Hard life with weak binders

Massimo Bartoletti, Pierpaolo Degano, Gian Luigi Ferrari
Dipartimento di Informatica, Università di Pisa

Roberto Zunino
Dipartimento di Informatica e Telecomunicazioni, Università degli Studi di Trento
Motivations

• in the $\pi$-calculus, the meaning of the process $\nu n. P$ is twofold:
  • it restricts the scope of the name $n$ to $P$
  • it declares a fresh name $n$ ($\text{gensym}$)
• $\nu$-binders can floated out through the scope extension axioms, e.g.:

\[ P | (\nu n)Q = (\nu n)(P | Q) \quad \text{if } n \notin \text{fn}(P) \]

• the only construct that a $\nu$-binder cannot step over is recursion, i.e. $!(\nu n)P \neq (\nu n)!P$
The idea (1)

- the expansion axioms imply a *normal form* of processes where \( \nu \)-binders are floated out as much as possible.
- is it possible to neglect the “scope restriction” feature of \( \nu \)-binders, and only work with the “*gensym* feature”?
- we replace \( \nu \)-binders with *gensyms* for names
  
  \[(\nu n)P \text{ becomes } new(n) \cdot P\]
- the atomic action \( new(n) \) is our *weak binder*
The idea (2)

Why should weak binders be interesting?

- “lighweight” construct: they permit to reason about freshness in calculi without $\nu$-binders. This can be useful when embedding binders on top of "something" without binders.

- computational advantages: compiling strongly bound processes into weakly bound ones may improve efficiency of implementations $\alpha$-conversion is costly! (cf. explicit substitutions)

- academic speculation: which properties are lost when $\nu$-binders are omitted?
The problems

Discarding \( \nu \)-binders weakens the structure of processes, so giving rise to several issues:

- how to define operational/denotational semantics of weakly bound processes?
- should all weakly bound processes be given a semantics? How to deal with “ambiguous” processes, e.g.:

  \[
  \text{new}(n) \cdot \text{new}(n) \cdot \alpha(n)
  \]

- do weakly bound processes still enjoy some interesting semantic properties?
Our contribution (1)

- we have compared two extensions of BPAs:
  - “strongly bound” PA = BPA + ρ-binders
  - “weakly bound” PA = BPA + weak binders
- we have studied sufficient conditions for a weakly bound process to behave coherently with a strongly bound one. For instance:

\[
\new(n) \cdot \alpha(n) + \new(n') \cdot \alpha(n')
\]

has the same semantics of:

\[
(\forall n)(\forall n') (\new(n) \cdot \alpha(n) + \new(n') \cdot \alpha(n'))
\]
Our contribution (2)

• we have called this sufficient condition “well-boundness” – in symbols \( wb(p) \)

• for the weakly bound processes that are also well-bound, we have defined a transformation “bindification” into strongly bound processes.

\[
\text{bindify}\left(\text{new}(n) \cdot \alpha(n) + \text{new}(n') \cdot \alpha(n')\right)
= (\forall n)(\forall n')(\text{new}(n) \cdot \alpha(n) + \text{new}(n') \cdot \alpha(n'))
\]

• the bindification transformation is sound:

\[
\llbracket \text{bindify}(p) \rrbracket = \llbracket p \rrbracket \quad \text{if } wb(p)
\]
Our contribution (3)

• we have defined and compared equational theories and **trace preorders** for strongly bound and weakly bound processes.

\[ p \preceq q \implies \llbracket p \rrbracket \subseteq \llbracket q \rrbracket \]

• weakly bound processes require particular care to avoid breaking well-boundness.

E.g., naïvely using the unfolding rule we would have:

\[ !\text{new}(n) = \text{new}(n) \cdot \text{new}(n) \cdot !\text{new}(n) \]

but according to our theory the first process is well-bound, while the second is not.
Hard life with weak binders

The idea of weak binders seems nice, so why the “hard life” in the title?

- a lot of side conditions have to be explicitly required in order to make our statements true. (*cf.* the implicit “capture avoiding” condition of $\alpha$-conversion)

- these side conditions have to be preserved by the inductive statements of the proofs. This required a lot of trial-and-error steps before getting right the inductive statements.

- compositionality is reduced, e.g. $wb(p)$ and $wb(q)$ do not imply $wb(p \cdot q)$. 
Legal disclaimer

Weak binders are provided “as is” WITHOUT ANY WARRANTY of any kind, either express or implied.

