Session Types and Game Semantics
Synchrony and Asynchrony

Simon Castellan\textsuperscript{1}, Pierre Clairambault\textsuperscript{2}, Nobuko Yoshida\textsuperscript{1}

\textsuperscript{1}Imperial College London \hspace{1cm} \textsuperscript{2}LIP, ENS Lyon

14 April 2018
GaLoP 2018
The $\pi$-calculus [MPW92]

The $\pi$-calculus describes agents communicating through channels:

$$P, Q ::= 0$$ $$\mid (P \mid Q)$$ $$\mid (\nu ab)P$$ restriction $$\mid \ddagger !\ell \langle u \rangle . P$$ output $$\mid \sum_{i \in I} a?\ell(x_i).P_i$$ input $$\mid P + Q$$ nondet. choice

Communication: data ($\ell$) and channels ($u$).

**Short-hands:** $\ddagger \langle u \rangle := \ddagger ! \star \langle \ddagger u \rangle$ $\mid a(x) := a? \star (x)$
Game semantics for the $\pi$-calculus

Existing models:

- Laird [Lai05] refined by Tsukada & Sakayori [ST17] (for the asynchronous fragment)
- Hirschowitz et. al. [EHS15]

滨州 In this talk, focus on analyzing the first line of interpretation.
Game semantics for the $\pi$-calculus

Existing models:

- Laird [Lai05] refined by Tsukada & Sakayori [ST17] (for the asynchronous fragment)
- Hirschowitz et. al. [EHS15]

$\leadsto$ In this talk, focus on analyzing the first line of interpretation.

Basic idea: interpret channels as an effect like references:

$$[#T] = [T] \bot \times [T]$$

$$[(\nu a)P] = [P] \odot \text{cc}$$
Asynchrony: game semantics

Concurrent game semantics is traditionally asynchronous:

\[ \mathcal{C}_A \otimes \sigma = \sigma \xrightarrow{\text{courteous}} \sigma \]  

\[ \begin{array}{ccc}
B & B & B \\
q & q & q \\
\downarrow & \downarrow & \downarrow \\
tt & tt & tt
\end{array} \]

This forces some equations in the model:

\[ [\bar{a}\langle u \rangle . \bar{b}\langle v \rangle . P] = [\bar{b}\langle v \rangle . \bar{a}\langle u \rangle . P] \quad [a(x).b(y).P] = [b(y).a(x).P] \]

\[ \rightsquigarrow \text{Limits adequacy results} \ldots \]
Asynchrony: $\pi$-calculus [HT91]

Asynchrony in $\pi$-calculus: no continuation after sends.

$\vdash \bar{a}(u)\bar{b}(v)$ is not a term!

Moreover, in asynchronous $\pi$-calculus:

$$a(x).b(y).P \cong_{\text{may}} b(y).a(x).Q$$

$\vdash$ Models of [Lai05, ST17] adequate for may.
Asynchrony: $\pi$-calculus [HT91]

Asynchrony in $\pi$-calculus: no continuation after sends.
$\leadsto \bar{a}\langle u \rangle.\bar{b}\langle v \rangle$ is not a term!

Moreover, in asynchronous $\pi$-calculus:

$$a(x).b(y).P \simeq_{\text{may}} b(y).a(x).Q$$

$\leadsto$ Models of [Lai05, ST17] adequate for may.

However,

$$a(x).b(y).P \not\simeq_{\text{must}} b(y).a(x).Q$$

No adequacy possible for non-angelic testing equivalences . . .

$\Rightarrow$ Need to take synchrony seriously!
Session types [HVK98]

Typing discipline where types are protocols:

\[
S, T ::= \text{end } \\
\quad | \bigoplus_{i \in I} (\ell_i(S_i)). T_i \\
\quad | \&_{i \in I} (\ell_i(S_i)). T_i
\]

Typing \( \vdash P :: a_1 : S_1, \ldots, a_n : S_n \) ensures protocol preservation.

\[
\vdash P : a : T_k, \Delta \\
\vdash a!_{\ell_k(u)} P :: a : \bigoplus_{i \in I} (\ell_i(S_i)). T_i, \Delta, u : S_k
\]

Duality expresses compatible endpoints:

\[
\vdash P :: \Delta, a : S, b : S_\perp \\
\vdash (\nu a b) P :: \Delta
\]
This talk

