PLACEHOLDER

Julien Gabet

Mars-June, 2018

Introduction

This is an interesting introduction to an interesting subject

1 Definitions

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

Definition

The annotated terms are defined by the following grammar:

$$
\begin{aligned}
P, Q ::= & \; x \leftrightarrow y \;\; ; \;\; 0_x \;\; ; && \textit{base terms} \\
& \epsilon_x.P \;\; ; \;\; \lambda_x y.P \;\; ; && \textit{inactive prefixes} \\
& u_x(t).P \;\; ; \;\; \bar{u}_x\langle v\rangle.P \;\; ; && \textit{action prefixes} \\
& P|_x Q \;\; ; \;\; P||_x Q \;\; ; && \textit{parallel and cut} \\
& (\nu u)P && \textit{name binding prefix}
\end{aligned}
$$

Note the existence of a cut rule, that we would want eliminated. Thus, we define reduction relations and corresponding equivalences to help with that.

Definition

We define a cut-eliminating reduction rule (that does not hold under action prefixes) as follows:

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

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

And we extend it into $\rightsquigarrow$ as follows, allowing for it to act under action prefixes as well:

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

Definition

We also define a congruence rule:

$$
\begin{aligned}
P|_x Q &\equiv Q|_x P && a \\
(P|_x Q)|_y R &\equiv P|_x(Q|_y R) && y \notin fv(P), x \notin fv(R) \quad b \\
P|_x \alpha_y.Q &\equiv \alpha_y.(P|_x Q) && \textit{symmetric in } |, \alpha. \in \{\epsilon., \lambda.t\}, y, t \notin fv(P) \quad c \\
\alpha_x.\beta_y.P &\equiv \beta_y.\alpha_x.P && \alpha., \beta. \in \{\epsilon., \lambda.t\}, x \neq y \quad d \\
(\nu u)P|_x Q &\equiv (\nu u)(P|_x Q) && \textit{symmetric in } |, u \notin fn(Q) \quad e \\
x \leftrightarrow y &\equiv y \leftrightarrow x && e
\end{aligned}
$$

And we also extend it into $\cong$ by allowing α, β to be action prefixes in rules b and c:
$\alpha., \beta. \in \{\epsilon., \lambda.t, u.(t), \bar{u}.\langle v\rangle\}$

The first property we would like to have is the congruence 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

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:
FILL THIS IN WITH REDUCTION TREES

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

2 A typing system that works with cut elimination

Definition

Define the typing system here

Proposition

Cut elimination holds and terminates

▷ proof here □

Proposition

cut elimination works under the restrained $\to$ *and* $\equiv$

▷ proof here □

3 Links with the usual π-calculus

Definition

Define π-calculus

Definition

Define projection $\lfloor \cdot \rfloor$

Proposition

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.