Internals of the Mobility Workbench

Björn Victor

Bjorn.Victor@it.uu.se

Uppsala University
Overview

- Equivalence checking
  - Definitions
  - Algorithm
  - Optimizations and details
- De Bruijn name representation
  - Examples
  - Effects on rewrite rules
  - Effects on transition rules
Basics

- Let $P \xrightarrow{\alpha} P'$ be defined using the “commitment” semantics, i.e., using abstractions and concretions, where $\alpha$ is $u$ or $\overline{u}$ for some name $u$.

- Let, likewise, $P \xrightarrow{M,\alpha} P'$ be a symbolic transition with the (positive) condition $M$.

- Let the arity of an agent $P$ be the (positive/negative) length of $\tilde{x}$ of its standard form $(\lambda\tilde{x})P'$ or $(\nu\tilde{y})[\tilde{x}]P'$.

- $P$ is a process if its standard form has arity 0, a concretion if it has arity $<0$, and an abstraction if it arity $>0$.

- $P \bullet Q$, where $P$ is an abstraction $(\lambda\tilde{x})P'$ and $Q$ a concretion $(\nu\tilde{z})[\tilde{y}]Q'$, is $(\nu\tilde{z})(P'[\tilde{y}/\tilde{x}] | Q')$
Open bisimulation (1)

The set $\mathcal{R} = \{S_D\}_D$ where $D$ is a set of distinctions, is an indexed open simulation if for each $S_D \in \mathcal{R}$ and for each $\sigma$ which respects $D$, $(P, Q) \in S_D$ implies

1. whenever $P\sigma \xrightarrow{a} P'$ then there exists a $Q'$ such that $Q\sigma \xrightarrow{a} Q'$ and $(P', Q') \in S_D\sigma$ where $S_D\sigma \in \mathcal{R}$.

2. when $P\sigma$ has standard form $(\nu \tilde{x})[\tilde{y}]P'$ then $Q\sigma$ has standard form $(\nu \tilde{x})[\tilde{y}]Q'$ and $(P', Q') \in S_D'$, where $D' = D\sigma \cup \{\tilde{x} \times \text{fn}(P\sigma, Q\sigma)\}$ and $S_D' \in \mathcal{R}$.

3. when $P\sigma$ has standard form $(\lambda \tilde{x})P'$ then $Q\sigma$ has standard form $(\lambda \tilde{x})Q'$ and $(P', Q') \in S_D\sigma$, where $S_D\sigma \in \mathcal{R}$. 
Open bisimulation (2)

- $\mathcal{R} = \{S_D\}_D$ is an indexed open bisimulation if both $\{S_D\}_D$ and $\{S^{-1}_D\}_D$ are indexed open simulations.

- The agents $P$ and $Q$ are open $D$-bisimilar, written $P \sim_D Q$, if there is an indexed open bisimulation $\mathcal{R} = \{S_D\}_D$ such that $(P, Q) \in S_D$. 
Symbolic open bisimulation

The set $\mathcal{R}$ is an *indexed symbolic open simulation* if for each $S_D \in \mathcal{R}$, $(P, Q) \in S_D$ implies

1. whenever $P \overset{M,a}{\xrightarrow{}} P'$ such that $M$ respects $D$, then there exist $N, b$ and $Q'$ such that $Q \overset{N,b}{\xrightarrow{}} Q'$ and
   - $M \Rightarrow N$
   - $a\sigma_M = b\sigma_M$
   - $(P'\sigma_M, Q'\sigma_M) \in S_{D\sigma_M}$ where $S_{D\sigma_M} \in \mathcal{R}$.

2. and 3. as before

The agents $P$ and $Q$ are *conditional open $D$-bisimilar*, written $P \overset{\approx_D}{\sim} Q$, if there is an indexed conditional open bisimulation $\mathcal{R}$ such that $(P, Q) \in S_D$ for $S_D \in \mathcal{R}$. 
The algorithm (0)

Tries to build a bisimulation relation.

- Represent indexed relation by \((P, Q, D)\) triples.
- To check if \(P \simeq_D Q\), apply algorithm to \((P, Q, D)\) and initial relation \(\emptyset\).
- Note that \(P \simeq_{D'} Q\) implies \(P \simeq_D Q\) if \(D'\) is weaker than \(D\).
The algorithm (1)

Check \((P, Q, D)\) given initial relation \(R\):

1. If \((P, Q, D') \in R\) for some \(D' \subseteq D\), return \(R\), else assume they are equivalent: \(R' = R \cup \{(P, Q, D)\}\)

2. If \(P\) and \(Q\) are processes,
   
   (a) for each \(P \xrightarrow[]{M,a} P'\) respecting \(D\),
       
       i. find a matching transition \(Q \xrightarrow[]{N,b} Q'\)
       
       ii. check \((P'\sigma_M, Q'\sigma_M, D)\) for \(R'\)
       
       iii. if this fails (returns \(\emptyset\)), try next transition of \(Q\)
   