Synchronous Processes

Concurrent Strategies

finite def up to $\equiv$

inadequate interpretation

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
This talk

Synchronous Processes

Courteous Processes

Concurrent Strategies

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
This talk

Synchronous Processes \(\xrightarrow{\text{(3) adequate interp.}}\) (3) Coincident strategies

\[\xymatrix{\text{Synchronous Processes} \ar[r] & \text{(3) Coincident strategies} \\
\text{Courteous Processes} \ar[u] \ar[r] & \text{Concurrent Strategies}}\]

\[\xymatrix{\text{Synchronous Processes} \ar[r] & \text{(3) Coincident strategies} \\
\text{Courteous Processes} \ar[u] \ar[r] & \text{Concurrent Strategies} \ar[u] \ar[l]}\]

\((A)synchrony\) in game semantics · C., Pierre Clairambault, Nobuko Yoshida
This talk

Synchronous Processes → Coincident strategies

Courteous Processes → Concurrent Strategies

adequate interp.

inadequate interpretation

finite def up to \( \simeq \)

adequate interp.

encoding

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
I. Session types into concurrent games
Types as games

In concurrent games, games are polarized event structures:

\[
\begin{array}{c}
\mathbb{B} \Rightarrow \mathbb{B} \\
q \\
\triangleleft \triangleleft \\
\triangledown \triangledown \\
\triangledown \triangledown \\
tt \sim \sim \ ff
\end{array}
\]
Types as games

In concurrent games, games are polarized event structures:

\[
\begin{align*}
\mathbb{B} & \Rightarrow \mathbb{B} \\
Q & \rightarrow \rightarrow \rightarrow \\
\triangleleft & \triangleleft \triangleleft \\
tt & \rightsquigarrow \rightsquigarrow ff \\
tt & \rightsquigarrow \rightsquigarrow ff
\end{align*}
\]

Interpretation of types is given by induction:

\[
\begin{align*}
& \left[ \&_{i \in I} \ell_i(S_i).T_i \right] = \sum_{i \in I} \ell_i \cdot (\llbracket S_i \rrbracket \parallel \llbracket T_i \rrbracket) \\
& \left[ \oplus_{i \in I} \ell_i(S_i).T_i \right] = \sum_{i \in I} \ell_i \cdot (\llbracket S_i \rrbracket^\perp \parallel \llbracket T_i \rrbracket)
\end{align*}
\]
Types as games

In concurrent games, games are polarized event structures:

\[
\begin{align*}
B & \;\Rightarrow\; B \\
\triangledown & \;\triangleleft\; \triangledown \\
tt & \;\sim\; ff
\end{align*}
\]

Interpretation of types is given by induction:

\[
\begin{align*}
[\&_{i\in I} l_i(S_i).T_i] &= \sum_{i\in I} l_i \cdot ([S_i] \parallel [T_i]) \\
[\oplus_{i\in I} l_i(S_i).T_i] &= \sum_{i\in I} l_i \cdot ([S_i]^{\perp} \parallel [T_i])
\end{align*}
\]

Lemma

Every tree-like game is the interpretation of a type.
Processes as strategies

Interpretation is by induction, eg.

\[
\begin{align*}
\vdash P : a : T_k, \Delta & \quad k \in I \\
\vdash a! \ell_k(u).P :: a : \oplus_{i \in I} \ell_i(S_i). T_i, \Delta, u : S_k
\end{align*}
\]

\(= \ell_k \cdot (\mathcal{C}_{[S_k]} \parallel [P]).\)

Restriction uses duality:

\[
\begin{align*}
\vdash P :: \Delta, a : S, b : S^\perp \\
\vdash (\nu ab) P :: \Delta
\end{align*}
\]

\(= [P] \odot \mathcal{C}_{[S]} .\)
Processes as strategies

Interpretation is by induction, eg.

\[
\begin{array}{c}
\vdash P : a : T_k, \Delta \\
\vdash a!\ell_k(u).P : a : \bigoplus_{i \in I} \ell_i(S_i), T_i, \Delta, u : S_k \\
\end{array}
\]

\[
= \ell_k \cdot (c_{[S_k]} \parallel [P]).
\]

Restriction uses duality:

\[
\begin{array}{c}
\vdash P : \Delta, a : S, b : S^\bot \\
\vdash (\nu ab)P : \Delta \\
\end{array}
\]

\[
= [P] \odot c_{[S]}.
\]

