Fully-abstract games for $\pi$

Clovis Eberhart  Tom Hirschowitz  Thomas Seiller

Galop '14, Grenoble
Standard games semantics (HO, AJM, Nickau)

- Initial aim: denotational semantics for functional languages.
- Went beyond: references, control, etc.
- Even concurrent languages (Laird, Ghica-Murawski), but for coarse behavioural equivalence, may testing equivalence.
Asynchronous game semantics (Abramsky, Melliès, Mimram)

- Aim: denotational semantics for full linear logic.
- Main ingredient: plays should not need to record the exact order of moves, only their causal dependencies.
- Technically: games played on event structures.
- (Though plays remain sequences of moves, asynchrony being implemented as stability of strategies under permutations.)
Asynchronous game semantics and concurrency

Surprisingly, never applied beyond *confluent* languages.

Possible reason:

- Standard notion of strategy: set of `accepted' plays.
- Akin to trace equivalence in concurrency.
- Does not account for branching behaviour.
- E.g., does not distinguish between

```
\begin{align*}
\chi & \xrightarrow{a} \star \xrightarrow{a} \chi' \\
& \downarrow b \\
y & \xrightarrow{} y'
\end{align*}
```

and

```
\begin{align*}
\chi & \xrightarrow{a} \\
& \downarrow b \\
y & \xrightarrow{c} y'
\end{align*}
```

- Hence, could not have been finer than Laird or Ghica-Murawski.
Concurrent games (Winskel, Castellan, Clairambault, Rideau)

Keep the idea of games on event structures.

<table>
<thead>
<tr>
<th>New notion of strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>May accept plays in <strong>more than one way</strong>.</td>
</tr>
</tbody>
</table>

E.g.,
- toss a coin, `privately',
- remember the result,
- modify behaviour accordingly,
- which may show only a few moves later.

To our knowledge, only starting to lead to new models of concurrent languages.
This work

Follow up on previous work on CCS (with Pous, ICE '11 and SACS, then CALCO '13).
- Same ideas, independently:
  - concurrent notion of play,
  - strategies may accept plays in several ways.
- Implemented very differently!
How differently?

1. Truly multi-player game.
2. Plays form a proper category.
   - May be several morphisms between plays.
   - Departs from plays as configurations in event structures.
   - Probably close to Levy's notion of morphisms between plays.
3. Borrow notion of strategies following presheaf models (JNW):
   - prefix-closed sets of plays;
   - functors $\text{Plays}^{\text{op}} \rightarrow 2$, where 2 is the poset $0 \leq 1$;
   - functors $\text{Plays}^{\text{op}} \rightarrow \text{sets}$.
4. Reincorporate game semantical innocence.

Last point is not obvious, let's explain a bit more.
Innocent presheaves

- Subcategory Views $\rightarrow$ Plays.
- Innocent presheaves: $\text{Views}^{\text{op}} \rightarrow \text{sets}$.

**Recovering global behaviour**

By right extension along $\text{Views}^{\text{op}} \rightarrow \text{Plays}^{\text{op}}$.

(Works thanks to the more general notion of morphism.)

- Boolean setting: play accepted iff all views are.
- General case: way to accept a play = compatible family of ways to accept views.
What do we obtain?

- A concurrent game semantics for $\pi$.
- Equality of strategies is very intensional.
- But the model enjoys a lot of structure. Simple categorical tools $\rightsquigarrow$
  - parallel composition of strategies,
  - global behaviour of a strategy, hence
  - semantic fair testing: $S \sim T$.

Main result

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

- Detects branching behaviour.
- Perceived as one of the most sensible testing equivalences.
- Testing equivalences more intrinsic than LTSs.
- E.g., bisimilarity depends on the chosen LTS.
This talk

The game for $\pi$ is the crux of the construction: emphasised here.

1. The game for $\pi$.
2. Innocent presheaves and global behaviour.
3. Semantic fair testing.
4. Idea of the translation $\pi \rightarrow$ Strategies.
What is meant by `a game'

- Set of positions: players and channels (truly multi-player).
- Maybe surprising: moves are not just a binary relation:
  - more than when there is a move $X \rightarrow Y$,
  - rather, how one moves from $X$ to $Y$.
- Here, the how is represented as an object $M$ in a category of diagrams.

\[
\text{Moves } X \rightarrow Y \text{ are cospans } X \rightarrow M \leftarrow Y.\]
Positions

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

Now, what are diagrams? A kind of higher-dimensional graph.
Higher-dimensional graph for the input move

- Thicker, blue line = `pipe' = carrier channel.
- Green arrow = received channel.
- `Late' feel: the input arrow is fresh (for input alone).
- One such graph for each arity (here 3).
- Formal definition: please ask if interested, i find it cute.
The input move

And written as $\iota_{2,1} : 3 \rightarrow 2$. 
**Moves: input/output**

Using the previous convention:

Output

Synchronisation

Input

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.:

  - First player outputs to the environment.
  - Second player does not notice anything.
Obtained by piling up global moves:

(identification the two red nodes)
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 $P_X^{op} \to \text{sets}$.

