What is the right structural congruence for the (Reversible) Calculus of Communicating Systems?

11th International School on Rewriting

Clément Aubert[1] Ioana-Domnina Cristescu[2]

[1]Augusta University – School of Computer & Cyber Sciences

[2]INRIA – TAMIS team

Paris, 1-6 July 2019

Goal

Specifying Reversible Concurrent Computation

- What?
  Concurrent (multiprocessing, parallel, distributed, etc.) computation that can backtrack. Memory needs to be "enough", "not too big", **and** distributed.
- Why?
  - Combine all the *benefits* of reversible *and* concurrent computation!
  - But also all the *difficulties* . . .
  - Network of reversible computers!
- How?
  Reversing process calculi, reversible event structures, etc.

Goal

Specifying Reversible Concurrent Computation

RCCS
adds
Reversibility
to the
Calculus of **C**ommunicating **S**ystems

CCS System

1. Operators:

$$P, Q \coloneqq \lambda.P \;\Big|\; \sum_{i \in I} P_i \;\Big|\; A \;\Big|\; P \mid Q \;\Big|\; P \backslash a \;\Big|\; P[a \leftarrow b] \;\Big|\; 0$$

2. Labeled Transition System:

$$\frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}, \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'},$$

$$\frac{P \xrightarrow{\lambda} P' \qquad Q \xrightarrow{\overline{\lambda}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}, \qquad \text{etc.}$$

3. Structural Equivalence:

$P \mid 0 \equiv P$, $\quad P \mid Q \equiv Q \mid P$, $\quad P + Q \equiv Q + P$, $\quad$ etc.

RCCS System

1. Operators:

$$T \coloneqq m \triangleright P \qquad \text{(Reversible Thread)}$$

$$R, S \coloneqq T \;\Big|\; R \mid S \;\Big|\; R \backslash a \qquad \text{(RCCS Processes)}$$

2. Labeled Transition System:

$$\frac{}{m \triangleright \lambda.P \xrightarrow{i:\lambda} \langle i, \lambda, 0 \rangle.m \triangleright P} \text{'} \qquad \frac{}{\langle i, \lambda, 0 \rangle.m \triangleright P \xrightarrow{i:\lambda}_* m \triangleright \lambda.P} \text{'}$$

etc.

3. Structural Equivalence:

$$m \triangleright (P \mid Q) \equiv (\curlyvee.m \triangleright P) \mid (\curlyvee.m \triangleright Q)$$

But hold on

1. Isn’t that mixing the syntactical sugar and the system?
2. How come the congruence does not include e.g. $R \mid S \equiv S \mid R$?
3. How do we know it’s the right $\equiv$?

Lemma

If $P \xrightarrow{\alpha} P'$ with the "pure" LTS and $P \equiv Q$ then $Q \xrightarrow{\alpha} Q'$ with the "sweetened" LTS and $P' \equiv Q'$.

Semantics

$\forall P, Q, [\![P]\!] \cong [\![Q]\!] \iff P \equiv Q$

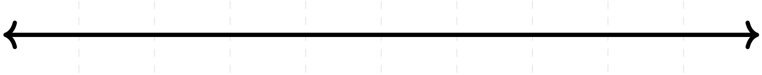
Syntactics

Every term P has a "normal form".

Lemma

If $P \xrightarrow{\alpha} P'$ with the "pure" LTS and $P \equiv Q$ then $Q \xrightarrow{\alpha} Q'$ with the "sweetened" LTS and $P' \equiv Q'$.

Where are we?



$\forall P, Q, P \not\equiv Q$ $\qquad\qquad$ $\forall P, Q, P \equiv Q$

Semantics

$\forall P, Q, [\![P]\!] \cong [\![Q]\!] \iff P \equiv Q$

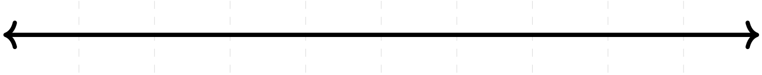
Syntactics

Every term P has a "normal form".

Lemma

If $P \xrightarrow{\alpha} P'$ with the “pure” LTS and $P \equiv Q$ then $Q \xrightarrow{\alpha} Q'$ with the “sweetened” LTS and $P' \equiv Q'$.

Where are we?



$\forall P, Q, P \not\equiv Q$ $\qquad$ $\forall P, Q, P \equiv Q$

Semantics

$\forall P, Q, [\![P]\!] \cong [\![Q]\!] \not\Leftrightarrow P \equiv Q$

No! Usually, $[\![P + 0]\!] \cong [\![P]\!]$.

Syntactics

Every term P has a “normal form”.

Lemma

If $P \xrightarrow{\alpha} P'$ with the "pure" LTS and $P \equiv Q$ then $Q \xrightarrow{\alpha} Q'$ with the "sweetened" LTS and $P' \equiv Q'$.

Where are we?

$\forall P, Q, P \not\equiv Q$ $\forall P, Q, P \equiv Q$

Semantics

$\forall P, Q, [\![P]\!] \cong [\![Q]\!] \;\not\Longleftrightarrow\; P \equiv Q$

No! Usually, $[\![P + 0]\!] \cong [\![P]\!]$.

Syntactics

Every term P has a "normal form".

So what?