

Probabilistic Methods in Concurrency

Lecture 2

The pi-calculus hierarchy: encodings.

Catuscia Palamidessi

catuscia@lix.polytechnique.fr

www.lix.polytechnique.fr/~catuscia

Page of the course:

www.lix.polytechnique.fr/~catuscia/teaching/Pisa/

Expressive power of π_a wrt π

- Clearly the (synchronous) π -calculus is more expressive than the asynchronous π -calculus. In fact, the latter is practically a subset of the former.

Indeed, the output action can be seen as the output-prefix process with continuation 0. This relation is a strong bisimulation:

$$\bar{x}y \sim \bar{x}y.0$$

- What about the opposite direction?
- In general, in order to compare the expressive power of two languages, we look for the existence/non existence of an **encoding** with certain properties among these languages
- What is a good notion of encoding to be used as basis to measure the relative expressive power?

A "good" notion of encoding

In general we would be happy with an encoding $\llbracket \cdot \rrbracket : \pi \rightarrow \pi_a$ being:

- Compositional wrt the operators $\llbracket P \text{ op } Q \rrbracket = C_{op}[\llbracket P \rrbracket, \llbracket Q \rrbracket]$
- (Preferably) homomorphic wrt | (distribution-preserving) $\llbracket P \mid Q \rrbracket = \llbracket P \rrbracket \mid \llbracket Q \rrbracket$
- Preserving some kind of semantics. Here there are several possibilities
 - Preserving observables $Obs(P) = Obs(\llbracket P \rrbracket)$
 - Preserving equivalence $\llbracket P \rrbracket \text{ equiv } \llbracket Q \rrbracket \Rightarrow P \text{ equiv}' Q$ (soundness)
 $\llbracket P \rrbracket \text{ equiv } \llbracket Q \rrbracket \Leftarrow P \text{ equiv}' Q$ (completeness)
 $\llbracket P \rrbracket \text{ equiv } \llbracket Q \rrbracket \Leftrightarrow P \text{ equiv}' Q$ (full abstraction, correctness)

This is one of the most popular requirements for an encoding. However it is not clear how it relates to the notion of expressive power.

- (Preferably) the encoding should not introduce divergences, in the sense that if in the original process all the computations converge, then the same holds for its translation. Note that weak bisimulations are insensitive wrt divergences

Encoding the output prefix

- The encoding of Boudol

Boudol [1992] provided the following encoding of π (without choice) into π_a : The idea is to force both partners to proceed only when it is sure that the communication can take place, by using a sort of rendez-vous protocol

- $\llbracket \bar{x}y.P \rrbracket = (\nu z)(\bar{x}z \mid (z(w).\bar{w}y \mid \llbracket P \rrbracket))$

- $\llbracket x(y).Q \rrbracket = x(z).(\nu w)(\bar{z}w \mid w(y).\llbracket Q \rrbracket)$

$\llbracket \cdot \rrbracket$ is homomorphic for all the other operators. Namely:

- $\llbracket 0 \rrbracket = 0$

- $\llbracket P \mid Q \rrbracket = \llbracket P \rrbracket \mid \llbracket Q \rrbracket$

- $\llbracket (\nu x)P \rrbracket = (\nu x)\llbracket P \rrbracket$

- $\llbracket ! P \rrbracket = ! \llbracket P \rrbracket$

- Boudol proved this encoding sound wrt the Morris ordering

- **Exercise.** Define an encoding which takes only two steps instead than three. (Such a kind of encoding was defined by Honda-Tokoro [1992].)

Encoding the output prefix

- The encoding of Honda-Tokoro

Honda-Tokoro [1992] defined the following encoding of π (without choice) into π_a in which the communication protocol takes two steps instead than three. The idea is to let the receiver take the initiative (instead than the sender)

- $\llbracket \bar{x}y.P \rrbracket = x(z).(\bar{z}y \mid \llbracket P \rrbracket)$

- $\llbracket x(y).Q \rrbracket = (\nu z)(\bar{x}z \mid z(y).\llbracket Q \rrbracket)$

$\llbracket \cdot \rrbracket$ is homomorphic for all the other operators. Namely:

- $\llbracket 0 \rrbracket = 0$

- $\llbracket P \mid Q \rrbracket = \llbracket P \rrbracket \mid \llbracket Q \rrbracket$

- $\llbracket (\nu x)P \rrbracket = (\nu x)\llbracket P \rrbracket$

- $\llbracket ! P \rrbracket = ! \llbracket P \rrbracket$

- Honda proved this encoding sound and "almost" complete wrt the a certain logical semantics
- Honda-Tokoro defined also another encoding of π (without choice) into a polyadic version of π_a in which the communication protocol takes two steps and the sender takes the initiative. **Exercise:** find out how to define such an encoding.