In general \([P]\) is not courteous, however we still get a sound model:

**Lemma**

If \(P \rightarrow Q\) then \([P] \preceq [Q]\).
Interestingly, if we have any strategy:

\[ \sigma : [\Delta] \quad \begin{array}{cccc} & 1 & 1 & \bot \\ \cdot & \triangle & \cdot & \oplus \\ \cdot & \cdot & \cdot & \ominus \end{array} \]
Interestingly, if we have any strategy:

\[ \sigma : \quad [\Delta] \quad 1 \quad 1 \quad \bot \]

\[ \begin{array}{c}
\hline
\cdot \\
\triangleleft \\
\cdot \\
\cdot \\
\cdot \\
\hline
\end{array} \]

\[ \begin{array}{c}
\oplus \\
\oplus \\
\ominus \\
\end{array} \]
Finite definability

Interestingly, if we have any strategy:

\[ \sigma' : [\Delta] \quad 1 \quad 1 \quad \bot \]

\( \sigma' \) is more tree-like than \( \sigma \)
Finite definability

Interestingly, if we have any strategy:

\[ \sigma' \odot [\text{join}] : [\Delta] \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]

\[ \vdots \]
Finite definability

Interestingly, if we have any strategy:

\[
\sigma' \ast [\text{join}] : \ [\Delta] \quad 1 \quad 1 \quad \bot
\]

\[
\sigma' \text{ is more tree-like than } \sigma \text{ and: }
\]

\[
\sigma' = \sigma \ast [\text{join}].
\]

**Theorem**

Every \( \sigma : [\Delta] \) is the interpretation of a process.
Inadequacy

\[(\nu a\bar{a})(\nu u\bar{u})(\nu u'\bar{v}')(\bar{a}\langle u, v \rangle | \bar{u} \bar{v} | a(x, y) y.x)\] deadlocks:

\[
\begin{align*}
\bar{u} & \quad \bar{a} & \quad a \\
\downarrow & \quad \downarrow & \quad \downarrow \\
\bar{v} & \quad u & \quad v & \quad y & \quad y \\
\downarrow & & \downarrow & & \downarrow \\
x
\end{align*}
\]
Inadequacy

\[(\nu a \bar{a})(\nu u \bar{u})(\nu' u' \bar{v}')(\bar{a} \langle u, v \rangle \mid \bar{u} \bar{v} \mid a(x, y) y.x)\] deadlocks:

\[
\begin{array}{c}
\bar{u} \\
\downarrow \\
\bar{v} \\
\end{array} \hspace{1cm}
\begin{array}{c}
\bar{a} \\
\downarrow \\
\bar{v} \\
\end{array} \hspace{1cm}
\begin{array}{c}
a \\
\downarrow \\
x \\
\end{array}
\]

In the model, copycat deals with communication and adds delay:

\[
\begin{array}{c}
\bar{u} \\
\downarrow \\
\bar{v} \\
\end{array} \hspace{1cm}
\begin{array}{c}
u \\
\downarrow \\
a_0 \\
\end{array} \hspace{1cm}
\begin{array}{c}
\bar{a} \\
\downarrow \\
a_1 \\
\end{array} \hspace{1cm}
\begin{array}{c}
a \\
\downarrow \\
x \\
\end{array}
\]

\[\leadsto\] No deadlock anymore.

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
Inadequacy

\[(νa\bar{a})(νu\bar{u})(νu'\bar{v}')(a\langle u, ν \rangle | \bar{u}.\bar{v} | a(x, y) y.x)\) deadlocks:

\[
\begin{align*}
\bar{u} & \quad \bar{a} & \quad a \\
\downarrow & & \downarrow \\
\bar{v} & \quad u & \quad v & \quad y \\
\downarrow & & \downarrow & \downarrow \\
& & \downarrow & x
\end{align*}
\]

In the model, copycat deals with communication and adds delay:

\[
\begin{align*}
\bar{u} & \quad u & \quad v \\
\downarrow & \downarrow & \downarrow \\
\bar{v} & \quad a_0 & \quad a_1 \\
\downarrow & \downarrow & \downarrow \\
& \quad y & \quad x
\end{align*}
\]

\[\rightsquigarrow \text{No deadlock anymore.}\]
II. Courteous processes

Synchronous Processes

Courteous process

Concurrent Strategies

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
Definition & Adequacy

A process $P$ is courteous when $\llbracket P \rrbracket$ is courteous.

