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:

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

Remark: the parallel and synchronization constructions behave differently on their variable. The parallel rule gives a term that puts together the two instances of the variable observed, and keeps it available for further use, while the synchronization rule uses both instances of the variable observed and thus has it unavailable for further use in the created term. The synchronization is a binding operator.

Remark: the lambda is also binding for its second variable (y here), keeping availability for its first variable only. The ϵ, on the opposite, creates its variable, that does not exist in the base term.

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:

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

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, when no other rule can apply. 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. First, we remark that $\to$ is strongly confluent. Then, we will need an equivalence relation to be defined for the confluence of $\geq$.

Proposition 3

Relation $\rightarrow$ is strongly confluent.

$\triangleright$ Since a synchronization makes the observed variable unavailable to the term, and because $\rightarrow$ does not allow for an arbitrary term on any side but only explicitly fixed constructions, and not allowing for these constructions to be a synchronization, we can remark that two reductions using $\rightarrow$ in the same term cannot interfere with each other. Thus, a term capable of two reductions using $\rightarrow$ has one of them be either a strict subterm of the other, or distinct parts of a more general term. The reductions then do not interfere with each other, and can be followed in any order. $\square$

Remark: the $\rightsquigarrow$ relation works the same (but with reductions under action prefixes), and so has the same property and proof.

Remark: interaction of $\rightarrow$ and $\geq$ are commutative as well, because the specificity of the constructions in the $\rightarrow$ rule makes it so it does not interact with $\geq$ rules. As such, the following property is immediate:

Proposition 4

Relation $\geq$ is a simulation for $\rightarrow$.

We remind the definitions of a simulation ans a bisimulation, as those tools are uncommon enough to justify being introduced here:

Definition 5

- *A simulation $\mathcal{R}$ is a binary relation on terms such that, for all terms P, Q, if $P\mathcal{R}Q$ then forall $P \rightarrow P'$ there exists a term Q' such that $Q \rightarrow Q'$ and $P'\mathcal{R}Q'$.*
- *A bisimulation $\mathcal{R}$ is a binary relation on terms such that $\mathcal{R}$ and $\mathcal{R}^{-1}$ are simulations.*

The $\geq$ relation in itself has a bit more complexity, as two instances can interact with each other. A congruence is needed to make this relation confluent.

Definition 6

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

$(P\vert_x Q)\vert_y R \equiv (P\vert_y R)\vert_x Q$	$y \notin fv(Q), x \notin fv(R)$	a_1
$P\vert_x (Q\vert_y R) \equiv Q\vert_y (P\vert_x R)$	$y \notin fv(P), x \notin fv(Q)$	a_2
$(P\vert_x Q)\vert_y R \equiv P\vert_x (Q\vert_y R)$	$y \notin fv(P), x \notin fv(R)$	b
$P\vert_x \alpha_y.Q \equiv \alpha_y.(P\vert_x Q)$	*symmetric in* $\vert, \alpha. \in \{\epsilon., \lambda.z\}, y, t \notin fv(P)$	c
$\alpha_x.\beta_y.P \equiv \beta_y.\alpha_x.P$	$\alpha., \beta. \in \{\epsilon., \lambda.z\}, x \neq y$	d
$(\nu u)P\vert_x Q \equiv (\nu u)(P\vert_x Q)$	*symmetric in* $\vert, u \notin fn(Q)$	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:

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

This equivalence allows for the confluence of $\geq$:

Proposition 7

Relation $\geq$ is confluent up to $\equiv$.

$\triangleright$ There are 3 possible cases here, for two groups of rules in the $\geq$ relation.
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:

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

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 $5b$):

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

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

That ends the proof of confluence for $\geq$ up to $\equiv$. $\square$

Remark: the specific cases added by $\gtrsim$ are treated the exact same way, and are confluent with $\cong$.

We need to make sure that $\equiv$ does not change the behavior of $\rightarrow$, meaning we would want $\equiv$ to be a bisimulation for $\rightarrow$. That is not the case though, but we can observe an interesting behavior if $\geq$ is allowed to go in there, as reductions can be closed in several steps if $\geq$ is allowed to be used where applicable on the variable the reduction happened in the other branch, for example:

$$\begin{tikzcd}
\lambda_x y.\epsilon_z.P||_x(Q|_xR) \arrow[d] & \equiv & \epsilon_z.\lambda_x y.P||_x(Q|_xR) \arrow[d, dashed, "\geq"] \\
(\epsilon_z.P||_xQ)||_yR[y/x] \arrow[d, dashed, "\geq"] & & \epsilon_z.(\lambda_x y.P||_x(Q|_xR)) \arrow[d] \\
(\epsilon_z.(P||_xQ))||_yR[y/x] & \geq & \epsilon_z.((P||_xQ)||_yR[y/x])
\end{tikzcd}$$

Do we need an other example here? (to pick in the example pool I did on the blackboard maybe)
Also this last paragraph might need a rewrite, the formulation feels a bit sloppy

2 Typing decorated terms with MLL

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

Definition 8

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

Rules for neutral elements:

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

Construction rules:

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

Action rules:

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

Nu rule:

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

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 relations $\to$, $\geq$, $\gtrsim$ 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 9

Cut elimination holds and terminates for $\to$ and $\geq$ under $\equiv$.
For $\to$, we need one more hypothesis: that no quantified type remains in the type at the root of the deduction tree/all quantified types must be closed by the term or context. Should we speak about contexts and this closure condition before this property?

$\triangleright$ Proof by induction on the sizes of the subterms of each rule. Especially, the base cases that eliminate cut are ϵ against 0 and anything against $x \leftrightarrow y$, and all other cases strictly decrease a well chosen tuple of the sizes of subterms. Make a special note that no reduction step will be blocked by an action prefix, as all action prefixes must have a complementary action to be cut against for the context to be closed on quantified variables. Copy and adapt the proof of the temporary paper □

Remark: With no exponential, it is no surprise that cut elimination terminates. We can also relax the closure hypothesis on quantified types, to the condition of proving the holding for $\gtrsim$ and $\rightsquigarrow$ instead, because we need the ability to commute cuts with unmatched action prefixes as well as continue cut elimination under those.

Do we need a proper proposition for that last remark, as well as its proof maybe, or at least some elements for the added rules?

Also, insert here the followed example from the other paper that is to be shown in part 1, to show here that it is not only reducing well, but also well-typed. (Question for later: Is there a term, simplest possible if exists, that reduces well without typing, but cannot be well-typed?)

3 Links with the usual π-calculus

Definition 10

Define π-calculus

Definition 11

Define projection $\lfloor\cdot\rfloor$

Proposition 12

Something about $\to$ 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.