Properties of output encodings wrt testing

- Definition of testing semantics:
 - A process P **may** satisfy a test T (notation $P \text{ may } T$) iff there exists a computation of $[P | T]$ which reaches a state where the action ω (a special action of the test) is enabled.
 - A process P **must** satisfy a test T (notation $P \text{ must } T$) iff every computation of $[P | T]$ reaches a state where the action ω (a special action of the test) is enabled.
 - $P \sqsubseteq_{\text{may}} Q$ iff for every test T , if $P \text{ may } T$ then $Q \text{ may } T$
 - $P \sqsubseteq_{\text{must}} Q$ iff for every test T , if $P \text{ must } T$ then $Q \text{ must } T$
 - $P \simeq_X Q$ iff $P \sqsubseteq_{\text{must}} Q$ and $Q \sqsubseteq_{\text{must}} P$, $X = \text{may}, \text{must}$
- In contrast to weak bisimulation, testing semantics is sensitive wrt divergency
- We don't expect the encodings of output prefix to be correct wrt testing semantics (why?), but we would like the encoding to satisfy at least the following properties :
 - $P \text{ may } T$ iff $\llbracket P \rrbracket \text{ may } \llbracket T \rrbracket$
 - $P \text{ must } T$ iff $\llbracket P \rrbracket \text{ must } \llbracket T \rrbracket$

Properties of output encodings wrt testing

- The encodings of Boudol and Honda-Tokoro
 - Verify $P \text{ may } T \text{ iff } \llbracket P \rrbracket \text{ may } \llbracket T \rrbracket$
 - Do not verify $P \text{ must } T \text{ iff } \llbracket P \rrbracket \text{ must } \llbracket T \rrbracket$
(they preserve may testing but not must testing)
- **Theorem** [Cacciagrano, Corradini and Palamidessi, 2004] Let $\llbracket \cdot \rrbracket$ be an encoding of π (without choice) into π_a such that:
 - $\llbracket \cdot \rrbracket$ is compositional wrt the prefixes
 - There exists a P such that $\llbracket P \rrbracket$ divergesthen $\llbracket \cdot \rrbracket$ does not preserve must testing

The problem however is uniquely a problem of fairness:

- **Theorem** [Cacciagrano, Corradini and Palamidessi, 2004] The encodings of Boudol and Honda-Tokoro
 - A) preserve must testing if we restrict to fair computations only
 - B) preserve fair must testing

Encoding internal choice in π_a

The blind choice (or internal choice) construct $P \oplus Q$ has the following semantics

$$\frac{}{P \oplus Q \xrightarrow{\tau} P} \quad \frac{}{P \oplus Q \xrightarrow{\tau} Q}$$

In π this operator can be represented by the construct $\tau.P + \tau.Q$

Exercise: Let π^- be π without the $+$ operator, and π^\oplus be π where the $+$ operator can only occur in the context of (a construct representing) a blind choice. Give an encoding $\llbracket \cdot \rrbracket : \pi^\oplus \longrightarrow \pi^-$ such that $\forall P \llbracket P \rrbracket \sim P$

Encoding input-guarded choice in π_a

- Input-guarded choice is a construct of the form: $\sum_{i \in I} x_i(y_i).P_i$
- Let π^i be π where $+$ can only occur in an input-guarded choice. The following encoding of π^i into π_a was defined by Nestmann and Pierce [1996]

$$\llbracket \sum_{i \in I} x_i(y_i)P_i \rrbracket = (\nu l)(\bar{l} \text{ true} \mid \prod_{i \in I} \text{Branch}_{\ell_i})$$

$$\text{Branch}_{\ell_i} = x_i(z_i).\ell(w).(if \quad w = \text{true} \\ \text{then } (\bar{l} \text{ false} \mid \llbracket P_i \rrbracket) \\ \text{else } (\bar{l} \text{ false} \mid \bar{x}_i z_i))$$

- Nestmann and Pierce proved that his encoding is fully abstract wrt coupled bisimulation, and it does not introduce divergences.