarrow$ sets.

Problem: no obvious inclusion innocent $\subseteq$ naive.

Fair testing: overview

- Global behaviour: essentially, innocent $\longrightarrow$ naive.
- Interaction.
- Fair testing.

Global behaviour

By right Kan extension:

$$\begin{array}{ccccc} V_X^{op} & & \xrightarrow{\quad i^{op} \quad} & & P_X^{op} \\ & & \cong & & \\ & S \searrow & & \swarrow \bar{S} = \mathrm{ran}_{i^{op}}(S) & \\ & & \text{sets} & & \end{array}$$

Explicit formula

- General : $\bar{S}(p) = \int_{v \in V_X} S(v)^{P_X(v,p)}$.
- Boolean case; p accepted iff all its views are : $\bar{S}(p) = \bigwedge_{\{(v \xrightarrow{\alpha} p) \in P_X\}} S(v)$.

Global behaviour

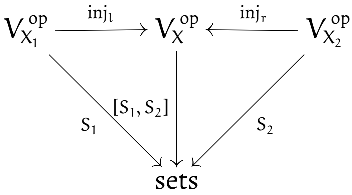
By restricting to closed-world plays: $S \mapsto \bar{S}$

Interaction

- Split the players of position X into two teams.
- Obtain two subpositions $X_1 \hookrightarrow X \hookleftarrow X_2$ sharing no player.
- We have

$$V_X \simeq V_{X_1} + V_{X_2}.$$

- Let S_1 play against S_2 by *copairing* :



Fair testing

- Successful play: one with at least one ✓.
- $S \perp T$: all unsuccessful executions of $[S, T]$ extend to successful ones.

Definition 3. Semantic fair testing equivalence

$S \sim S'$ iff $\forall T, S \perp T \Leftrightarrow S' \perp T$.

A syntax for strategies

- Derivable from any playground.
- Idea:

  A strategy = what remains of it after each atomic view b.

- For CCS:

$$\frac{\ldots \quad n_b \vdash S_b \quad \ldots}{n \vdash_D \left\langle (S_b)_b \right\rangle}\text{Definite strategies} \qquad \frac{\ldots \quad n \vdash_D D_i \quad \ldots}{n \vdash \bigoplus_{i \in p} D_i}\text{Plain strategies}$$

where $b : n_b \to n$, for all b.

The translation

$$P \mid Q \mapsto \boxed{\left\langle \begin{array}{rcl} \pi_n^l & \mapsto & [\![P]\!] \\ \pi_n^r & \mapsto & [\![Q]\!] \\ - & \mapsto & \emptyset \end{array} \right\rangle}$$

$$\nu a\,.\,P \mapsto \boxed{\left\langle \begin{array}{rcl} \nu_n & \mapsto & [\![P]\!] \\ - & \mapsto & \emptyset \end{array} \right\rangle}$$

$$a\,.\,P \mapsto \boxed{\left\langle \begin{array}{rcl} \iota_{n,a} & \mapsto & [\![P]\!] \\ - & \mapsto & \emptyset \end{array} \right\rangle}$$

$$\ldots$$

Main result

Theorem.

$P \sim_s Q$ iff $[\![P]\!] \sim [\![Q]\!]$.

Future work

- Scale the approach to π (almost), Join, λ,...
- Tools for generating playgrounds (with Clovis Eberhart).
- Investigate morphisms of playgrounds.
- Link with exotic settings like cellular automata.
- \`Double category of elements' $\rightsquigarrow$ new notion of abstract rewriting system.