PLACEHOLDER

Julien Gabet

Mars-June, 2018

Introduction

This is an interesting introduction to an interesting subject.

- state of the art?
- what we want: have terms wear the proof of a working reduction, which is proof they're well built.
- how we do it: introducing an annotated calculus, which annotations will allow us to easily type it by allowing interactions between two terms over one variable only. Defining the system well allows us to project to usual π-calculus, and maybe to compare with existing type systems for it.
- we would like the reduction to be confluent, the typing system to be without cut or with a cut elimination property (the latter holds), and the reduction with the typing system to preserve typing (it does).

1 Definitions

We first need to define the language of terms of our annotated π-calculus.

Definition 1

Take two countable sets, $N := \{t, u, v, \ldots\}$ the set of names and $V := \{x, y, z, \ldots\}$ the set of variables.
The annotated terms are defined by the following grammar:

$$\begin{array}{lr}
P, Q ::= x \leftrightarrow y \;\; ; \;\; 0_x \;\; ; & x, y \in V \quad \textit{base terms} \\
\qquad \epsilon_x.P \;\; ; \;\; \lambda_x y.P \;\; ; & x, y \in V \quad \textit{variable introduction and modality prefixes} \\
\qquad u_x(t).P \;\; ; \;\; \bar{u}_x\langle v\rangle.P \;\; ; & t, u, v \in N, x \in V \quad \textit{action prefixes} \\
\qquad P|_x Q \;\; ; \;\; P||_x Q \;\; ; & x \in V \quad \textit{parallel and synchronization} \\
\qquad (\nu u)P & u \in N \quad \textit{name binding prefix}
\end{array}$$

The synchronization rule will be used to guide the behavior of terms, thus we define a reduction system over this construction specifically, as follows:

Definition 2

We define a reduction rule for synchronization (that does not hold under action prefixes) as follows:

$$\begin{array}{lr}
\epsilon_x.P||_x 0_x \to P & \textit{symmetric in } || \quad 1 \\
P||_x x \leftrightarrow y \to P[y/x] & \textit{symmetric in } || \quad 2 \\
(P|_x Q)||_x \lambda_x y.R \to P||_x (Q[y/x]||_y R) & 3a \\
\lambda_x y.R||_x (P|_x Q) \to (R||_x P)||_y Q[y/x] & 3b \\
\bar{u}_x\langle v\rangle.P||_x u_x(t).Q \to P||_x Q[v/t] & \textit{symmetric in } || \quad 4
\end{array}$$

We also define rules for commuting the synchronization with other rules, at first not holding under action prefixes:

$$\begin{array}{lr}
(P|_x Q)||_y R \geq (P||_y R)|_x Q & \textit{symmetric in } ||,\ y \notin fv(Q) \quad 5a \\
(P|_x Q)||_y R \geq P|_x (Q||_y R) & \textit{symmetric in } ||,\ y \notin fv(P) \quad 5b \\
\epsilon_x.P||_y Q \geq \epsilon_x.(P||_y Q) & \textit{symmetric in } || \quad 6a \\
\lambda_x y.P||_z Q \geq \lambda_x y.(P||_z Q) & \textit{symmetric in } ||,\ y \notin fv(Q) \quad 6b \\
(\nu u)P||_x Q \geq (\nu u)(P||_x Q) & \textit{symmetric in } ||,\ u \notin fn(Q) \quad 6c
\end{array}$$

And we extend it into $\gtrsim$ as follows, allowing for it to act on (and under) action prefixes:

$$\begin{array}{lr}
u_x(t).P||_y Q \gtrsim u_x(t).(P||_y Q) & \textit{symmetric in } || \quad 6d \\
\bar{u}_x\langle v\rangle.P||_y Q \gtrsim \bar{u}_x\langle v\rangle.(P||_y Q) & \textit{symmetric in } || \quad 6e
\end{array}$$

Relation $\to$ is also extended into $\rightsquigarrow$ by allowing it to act under action prefixes (no specific rule added).

Remark: In rules 5 and 6, some parts have strong requirements, such as a variable not being present in one or more subterms. Failure to meet those requirements can lead to terms that do not reduce past a certain point. See an example below of a term annotated to be able to reduce, and the same term with annotations changed that fails to do so.

INCLUDE EXAMPLE HERE

We would like these arrows to have a confluent behavior. For that, we need equivalences, that we will define below.

