Asynchronous Games
Innocence without Alternation

Paul-André Melliès    Samuel Mimram

Laboratoire PPS, CNRS – Université Paris 7

CONCUR 2007
September 7, 2007
Game semantics

**Denotational semantics**
Giving properties of programs which are invariant during the execution.
Denotational semantics
Giving properties of programs which are invariant during the execution.

Game semantics

1. **formulas** \( A \) are interpreted by **games**
2. **proofs** \( \pi : A \rightarrow B \) are interpreted by **strategies**
Denotational semantics
Giving properties of programs which are invariant during the execution.

Game semantics

1. formulas $A$ are interpreted by games
2. proofs $\pi : A \rightarrow B$ are interpreted by strategies

We also want composition (and other structures) to be preserved by the interpretation.
Concurrency in game semantics

Game semantics is a *trace semantics*.

The program $P$ emits and receives moves

$$ P \xrightarrow{m_0} P_1 \xrightarrow{m_1} P_2 \xrightarrow{m_2} \ldots $$

played in a game.
Concurrency in game semantics

Game semantics is a *trace semantics*.

The program $P$ emits and receives moves

$$P \xrightarrow{m_0} P_1 \xrightarrow{m_1} P_2 \xrightarrow{m_2} \ldots$$

played in a game.

Here, we will refine it as

a Mazurkiewicz trace semantics for proofs

based on event structures.
Unifying semantics of linear logic

Sequential games
Hyland, Ong 1994
Abramsky, Jagadeesan, Malacaria 1994

Asynchronous games
Melliès 2004

Coherence spaces
Relational model
Girard 1987

Concurrent games
Abramsky, Melliès 1999

Event structures
Curien, Faggian 2005
Varraca, Yoshida 2006
Part I

Asynchronous games
Asynchronous games

A 2-player event structure

$$(M, \leq, \#, \lambda)$$

consisting of

- a set of moves $M$
- a partial order $\leq$ expressing causal dependencies
- a symmetric relation $\#$ expressing incompatibilities
- a polarization of moves $\lambda : M \rightarrow \{O, P\}$
• **positions** are downward-closed sets of compatible moves
• **plays** are paths between positions, starting from $\emptyset$
Playing in games

- **positions** are downward-closed sets of compatible moves
- **plays** are paths between positions, starting from $\emptyset$

![Diagram](image)
Playing in games

- **positions** are downward-closed sets of compatible moves
- **plays** are paths between positions, starting from $\emptyset$

\[ q_L \quad q_R \]

\[ \emptyset \quad \{q_L\} \quad \{q_R\} \]

\[ \{q_L, q_R\} \]

\[ o \otimes o \]
Playing in games

- **positions** are downward-closed sets of compatible moves
- **plays** are paths between positions, starting from $\emptyset$

\[ \begin{align*}
    & q_L & \rightarrow & \{ q_L \} & \sim & \{ q_R \} \\
    & o \otimes o & \rightarrow & \{ q_L \} & \sim & \{ q_R \} \rightarrow & \{ q_L, q_R \} \\
    & q_R & \rightarrow & \{ q_R \} & \rightarrow & \{ q_L \} \\
\end{align*} \]
An approach to interferences

The Mazurkiewicz approach to *true concurrency*.

\[ a \parallel b \quad \text{vs.} \quad a \cdot b + b \cdot a \]

\[ \begin{array}{c}
\begin{array}{c}
\xymatrix{a
d \sim \ar[rr] & & b
\ar[rrd] & & \ar[lld]
\ar[rr] & & a
b}
\end{array}
\end{array} \]

\[ \begin{array}{c}
\begin{array}{c}
\xymatrix{a
d \sim \ar[rr] & & b
\ar[rrd] & & \ar[lld]
\ar[rr] & & a
b}
\end{array}
\end{array} \]

\[ x := 4 \parallel y := 5 \quad \text{multiplicatives} \]

\[ x := 4 \parallel x := 5 \quad \text{additives} \]
Implementations of the conjunction

\[ \text{bool} \otimes \text{bool} \rightarrow \text{bool} \]

Left and
Implementations of the conjunction

Right and

\[
\text{bool} \otimes \text{bool} \rightarrow \text{bool}
\]
Implementations of the conjunction

Parallel and

\[ \text{bool} \otimes \text{bool} \rightarrow \text{bool} \]
Implementations of the conjunction