   (b) if no matches were found, return \(\emptyset\);

   (c) else return \(R'\)
The algorithm (2)

3. **Concretions**: If $P \equiv (\nu \tilde{x})[\tilde{y}]P'$ and $Q \equiv (\nu \tilde{z})[\tilde{w}]Q'$, where $|\tilde{x}| = |\tilde{z}|$, and $\tilde{y}\{\tilde{n}/\tilde{x}\} \equiv \tilde{w}\{\tilde{n}/\tilde{z}\}$ where $\tilde{n} = \text{freshnames}(|\tilde{x}|, \text{fn}(P, Q))$,
   - check $(P'\{\tilde{n}/\tilde{y}\}, Q'\{\tilde{n}/\tilde{w}\}, D')$ for $R'$, where $D' = D \cup \{\tilde{n} \times \text{fn}(P, Q)\}$

4. **Abstractions**: If $P \equiv (\lambda \tilde{x})P'$ and $Q \equiv (\lambda \tilde{y})Q'$, where $|\tilde{x}| = |\tilde{y}|$,
   - check $(P'\{\tilde{n}/\tilde{x}\}, Q'\{\tilde{n}/\tilde{y}\}, D)$ for $R'$

5. Else return $\emptyset$. 
Optimizations

- Represent $D$ by a lexicographically sorted list of pairs
- Represent $M$ by a sorted list of equivalence classes (sorted lists)
- Whenever $P \xrightarrow{M,a} P'$ has been derived, record this in a hash table indexed by $P$. So after

\[
\begin{align*}
\text{COM} & \quad u \cdot P \xrightarrow{\emptyset,u} P' \\
& \quad \overline{v} \cdot Q \xrightarrow{\emptyset,v} Q' \\
& \quad P \mid Q \xrightarrow{[u=v],\tau} P' \bullet Q'
\end{align*}
\]

we have recorded all three transitions involved.
De Bruijn indices

Remove the need for $\alpha$-conversion!

- Each name is represented by an index (number) telling how far the occurrence is from its binding.
- Binders ($\lambda$ and $\nu$) no longer take arguments, but only indicate that a binding is taking place.
- All $\alpha$-equivalent agents have the same representation.
- A free occurrence has an index $\geq$ the number of bindings operators preceding it.
- At top level, free names are given indices in some order.
De Bruijn examples

\(\alpha\)-equivalent terms have same representation:

\[
(\lambda x)(\nu y)\bar{x} \cdot [y]0 \Rightarrow \lambda \nu \bar{1}.[0]0
\]

\[
(\lambda a)(\nu b)\bar{a} \cdot [b]0 \Rightarrow \lambda \nu \bar{1}.[0]0
\]

More complex example:

\[
(\lambda a)(\lambda b)a.(\lambda x)b.(\lambda y)[x = y]a.[b]0 \Rightarrow \lambda \lambda 1.\lambda 1.\lambda[1 = 0]3.[2]0
\]

Free names:

\[
(\lambda a)a.(\lambda b)[b = c]\bar{b}.[a][d]0 \Rightarrow \lambda 0.\lambda[0 = 2]\bar{0}.[1][3]0
\]
De Bruijn rewrite rules

$((\nu \tilde{x})P \mid (\nu \tilde{y})Q)$ rewrites to $(\nu \tilde{x} \tilde{y})(P \mid Q)$

by $\alpha$-conversion. Using De Bruijn indices:

$((\nu)^nP \mid (\nu)^mQ)$ rewrites to $(\nu)^{n+m}(P' \mid Q')$

where $P'$ is $P$ with all indices increased by $m$, and $Q'$ is $Q$ with only indices corresponding to free names increased by $n$. 
De Bruin transition rules

\[
\text{S-RES} \quad P \xrightarrow{M, \alpha} P' \quad (\nu b) P \xrightarrow{M, \alpha} (\nu b) P' \quad b \notin \text{n}(\alpha, M)
\]

But no name identity using De Bruijn indices! How do we do?

- Example: \((\nu x)[x]0 \Rightarrow \nu \overline{0}.[0]0\).
- Inner agent: \(\overline{0}.[0]0 \xrightarrow{\emptyset, \overline{0}} [0]0\)
- Reduce indices of label when “going outside” the binding: \(\xrightarrow{\emptyset, \overline{0}} \) becomes \(\xrightarrow{\emptyset, \overline{1}} \) - invalid.
- Compare: \((\nu x)[y][x]0 \Rightarrow \nu \overline{1}.[0]0 \xrightarrow{\emptyset, \overline{0}} (\nu)[0]0\) - legal.