Definition 3

We define a congruence rule, that does not act on action prefixes:

$(P\|_xQ)\|_yR \equiv (P\|_yR)\|_xQ$	$y \notin fv(Q), x \notin fv(R)$	a_1
$P\|_x(Q\|_yR) \equiv Q\|_y(P\|_xR)$	$y \notin fv(P), x \notin fv(Q)$	a_2
$(P\|_xQ)\|_yR \equiv P\|_x(Q\|_yR)$	$y \notin fv(P), x \notin fv(R)$	b
$P\|_x\alpha_y.Q \equiv \alpha_y.(P\|_xQ)$	*symmetric in* $\|, \alpha. \in \{\epsilon., \lambda.z\}, y, t \notin fv(P)$	c
$\alpha_x.\beta_y.P \equiv \beta_y.\alpha_x.P$	$\alpha., \beta. \in \{\epsilon., \lambda.z\}, x \neq y$	d
$(\nu u)P\|_xQ \equiv (\nu u)(P\|_xQ)$	*symmetric in* $\|, u \notin fn(Q)$	e
$(\nu u)\alpha_x.P \equiv \alpha_x.(\nu u)P$	$\alpha. \in \{\epsilon., \lambda.z\}$	f
$x \leftrightarrow y \equiv y \leftrightarrow x$		g

And we also extend it into $\cong$ *by allowing* α, β *to be action prefixes in rules* c, d:

$$\alpha., \beta. \in \{\epsilon., \lambda.z, u.(t), \bar{u}.\langle v\rangle\};$$

as well as in rule f:

$$\alpha. \in \{\epsilon., \lambda.z, v.(t), \bar{v}.\langle w\rangle, u \neq v, u \neq t, u \neq w\}.$$

As said before, the first property we would like to have is the confluence of our arrows, that act as reduction rules in our annotated system. That is done for each one up to the corresponding equivalence defined above.

Proposition 4

Changes to go here: $\rightarrow$ *is strictly confluent,* $\geq$ *is confluent up to* $\equiv$ *and is a simulation for* $\rightarrow$*. Maybe 2 different propositions? Also rewrite the proof to go with the new propositions.*
Relation $\rightarrow$ *is confluent up to* $\equiv$*, and relation* $\rightsquigarrow$ *is confluent up to* $\cong$

$\triangleright$ Most cases are treated in the $\rightarrow$ rule with $\equiv$, and we can note that rules 1 through 4 interacting with any rule means that one of the rules is a sub-term of the other. That means that, should we choose to reduce the inner rule first, we only replace the subterm in question with the result of the innermost rule, and should we choose to reduce the outer rule first, the innermost rule can still be applied in the subterm that has not been touched (worst case scenario would be a substitution in this term, but that does not affect the application of the rule). Cases where both choices lead to non-strictly equal terms are rules 5 and 6 interacting with themselves or each other. We detail them below: Rule 5 against rule 5: there are 4 cases, depending on where the variable cut against is situated. The variable is noted as an exponent in the terms where it appears, and only 2 cases are treated (as the other two are their complements, and are treated the same):

$$\begin{tikzcd}
 & (P^y|_xQ)||_y(R^y|_zS) \arrow[dl, dashed, "\geq_{5a},\, left"] \arrow[dr, dashed, "\geq_{5a},\, right"] & \\
(P^y||_y(R^y|_zS))|_xQ \arrow[d, dashed, "\geq_{5a}"] & & ((P^y|_xQ)||_yR^y)|_zS \arrow[d, dashed, "\geq_{5a}"] \\
((P^y||_yR^y)|_zS)|_xQ & \equiv_{a_1} & ((P^y||_yR^y)|_xQ)|_zS
\end{tikzcd}$$

$$\begin{tikzcd}
 & (P^y|_xQ)||_y(R|_zS^y) \arrow[dl, dashed, "\geq_{5a},\, left"] \arrow[dr, dashed, "\geq_{5b},\, right"] & \\
(P^y||_y(R|_zS^S))|_xQ \arrow[d, dashed, "\geq_{5b}"] & & R|_z((P^y|_xQ)||_yS^y) \arrow[d, dashed, "\geq_{5a}"] \\
(R|_z(P^y||_yS^y))|_xQ & \equiv_b & R|_z((P^y||_yS^y)|_xQ)
\end{tikzcd}$$

Rule 6 against rule 6 works the same, here is an example with ϵ and ν:

$$\begin{tikzcd}
 & \epsilon_x.P||_y(\nu u)Q \arrow[dl, dashed, "\geq_{6a}"] \arrow[dr, dashed, "\geq_{6c}"] & \\
\epsilon_x.(P||_y(\nu u)Q) \arrow[d, dashed, "\geq_{6c}"] & & (\nu u)(\epsilon_x.P||_yQ) \arrow[d, dashed, "\geq_{6a}"] \\
\epsilon_x.(\nu u)(P||_yQ) \arrow[r, phantom, "\equiv_f"] & & (\nu u)\epsilon_x.(P||_yQ)
\end{tikzcd}$$