Parallel and

\( \text{bool} \otimes \text{bool} \rightarrow \text{bool} \)

\( q \)

\( q \)

\( V \)

\( F \)

\( F \)
Implementations of the conjunction

Parallel and

$\text{bool} \otimes \text{bool} \rightarrow \text{bool}$
Asynchronous games

Parallel and
Asynchronous games

Left and
Asynchronous games

A game induces an asynchronous graph $G$:

- vertices are positions (+ initial position $*$),
- edges are moves,
- 2-dimensional tiles

\[ \begin{array}{c}
\text{x} \\
\downarrow m \quad \downarrow n \\
\sim \\
\downarrow n \quad \downarrow m \\
z
\end{array} \]

generate homotopy between paths.
A logic for game semantics

- we only consider formulas of MALL:

$$
\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \otimes B} \quad \quad \quad \frac{\vdash \Gamma_1, A \quad \vdash \Gamma_2, B}{\vdash \Gamma_1, \Gamma_2, A \otimes B} \quad \quad \quad (\otimes)
$$

$$
\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} \quad \quad \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \quad \quad \quad (\oplus)
$$
A logic for game semantics

• we only consider formulas of MALL:

\[
\frac{\Gamma, A, B}{\Gamma, A \otimes B} \quad \frac{\Gamma_1, A, \Gamma_2, B}{\Gamma_1, \Gamma_2, A \otimes B}
\]

\[
\frac{\Gamma, A}{\Gamma, A \land B} \quad \frac{\Gamma, A}{\Gamma, A \oplus B}
\]

• with explicit moves:

\[
\frac{\Gamma, A}{\Gamma, \uparrow A} \quad \frac{\Gamma, A}{\Gamma, \downarrow A}
\]
In linear logic, the formula corresponding to booleans is

$$\text{bool} = \uparrow(\downarrow 1 \oplus \downarrow 1)$$

which is like of $1 \oplus 1$ with explicit changes of polarities.
From formulas to games

In linear logic, the formula corresponding to booleans is

$$\text{bool} = \uparrow(\downarrow 1 \oplus \downarrow 1)$$

It can be drawn as

```
  \uparrow
  \downarrow  \oplus  \downarrow
  1     1
```
In linear logic, the formula corresponding to booleans is

\[ \text{bool} = \uparrow(\downarrow 1 \oplus \downarrow 1) \]

It can be drawn as
In linear logic, the formula corresponding to booleans is

\[
\text{bool} = \uparrow(\downarrow 1 \oplus \downarrow 1)
\]

It can be drawn as

```
    q
   / \   /  \\
  V  #  F
```
In linear logic, the formula corresponding to booleans is

\[ \text{bool} = \uparrow (\downarrow 1 \oplus \downarrow 1) \]

It can be drawn as

```
      *  
     / \  
    q   q  
   / \  /  
  V   V  F  F
  |   |   |   |  
  F   V   F   V
```
From proofs to strategies

The game associated to $\uparrow A$ is of the form

$$
\uparrow
\begin{array}{c}
\arrown \uparrow
\end{array}
\begin{array}{c}
A
\end{array}
$$
From proofs to strategies

The game associated to $\uparrow A \otimes \uparrow B = \uparrow A \otimes \uparrow B$ is of the form

\[
\begin{array}{c}
\uparrow \\
A
\end{array}
\quad
\begin{array}{c}
\uparrow \\
B
\end{array}
\]
From proofs to strategies

The game associated to \( \uparrow A \otimes \uparrow B = \uparrow A \otimes \uparrow B \) is of the form

\[
\begin{array}{cc}
\uparrow & \uparrow \\
\downarrow & \downarrow \\
A & B
\end{array}
\]

The corresponding asynchronous graph contains

\[
\begin{array}{c}
* \uparrow * \\
\downarrow_{\uparrow A} & \downarrow_{\uparrow B} \\
\uparrow* \uparrow* & \uparrow* \\
\downarrow_{\uparrow B} & \downarrow_{\uparrow A} \\
\uparrow* \uparrow* & \uparrow* \uparrow*
\end{array}
\]

\[
\vdots
\]
From proofs to strategies

The game associated to $\uparrow A \otimes \uparrow B = \uparrow A \curlyvee \uparrow B$ is of the form

$$
\begin{array}{c}
\uparrow \\
A \\
\end{array}
\quad \quad \quad
\begin{array}{c}
\uparrow \\
B \\
\end{array}
$$

