Using session types to understand the gap between synchronous syntaxes and asynchronous game semantics.

Simon Castellan[1], Pierre Clairambault[2], Nobuko Yoshida[1]

[1]: Imperial College, London, UK
[2]: Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP, Lyon, France

Game semantics for concurrency is traditionally asynchronous: the order between moves of the same polarity is not guaranteed to be respected unless this order arises from justification in the game. Technically, this takes the form of the closure under an order (saturation condition) in the interleaving setting [Laird, 2005] and as courtesy in the truly concurrent setting [Mimram and Melliès, 2007]. These conditions are required to obtain a categorical structure: copycat being asynchronous, strategies that are invariant under composition with copycat are necessarily courteous [Rideau and Winskel, 2011].

In the world of syntax, and message-passing calculi, however, these restrictions have been given little consideration. Even the asynchronous pi-calculus, is too synchronous for game semantics. Game semantics models of asynchronous pi calculus can only be adequate for may testing [Laird, 2005], but more fine-grained equivalence (such as barbed congruence or fair testing) distinguishes:

```
a(x). b(y). P     and    b(y). a(x). P
```

that these models equate.

The only work we are aware that models interactively synchronous calculus is [Hirschowitz and Eberhart and Seiller, 2015], using presheaves over plays represented as string diagrams. However, this model does not define composition and uses ad-hoc play structures.

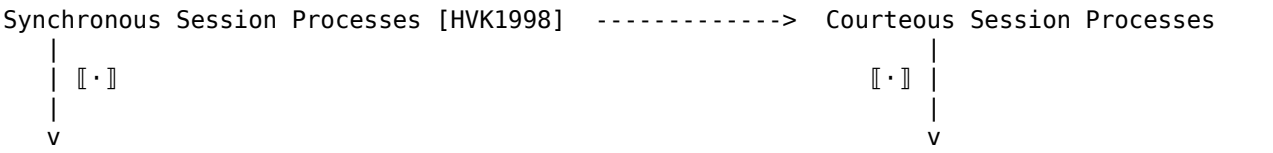
The talk will present a work in progress, aiming at understanding the gap between synchronous session typed processes [Honda and Vasconcelos and Kubo, 1998] and asynchronous causal game semantics [RW, 2011]. First, we start by showing that the semantic condition of courtesy can be internalised in the π-calculus by a simple syntactic condition, defining a subcalculus of courteous π-term. This class of processes can be faithfully embedded in strategies:

- Session types are in bijection with tree-like games
- Every finite strategy is the interpretation (up to iso) of a term.
- Interpretation is fully abstract for weak bisimulation.

This close correspondance makes these courteous session typed processes as a good syntax for courteous strategies; or strategies as a free model for courteous processes.

We then investigate an encoding on types and processes such that if P : A, then ↑P : ↑A is a *asynchronous* process. This encoding, type-driven, is different from the standard encoding of the synchronous π-calculus into the asynchronous π-calculus [Honda and Tokoro, 1991].

Finally, we explore the question of "synchronous strategies" and try to replicate this syntactic encoding at the semantic level in order to close the square:



[Honda and Tokoro, 1991]: An Object Calculus for Asynchronous Communication, ECOOP 1991

[Honda and Vasconcelos and Kubo 1998]: Language Primitives and Type Disciplines for Structured Communication-based Programming, ESOP 1998.

[Laird, 2005]: A Game Semantics of the Asynchronous π-Calculus, CONCUR 2005.

[Mimram and Melliès, 2007]: Asynchronous Games: Innocence Without Alternation, CONCUR 2007.

[Ghica and Murawski, 2008]: Angelic semantics of fine-grained concurrency, Annals of Pure and Applied Logic (2008).

[Rideau and Winskel, 2011]: Concurrent strategies.

[Hirschowitz and Eberhart and Seiller, 2015]: An intensionally fully-abstract sheaf model for pi, CALCO 2015