Other cases in the possible 6 against 6 rules are treated in the exact same manner. The last set of cases is a rule 5 against a rule 6. Those are all treated the same way as well, so we only treat two examples (one for 5a and one for 5b):

$$\begin{tikzcd}
 & (P^y|_xQ)||_y(\nu u)R \arrow[dl, dashed, "\geq_{5a}"] \arrow[dr, dashed, "\geq_{6c}"] & \\
(P^y||_y(\nu u)R)|_xQ \arrow[d, dashed, "\geq_{6c}"] & & (\nu u)((P^y|_xQ)||_yR) \arrow[d, dashed, "\geq_{5a}"] \\
(\nu u)(P^y||_yR)|_xQ \arrow[r, phantom, "\equiv_e"] & & (\nu u)((P^y||_yR)|_xQ)
\end{tikzcd}$$

$$\begin{tikzcd}
 & (P|_xQ^y)||_y(\nu u)R \arrow[dl, dashed, "\geq_{5b}"] \arrow[dr, dashed, "\geq_{6c}"] & \\
P|_x(Q^y||_y(\nu u)R) \arrow[d, dashed, "\geq_{6c}"] & & (\nu u)((P|_xQ^y)||_yR) \arrow[d, dashed, "\geq_{5b}"] \\
P|_x(\nu u)(Q^y||_yR) \arrow[r, phantom, "\equiv_e"] & & (\nu u)(P|_x(Q^y||_yR))
\end{tikzcd}$$

The specific cases added by $\rightsquigarrow$ and $\gtrsim$ are treated the exact same way, and are confluent with $\cong$. $\square$

Below maybe not needed after above rewrite

We also need to be sure the equivalence does not change the reduction, *ie.* that two equivalent terms have the same reductions. That is assured by showing the equivalence is a bisimulation relation.
DO WE NEED TO REMIND THE DEFINITION OF A (BI)SIMULATION HERE?

Proposition 5

Relation $\equiv$ is a bisimulation for reduction $\geq^? \rightarrow \geq^*$.

2 Typing decorated terms with MLL

Section intro here, stating MLL seems a good system to type terms of the decorated calculus

Definition 6

The typing system over MLL is given by the following rules:

Rules for neutral elements:

$$\frac{}{0_x \vdash x : 1}\ \mathit{NOP} \qquad \frac{P \vdash \Gamma}{\epsilon_x.P \vdash \Gamma, x : \bot}\ \mathit{BOT} \qquad \frac{}{x \to y \vdash x : E^\perp, y : E}\ \mathit{SUB}$$

Construction rules:

$$\frac{P \vdash \Gamma, x : E \quad Q \vdash \Delta, x : F}{P|_x Q \vdash \Gamma, \Delta, x : E \otimes F}\ \mathit{PARA} \qquad \frac{P \vdash \Gamma, x : E, y : F}{\lambda_x y.P \vdash \Gamma, x : E \parr F}\ \mathit{LAM} \qquad \frac{P \vdash \Gamma, x : E \quad Q \vdash \Delta, x : E^\perp}{P||_x Q \vdash \Gamma, \Delta}\ \mathit{CUT}$$

Action rules:

$$\frac{P \vdash \Gamma, x : A[v/t]^\perp}{\bar{u}_x\langle v\rangle.P \vdash \Gamma, x : \exists_u t.A^\perp}\ \mathit{IN} \qquad \frac{P \vdash \Gamma, x : A \quad t \notin \Gamma}{u_x(t).P \vdash \Gamma, x : \forall_u t.A}\ \mathit{OUT}$$

Nu rule:

$$\frac{P \vdash \Gamma \quad u \notin \Gamma}{(\nu u)P \vdash \Gamma}\ \mathit{NU}$$

One important thing to note here, is that for PARA and CUT rules, Γ and Δ are disjoint. The other important thing is that the synchronization construction becomes a CUT operation in the typing system. This rule has elimination transformations, given by the arrows $\to$ and $\rightsquigarrow$ defined in 2. Properties we would like to emerge from such a construction would be that the arrows preserve typing, and that their application for cut elimination terminates.

Proposition 7

Cut elimination holds and terminates, for both $\to$ under $\equiv$ and $\rightsquigarrow$ under $\cong$.

▷ proof here □

3 Links with the usual π-calculus

Definition 8

Define π-calculus

Definition 9

Define projection $\lfloor\cdot\rfloor$

Proposition 10

Something about $\rightarrow$ and $\equiv$ related to projection

$\triangleright$ proof here $\square$

Open about $\rightsquigarrow$ here or in the conclusion.

Conclusion

A nice conclusion and working trails for later here.