The corresponding asynchronous graph contains

```
\begin{array}{c}
\uparrow_A \\
\uparrow_* \uparrow_* \\
\uparrow_* \uparrow_* \\
\hline
\uparrow_* \uparrow_* \\
\end{array}
\quad \quad \quad
\begin{array}{c}
\uparrow_B \\
\uparrow_* \uparrow_* \\
\uparrow_* \uparrow_* \\
\hline
\uparrow_* \uparrow_* \\
\end{array}
```
From proofs to strategies

Three proofs of $\uparrow A \nRightarrow \uparrow B$:

$\vdash \uparrow A, \uparrow B$

*, *
From proofs to strategies

Three proofs of $\uparrow A \nleftrightarrow \uparrow B$:

$\vdash A, \uparrow B$

$\vdash \uparrow A, \uparrow B$ (↑)

$\vdash \uparrow A$

$\uparrow *$, *
Three proofs of $\uparrow A \Leftrightarrow \uparrow B$:
From proofs to strategies

Three proofs of $\uparrow A \nRightarrow \uparrow B$:

\[
\vdash A, B \\
\vdash A, \uparrow B (\uparrow) \\
\vdash \uparrow A, \uparrow B (\uparrow)
\]
From proofs to strategies

Three proofs of $A \pitchfork B$:

$\vdash A, B \vdash \overline{A, B}$
From proofs to strategies

Three proofs of \( \uparrow A \not\iff \uparrow B \):

\[
\frac{
\frac{
\vdash \uparrow A, B
}{
\vdash \uparrow A, \uparrow B
}
}{
\vdash \uparrow A, \uparrow B
}
\]

\(*, *, \uparrow B \rightarrow *, \uparrow *\)
From proofs to strategies

Three proofs of $\uparrow A \nleftrightarrow \uparrow B$:

\[
\begin{align*}
\vdash A, B \\
\vdash \uparrow A, B \\
\vdash \uparrow \uparrow A, \uparrow B
\end{align*}
\]
From proofs to strategies

Three proofs of \( \uparrow A \nleftrightarrow \uparrow B \):

\[
\begin{align*}
\vdash A, B \\
\vdash \uparrow A, \uparrow B (\uparrow) \\
\vdash \uparrow A, \uparrow B (\uparrow)
\end{align*}
\]
From proofs to strategies

Three proofs of $\uparrow A \nleftrightarrow \uparrow B$:

\[
\begin{align*}
\vdots \\
\vdash A, B \\
\vdash \uparrow A, \uparrow B (\uparrow, \uparrow)
\end{align*}
\]
From proofs to strategies

Three proofs of $\uparrow A \nleq \uparrow B$:

\[
\vdash A, B \\
\vdash \uparrow A, \uparrow B \quad (\uparrow, \uparrow)
\]
Proofs explore formulas

\[
\begin{array}{ll}
\text{play} & = \text{exploration of the formula} \\
\text{proof} & = \text{strategy of exploration}
\end{array}
\]

Every proof is a partial order on moves...
Proofs explore formulas

\[
\begin{array}{ll}
\text{play} & = \text{exploration of the formula} \\
\text{proof} & = \text{strategy of exploration}
\end{array}
\]

Every proof is a partial order on moves which refines the partial order of the game.
Proofs explore formulas

\[
\begin{align*}
\text{play} & = \text{exploration of the formula} \\
\text{proof} & = \text{strategy of exploration}
\end{align*}
\]

Every proof is a partial order on moves which refines the partial order of the game.

\[
\vdash A, B \\
\vdash A, \uparrow B (\uparrow) \\
\vdash \uparrow A, \uparrow B (\uparrow)
\]

\[
\begin{array}{c}
\vdash \uparrow A, \uparrow B (\uparrow)
\end{array}
\]
Proofs explore formulas

\[
\begin{align*}
\text{play} &= \text{exploration of the formula} \\
\text{proof} &= \text{strategy of exploration}
\end{align*}
\]

Every proof is a partial order on moves which refines the partial order of the game.

\[
\begin{array}{c}
\vdash A, B \\
\vdash A, B' \\
\vdash A', B \\
\vdash A', B'
\end{array}
\]

\[
\begin{array}{cc}
\uparrow & \langle \ldots \rangle \\
\uparrow & \uparrow
\end{array}
\]

\[
A \quad B
\]
Proofs explore formulas

\[
\begin{array}{|c|}
\hline
\text{play} & = & \text{exploration of the formula} \\
\text{proof} & = & \text{strategy of exploration} \\
\hline
\end{array}
\]

Every proof is a partial order on moves which refines the partial order of the game.

\[
\vdash A, B \\
\vdash \uparrow A, \uparrow B^{(\uparrow, \uparrow)}
\]

\[
\vdash \uparrow A, \uparrow B \\
\vdash A, B
\]
Towards innocence

Can we characterize the *definable* strategies?