Lemma

1. If $P \rightarrow Q$ and $P$ is courteous, then $Q$ is courteous
2. If $\llbracket P \rrbracket \prec \tau$ then $P \rightarrow Q$ with $\llbracket Q \rrbracket = \tau$
3. Every finite courteous $\sigma : [\Delta]$ is the interpretation of a courteous $P$
A strong link

From these results there is a strong correspondence between:

- The category of session types and courteous processes
- The category of games and strategies of [RW11, CCHW18]

Correspondence seems to play well with bisimulation & obs. eq.
A strong link

From these results there is a strong correspondence between:

- The category of session types and courteous processes
- The category of games and strategies of [RW11, CCHW18]

Correspondence seems to play well with bisimulation & obs. eq.

Hence:

- Session types and process provide a syntax for strategies
- Equivalent to interpret a language inside one or the other.
  (Generalizes [HO95] and [BHY01] to true concurrency and non-innocence)
III. Coincident strategies

Synchronous Processes $\rightarrow$ Coincident strategies

adequate interpretation

inadequate interpretation

Courteous process $\rightarrow$ Concurrent Strategies

adequate interpretation

finite def.

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
What is going on

Async forwarder. Given $S$, there is $\vdash [x = y] \bowtie x : S, y : \bar{S}$ with

$$[[x = y]] = c_{[S]}.$$
What is going on

**Async forwarder.** Given $S$, there is $\vdash [x = y] :: x : S, y : \bar{S}$ with

$$\llbracket [x = y] \rrbracket = \mathcal{E}_{\llbracket S \rrbracket}.$$ 

Our model interprets free output **indirectly**, indeed:

$$\llbracket \bar{a}(u) \rrbracket = \llbracket (\nu xy)(\bar{a}(x) \mid [y = u]) \rrbracket.$$ 

However $(\nu xy)(P(x) \mid [y = u]) \approx P(u)$ only if $P$ is courteous.

⇝ Change copycat to allow “coincidences” between $x$ and $y$. 

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
Coincident event structures

In event structures, event occurs separately of the others:

$$\emptyset \subseteq \{a_1\} \subseteq \{a_1, a_2\} \subseteq \ldots$$

\(^1\text{Known as Completeness and Stability.}\)
Coincident event structures

In event structures, event occurs separately of the others:

$$\emptyset \subseteq \{a_1\} \subseteq \{a_1, a_2\} \subseteq \ldots$$

Definition

A **coincident event structure** is a pair \((E, \mathcal{E})\) satisfying:\(^1\)

- if \(x, y \in \mathcal{E}\) bounded in \(\mathcal{E}\) then \(x \cup y \in \mathcal{E}\) and \(x \cap y \in \mathcal{E}\).

Covering chains are not sequences of events but of **coincidences**

$$\emptyset \subseteq X_1 \subseteq X_1 \cup X_2 \subseteq \ldots$$

[^1]: Known as *Completeness* and *Stability*.
Coincident event structures

In event structures, event occurs separately of the others:

\[ \emptyset \subseteq \{a_1\} \subseteq \{a_1, a_2\} \subseteq \ldots. \]

Definition

A coincident event structure is a pair \((E, \mathcal{E})\) satisfying:

1. if \(x, y \in \mathcal{E}\) bounded in \(\mathcal{E}\) then \(x \cup y \in \mathcal{E}\) and \(x \cap y \in \mathcal{E}\).

Covering chains are not sequences of events but of coincidences

\[ \emptyset \subseteq X_1 \subseteq X_1 \cup X_2 \subseteq \ldots. \]

Given a game \(A\), we can form the coincident copycat:

\[ \text{ccc}_A = (A^\perp \parallel A, \{x \parallel x \mid x \in \mathcal{C}(A)\}) \]

---

\(^1\)Known as Completeness and Stability.
Coincident strategies

Definition
A coincident strategy on $A$ is a map $S \rightarrow A$ such that its coincidence are singletons or of the form $\{a, b\}$.

$\Rightarrow$ A category without requiring courtesy!
Coincident strategies

Definition
A coincident strategy on $A$ is a map $S \rightarrow A$ such that its coincidence are singletons or of the form $\{a, b\}$.