The authors accepts no liability whatsoever, whether it was caused by:
1. Using weak binders for proving inductive statements about semantic properties of weakly bound processes.
2. Any theorem, lemma, and/or corollary provided/attached to weak binders.

The authors assume that all users understand risks involved within weak binders and/or their attached results.
Overview of the formal development

• strongly bound processes: operational and denotational semantics; full-abstraction: $[P]^{op} = [P]^s$.

• weakly bound processes: denotational semantics $[p]^w$.

• well-boundness $wb(p)$, and bindification $bindify(p)$.

• correctness of bindification: $[bindify(p)]^s = [p]^w$

• equational theory $p \approx q$ and trace preorder $p \preceq q$.

• correctness of the above. Roughly:

\[ p \approx q \implies [p]^w = [q]^w \quad \text{and} \quad wb(q) \iff wb(p) \]

\[ p \preceq q \implies [p]^w \subseteq [q]^w \quad \text{and} \quad wb(q) \implies wb(p) \]
What’s next

For the full technical development, please refer to the paper (and to the technical report for the proofs).

In the rest of the talk, I will just discuss some design choices we made, and show some insightful examples.
Strongly bound processes

$P, Q ::= \varepsilon$ \hspace{1cm} empty process
$\alpha(r)$ \hspace{1cm} event ($r$ is a resource)
$\alpha(n)$ \hspace{1cm} event ($n$ is a name)
$\nu n. P$ \hspace{1cm} $\nu$-binder
$P \cdot Q$ \hspace{1cm} sequential composition
$P + Q$ \hspace{1cm} choice
$h$ \hspace{1cm} variable
$\mu h. P$ \hspace{1cm} recursion

$(\nu n. P, R) \xrightarrow{\text{new}(r)} (P[r/n], R \cup \{r\}) \hspace{1cm} (r \notin R)$
Weakly bound processes

\[ p, q ::= \varepsilon \quad \text{empty process} \]
\[ \alpha(r) \quad \text{event (} r \text{ is a resource)} \]
\[ \alpha(n) \quad \text{event (} n \text{ is a name)} \]
\[ \text{new}(n) \quad \text{gensym} \]
\[ p \cdot q \quad \text{sequential composition} \]
\[ p + q \quad \text{choice} \]
\[ h \quad \text{variable} \]
\[ \mu h.p \quad \text{recursion} \]
Well-boundness examples

\[ x \ p_1 = \text{new}(n) \cdot \text{new}(n) \cdot \alpha(n) \]
\[ x \ p_2 = \alpha(n) \cdot \text{new}(n) \]
\[ x \ p_3 = \text{new}(n) + \alpha(n) \]
\[ x \ p_4 = (\varepsilon + \text{new}(n)) \cdot \alpha(n) \]
\[ \checkmark \ p_5 = \mu h. (\text{new}(n) \cdot h) \]
\[ \checkmark \ p_6 = (\text{new}(n) \cdot \alpha(n) + \text{new}(n) \cdot \alpha'(n)) \cdot \alpha''(n) \]
\[ \checkmark \ p_7 = \text{new}(n) \cdot (\mu h. (\text{new}(n) \cdot h)) \cdot \alpha(n) \]
Bindification examples

- $p_5 = \mu h. new(n) \cdot h$
  $\mu h. \forall n. new(n) \cdot h$

- $p_6 = (new(n) \cdot \alpha(n) + new(n) \cdot \alpha'(n)) \cdot \alpha''(n)$
  $\forall n. ((new(n) \cdot \alpha(n) + new(n) \cdot \alpha'(n)) \cdot \alpha''(n))$

- $p_7 = new(n) \cdot (\mu h. (\varepsilon + new(n) \cdot h)) \cdot \alpha(n)$
  $\forall n. (new(n) \cdot (\mu h. \forall n. (\varepsilon + new(n) \cdot h)) \cdot \alpha(n))$
If \( \text{wb}(p) \), the bindification \( \text{bindify}(p) \) of \( p \) is a strongly bound process, defined as follows:

\[
\text{bindify}(p) = \nu \text{bn}^{\diamond}(p) \cdot \beta(p)
\]

where \( \text{bn}^{\diamond}(p) \) are the names \( n \) that “may” be bound in \( p \) by some \( \text{new}(n) \), and the operator \( \beta \) is defined as follows:

\[
\begin{align*}
\beta(\varepsilon) &= \varepsilon \\
\beta(p \cdot q) &= \beta(p) \cdot \beta(q) \\
\beta(h) &= h \\
\beta(\alpha(r)) &= \alpha(r) \\
\beta(p + q) &= \beta(p) + \beta(q) \\
\beta(\mu h. p) &= \mu h. \text{bindify}(p)
\end{align*}
\]
Trace inclusion preorder