We have to restrict the space of strategies.

\[
\text{innocent strategy} \quad = \quad \text{strategy behaving like a proof}
\]
Part II

Traces vs. event structures
Traces vs. partial orders

formula = event structure on the moves

proof = refinement of the underlying partial order
From causal to sequential

Every event structure defines an asynchronous graph.
From sequential to causal

Here, one needs the Cube Property.
The Cube Property

Theorem

Paths modulo homotopy are given by a partial order on their moves.
Asynchronous games

By definition, an **asynchronous game** is a rooted asynchronous graph satisfying the Cube Property.
Definition
A strategy is a set of plays, closed under prefix.

Definition
A strategy is positional when its paths form a subgraph of the game.
Causal strategies

From now on, we consider *causal strategies* which

1. are positional
2. satisfy properties implying the Cube Property
Unfortunately, causal strategies do not compose...
Part III

A category of asynchronous games
Categories of games and strategies

\[ A \rightarrow B = A^* \ominus B = A^* \otimes B \]

The strategy **not**:

\[
\begin{array}{c}
\text{bool} \xrightarrow{\text{not}} \text{bool} \\
q \\
q \\
V \\
F
\end{array}
\]
Categories of games and strategies

\[ A \rightarrow B = A^* \otimes B = A^* \otimes B \]

The strategy **not**:

\[
\begin{align*}
\text{bool} & \xrightarrow{\text{not}} \text{bool} \\
q & \\
q & \\
F & \\
V &
\end{align*}
\]
Traces compose by *parallel composition*

\[
\begin{align*}
\text{bool} & \longrightarrow \text{bool} & \text{bool} & \longrightarrow \text{bool} \\
q & \quad & \quad & \quad \\
\end{align*}
\]
Composition

Traces compose by \textit{parallel composition}

\[
\begin{align*}
\text{bool} & \longrightarrow \text{bool} & \text{bool} & \longrightarrow \text{bool} \\
q & \quad & q \\
V & \quad & F
\end{align*}
\]
Composition

Traces compose by parallel composition

\[
\text{bool} \rightarrow \text{bool} \quad \text{bool} \rightarrow \text{bool}
\]

\[
\begin{array}{cc}
q & q \\
q & q \\
q & \\
V & \\
F & F \\
V & 
\end{array}
\]
Traces compose by parallel composition + hiding.

\[
\text{bool} \longrightarrow \quad \longrightarrow \quad \text{bool}
\]

\[
q
\]

\[
q
\]

\[
V
\]

\[
V
\]
Determinism

Definition
A strategy $\sigma : A$ is **deterministic** when

\[
\begin{align*}
\sigma \ni m &\quad \text{implies} \quad \sigma \ni n \\
y_1 &\quad y_2
\end{align*}
\]

where $m$ is a Proponent move.
Deterministic strategies do compose!

They form a monoidal category of asynchronous games.
Part IV

Concurrent strategies
**Definition**

A position of a strategy $\sigma$ is **halting** when there is no Proponent move $m : x \rightarrow y$ in $\sigma$.

We write $\sigma^\circ$ for the set of halting positions of $\sigma$. 
Halting positions

The game $\text{true} \otimes \text{false}$.
Halting positions

The *parallel* implementation of true and false.

\[
\begin{array}{c}
\text{Halting positions} \\
\text{The parallel implementation of true and false.}
\end{array}
\]
Halting positions

The left implementation of true and false.
Halting positions

The right implementation of true and false.
Ingenuous strategies

In the spirit of concurrent games Abramsky, Melliès 1999

we would like strategies to be characterized by their halting positions.
Ingenuous strategies

Definition

A strategy \( \sigma \) is **ingenuous** when it is

1. causal,
2. deterministic,
3. courteous:

\[
\begin{array}{c}
\sigma \ni m \\
y_1 \sim y_2 \\
\sigma \ni n \\
z
\end{array}
\]

implies

\[
\begin{array}{c}
\sigma \ni m \\
y_1 \sim y_2 \\
\sigma \ni n \\
\sigma \ni m \\
\sigma \ni n \\
z \ni m \\
m \in \sigma
\end{array}
\]

where \( m \) is a Proponent move.
Ingenuous strategies as relations

Theorem
Every ingenuous strategy $\sigma$ is characterized by its set $\sigma^\circ$ of halting positions.