Too general: consider the position

and the naive strategy

- accepting $x \to y$,
- accepting environment $\to z$,
- 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 $V_X^{op} \rightarrow \text{sets}$.

Problem: no obvious inclusion innocent $\subseteq$ naive.
Fair testing: overview

- Global behaviour: essentially, innocent $\mapsto$ naive.
- Interaction, a.k.a. parallel composition.
- Fair testing.
Global behaviour

By right Kan extension:

\[ V_X^{\text{op}} \xrightarrow{i^{\text{op}}} P_X^{\text{op}} \]

\[ S \xleftarrow{\cong} \text{sets} \xrightarrow{S' = \text{ran}_{X^{\text{op}}}(S)} \]

**Explicit formula**

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

**Global behaviour**

By restricting to closed-world plays: \( S \leftrightarrow S' \leftrightarrow \overline{S} \).
Interaction, a.k.a. parallel composition

- Split the players of position \( X \) into two teams.
- Obtain two subpositions \( X_1 \leftrightarrow X \leftrightarrow X_2 \) sharing no player.
- We have

\[
V_X \simeq V_{X_1} + V_{X_2}.
\]

- Let \( S_1 \) play against \( S_2 \) by copairing:

![Diagram showing the copairing of \( S_1 \) and \( S_2 \) with \( V_{X_1} \), \( V_X \), and \( V_{X_2} \).]
Fair testing

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

**Definition 3. Semantic fair testing equivalence**

$S \sim S'$ iff $\forall T, S \perp T \iff S' \perp T$. 
A syntax for strategies

- Idea:

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

- Generic syntax:

\[
\begin{align*}
\ldots & \quad n_b \vdash S_b \quad \ldots \\
& \quad n \vdash_D (S_b)_b \\
\end{align*}
\]

\[
\begin{align*}
\ldots & \quad n \vdash_D D_i \quad \ldots \\
& \quad n \vdash \bigoplus_{i \in p} D_i \\
\end{align*}
\]

where \( b : n_b \to n \) ranges over

- \( \iota_{n,a} : n + 1 \to n \),
- \( o_{n,a,b} : n \to n \),
- \( \pi^l_n : n \to n \),
- \( \pi^r_n : n \to n \),
- \( \nu_n : n + 1 \to n \),
- \( \Box_n : n \to n \).
The translation (examples)

\[ P \mid Q \rightarrow \langle \pi_n^l \rightarrow [P] \]
\[ \pi_n^r \rightarrow [Q] \]
\[ - \rightarrow \emptyset \rangle \]

\[ \forall a \cdot P \rightarrow \langle \forall_n \rightarrow [P] \]
\[ - \rightarrow \emptyset \rangle \]

\[ a \cdot P + a \cdot Q \rightarrow \langle \iota_{n,a} \rightarrow [P] \oplus [Q] \]
\[ - \rightarrow \emptyset \rangle \]

...
Main result

Recall standard fair testing equivalence:

- $P \in \underline{\cdot} \iff \forall \ P \Rightarrow Q, \exists \ Q \xrightarrow{\circ} R$,
- $P \sim_s Q \iff \forall R, ((P \mid R) \in \underline{\cdot}) \iff ((Q \mid R) \in \underline{\cdot})$.

**Theorem.**

$P \sim_s Q \iff \llbracket P \rrbracket \sim \llbracket Q \rrbracket$. 
Summary

- Concurrent game semantics through innocent presheaves.
- Full abstraction for a semantic analogue of fair testing equivalence.
- Techniques from categorical combinatorics:
  - various kinds of presheaves,
  - right Kan extension.
Left out from this talk

Playgrounds: an algebraic structure \( \approx \) rule of the game.

**Crucial point in playgrounds**

Plays fibred over positions.

I.e., operation to restrict a play on \( X \) along \( Y \to X \).

Much more satisfactory treatment than in previous work (using a factorisation system).

(Please ask if interested...
FAQ

Aren't strategies too syntactic (they contain terms)?

First answer: can't expect otherwise.
- HO strategies \( \approx \) \( \beta \)-normal \( \eta \)-long terms.
- In concurrency: no notion of normal form.

Second answer, maybe better viewpoint. Not really a game semantics, rather:
- a categorical combinatorics approach
- to a slightly generalised syntax / semantics duality.

Promising, thanks to high-level tools from categorical combinatorics.
Future work

- Scale the approach to Join, HOπP (passivation), λ,...
- Tools for generating playgrounds (with Clovis Eberhart).
- Investigate morphisms between playgrounds (c.f. compilation).
- Link with exotic settings like cellular automata.
- `Double category of elements' \( \sim \) new notion of abstract rewriting system.

Thanks for your attention!