The relation $\preceq_N$ over weakly bound processes is defined as follows ($\mathcal{N}$ is a set of names):

\[
\begin{align*}
p \preceq_N \emptyset q & \quad \text{if } p \approx q \\
p \preceq_N \emptyset p + q & \\
p \preceq_N \mathcal{N} \cup \mathcal{N}' \quad \text{if } p \preceq_N p' \text{ and } p' \preceq_N p'' \\
C(p) \preceq_N C(q) & \quad \text{if } p \preceq_N q \text{ and } \mathcal{N} \not\cap \text{names}(C) \\
p_\sigma\{\mu h. \ p/h\} \preceq_{\text{ran}(\sigma)} \mu h. \ p & \quad \text{if ran}(\sigma) \not\cap \text{fn}(p)
\end{align*}
\]
Trace inclusion preorder examples

Let $p = \mu h. \text{new}(n) \cdot \alpha(n) \cdot h.$

- $\checkmark \ \text{new}(n) \cdot \alpha(n) \cdot p \preceq \{n\} p$
- $\times \ \text{new}(n) \cdot \alpha(n) \cdot \text{new}(n) \cdot \alpha(n) \cdot p \not\preceq p$
- $\checkmark \ \text{new}(n') \cdot \alpha(n') \cdot p \preceq \{n'\} p$
- $\times \ (\text{new}(n') \cdot \alpha(n') \cdot p) \cdot \alpha'(n') \not\preceq p \cdot \alpha'(n')$
Trace inclusion preorder examples

Let \( p = \mu h. \text{new}(n) \cdot \alpha(n) \cdot h. \)

\( \checkmark \) \( \text{new}(n) \cdot \alpha(n) \cdot p \preceq_{\{n\}} p \)

\( \times \) \( \text{new}(n) \cdot \alpha(n) \cdot \text{new}(n) \cdot \alpha(n) \cdot p \not\preceq p \)

\( \checkmark \) \( \text{new}(n') \cdot \alpha(n') \cdot p \preceq_{\{n'\}} p \)

\( \times \) \( (\text{new}(n') \cdot \alpha(n') \cdot p) \cdot \alpha'(n') \not\preceq p \cdot \alpha'(n') \)

\( \checkmark \) \( (\text{new}(n') \cdot \alpha(n') \cdot p) \cdot \alpha'(n) \preceq_{\{n'\}} p \cdot \alpha'(n) \)

\( \times \) \( \alpha(n') \cdot \text{new}(n') \not\preceq \mu h. \alpha(n') \cdot \text{new}(n) \)

\( \checkmark \) \( \alpha(n') \cdot \text{new}(n'') \preceq_{\{n''\}} \mu h. \alpha(n') \cdot \text{new}(n) \)
Open issues

- necessary condition for a weakly bound process to behave like a strongly bound one?
- less restrictive notions of well-boundness?

\[(\text{new}(n) + \text{new}(n')) \cdot (\alpha(n) + \alpha(n'))\]

- can the opposite way (compiling strongly bound processes into weakly bound ones) be interesting, e.g. for gaining efficiency?
- how do the properties of weak binders depend on the choice of the particular process calculus?
Well-bound processes

Well-boundedness $\text{wb}(p)$ is defined as follows:

$$
\text{wb}(\varepsilon) = \text{wb}(\bot) = \text{wb}(\alpha(\rho)) = \text{true}
$$

$$
\text{wb}(\mu h. \ p) = \text{wb}(p)
$$

$$
\text{wb}(p + q) = \text{wb}(p) \land \text{wb}(q) \land
\begin{align*}
\text{bn}^\diamond(p) &\not\cap \text{fn}(q) \\
\text{bn}^\diamond(q) &\not\cap \text{fn}(p)
\end{align*}
$$

$$
\text{wb}(p \cdot q) = \text{bn}^\diamond(q) \not\cap (\text{bn}^\diamond(p) \cup \text{fn}(p)) \land
\begin{align*}
(\text{bn}^\diamond(p) \setminus \text{bn}^\Box(p)) &\not\cap \text{fn}(q)
\end{align*}
$$