This set $\sigma^\circ$ describes a closure operator.

| ingenuous strategies $\iff$ concurrent strategies |
Part V

Innocence
Preserving composition

Unfortunately, we don’t have

\[(\sigma; \tau)^\circ = \sigma^\circ; \tau^\circ\]
Preserving composition

The *livelock*:

\[(\sigma; \tau)^\circ \subseteq \sigma^\circ; \tau^\circ\]

\[
A \xrightarrow{\sigma} B \xrightarrow{\tau} C
\]

\[
\begin{array}{ccc}
\ast & \ast & \ast \\
\backslash & \downarrow & \backslash \\
\ast & \ast & \ast \\
\end{array}
\]

\[
\begin{array}{ccc}
X & \ast & Z \\
\downarrow & \downarrow & \ast \\
\vdots & \vdots & \ast \\
\end{array}
\]
Preserving composition

The *livelock*:

\[(\sigma; \tau)^\circ \subseteq \sigma^\circ; \tau^\circ\]

\[A \xrightarrow{\sigma} B \xrightarrow{\tau} C\]

Solution: handle infinite positions
The *deadlock*: 

\[(\sigma; \tau)^* \supseteq \sigma^*; \tau^*\]

\[A \xrightarrow{\sigma} B \xrightarrow{\tau} C\]

* * *

\(\star\) \(\star\) \(\star\)

\(\star\) \(\star\) \(\star\)

\(\star\) \(\star\) \(\star\)

\(\star\) \(\star\) \(\star\)

\(x\) \(y\) \(z\)

---

Preserving composition
Preserving composition

The *deadlock*:

\[(\sigma; \tau)^\circ \supseteq \sigma^\circ; \tau^\circ\]

\[
\begin{align*}
A \xrightarrow{\sigma} B \xrightarrow{\tau} C
\end{align*}
\]

\[
\begin{array}{ccc}
* & * & * \\
\vdash & \vdash & \vdash \\
\times & y & z \\
\end{array}
\]

Solution: add a scheduling criterion
The scheduling criterion

the left conjunction:

\[ \text{bool} \otimes \text{bool} \rightarrow \text{bool} \]
The scheduling criterion

The right boolean composed with the left conjunction:

\[
\text{bool} \otimes \text{bool} \rightarrow \text{bool}
\]
Two kinds of tensors: $\otimes$ and $\otimes$.

$$\text{bool} \otimes \text{bool} \rightarrow \text{bool} = \text{bool}^* \otimes \text{bool}^* \otimes \text{bool}$$
The scheduling criterion

Two kinds of tensors: $\otimes$ and $\exists$.

$\text{bool} \otimes \text{bool}$

$q$

$(\exists F)$

$q$

$(\forall V)$
The scheduling criterion

Two kinds of tensors: $\otimes$ and $\mathcal{A}$.

```
bool $\otimes$ bool
```

```
q
(\(F\)
  q
(\(V\)
  
```

The scheduling criterion

Two kinds of tensors: $\otimes$ and $\gamma$.
Functoriality

Definition
A strategy $\sigma : A$ is **receptive** when for every path $s : * \rightarrow x$ in $\sigma$ and for every Opponent move $m : x \rightarrow y$ the path $s \cdot m : * \rightarrow y$ is also in $\sigma$. 
Functoriality

Definition
A strategy $\sigma : A$ is **receptive** when for every path $s : * \to x$ in $\sigma$ and for every Opponent move $m : x \to y$ the path $s \cdot m : * \to y$ is also in $\sigma$.

Theorem
*Ingenuous strategies which satisfy the scheduling criterion and are receptive compose and satisfy*

$$(\sigma ; \tau)^\circ = \sigma^\circ ; \tau^\circ$$

This defines a monoidal functor *(realizing the *Timeless Games* programme initited by Baillot,Danos,Ehrard,Regnier 1998).*
Part VI

Full completeness
The scheduling criterion detects directed cycles.
The scheduling criterion does not detect non-directed cycles.

We thus elaborate a more subtle scheduling criterion.
Part VII

Thank you!