$\rightsquigarrow$ A category without requiring courtesy!

We can now change the interpretation of free output:

$$
\begin{align*}
\vdash P : a : T_k, \Delta & \quad k \in I \\
\vdash a! \ell_k \langle u \rangle . P :: a : \bigoplus_{i \in I} \ell_i(S_i). T_i, \Delta, u : S_k \\
\end{align*}
$$

$\rightsquigarrow$ An adequate interpretation of synchronous session types. However: semantic space too broad (no finite definability).

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
IV. THE ENCODING

Synchronous Processes → Coincident strategies

Courteous process → Concurrent Strategies

finite def.

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
Two worlds

Synchronous Processes

Courteous Processes

Coincident strategies

Courteous Strategies

adeq.

finite def.

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
Two worlds

Synchronous Processes ← Courteous Processes

<table>
<thead>
<tr>
<th>adeq.</th>
<th>adeq.</th>
<th>finite def.</th>
</tr>
</thead>
</table>

| Coincident strategies ← Courteous Strategies |

But: diagram does not commute
Two worlds

Synchronous Processes

Coincident strategies

adeq.

Courteous Processes

Courteous Strategies

finite def.

adeq.

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida

project (sound but inadequate)
Two worlds

Synchronous Processes  Courteous Processes

adeq.  adeq.  finite def.

Coincident strategies  Courteous Strategies

encode (adequate)

Idea: add acknowledgements to protocols

(A)synchrony in game semantics · C., Pierre Clairambault, Nobuko Yoshida
Two worlds

Synchronous Processes \(\xrightarrow{\text{encode}}\) Courteous Processes

Coincident strategies \(\xrightarrow{\text{encode (adequate)}}\) Courteous Strategies

\text{adeq.} \quad \text{finite def.}
Definition

1. Unfold the protocol: \( A \to \uparrow A \)

\[
\begin{align*}
a & \quad \mapsto \quad \text{req}_a \\
\downarrow & \quad \text{ack}_a
\end{align*}
\]

2. Unfold the strategies: \( \sigma \to \uparrow \sigma \)

\[
\begin{align*}
a \to b & \quad \mapsto \quad \text{req}_a \to \text{req}_b \\
\downarrow & \quad \downarrow
\end{align*}
\]

\[
\begin{align*}
a \dashrightarrow b & \quad \mapsto \quad \text{req}_a \to \text{req}_b \\
\downarrow & \quad \downarrow
\end{align*}
\]

\[
\begin{align*}
a & \quad \dashrightarrow b \\
\text{ack}_a & \quad \text{ack}_b
\end{align*}
\]
Properties

- Encoding is **injective**:

  configurations of $\sigma \simeq \text{complete}$ configurations of $\uparrow \sigma$

- Should preserve and reflect weak bisimulation

  $\sigma \simeq \tau \quad \text{iff} \quad \uparrow \sigma \simeq \uparrow \tau$

- Characterisation of the image: **well-acknowledging** strategies.

  $\rightsquigarrow$ Coincident strategies $\simeq$ subcategory of courteous strategies
Summary & Perspectives

▸ We show a tight correspondence between Session Types and Game Semantics

▸ Benefits both communities:
  ▸ Provide a precise syntactic description of concurrent strategies
  ▸ Describes the causal behaviour of session processes

Future work.

▸ Extend to the nonlinear setting.
  ~ A language for innocent concurrent strategies.

▸ Extend session types to non-tree-like protocols.
Martin Berger, Kohei Honda, and Nobuko Yoshida. 
Sequentiality and the pi-calculus. 

Simon Castellan, Pierre Clairambault, Jonathan Hayman, and Glynn Winskel. 
Non-angelic concurrent game semantics. 
2018. 
Accepted at FOSSACS’18.

Clovis Eberhart, Tom Hirschowitz, and Thomas Seiller. 
An intensionally fully-abstract sheaf model for pi. 

Pi-calculus, dialogue games and PCF.
Jim Laird.
A game semantics of the asynchronous pi-calculus.

Paul-André Melliès and Samuel Mimram.
Asynchronous games: Innocence without alternation.

Silvain Rideau and Glynn Winskel.
Concurrent strategies.
Ken Sakayori and Takeshi Tsukada.
A truly concurrent game model of the asynchronous $\pi$-calculus.