Psi-Calculi Workbench Exercises: Implementing Instances

Ramūnas Gutkovas

August 21, 2013

Formalise and implement the following instances of increasing complexity in Psi-Calculi Workbench (Pwb). These exercises will help you understand the Pwb’s architecture and determine the size of the project; they are not part of examination.

You can get Pwb from the following website

http://www.it.uu.se/research/group/mobility/applied/psiworkbench

Pwb runs on UNIX like systems. On Windows, you can use Cygwin.
To get started, you could try modifying the FHSS instance found in Pwb’s source psi-instances/fhss.ML into the following.

1. Define and implement an instance using the following description. Implement the following data structure

\[
L ::= \text{cons}(L, L) \mid \text{Nil} \mid x \in \mathcal{N}
\]

where \( \mathcal{N} \) is a set of names. This data structure also serves as a channel. Channels are equivalent if they are structurally the same. Use the usual substitution.

In addition to the channel equivalence \( \leftrightarrow \), define the following condition

\[
C ::= L = L
\]

which denotes structural equality, that is, \( \text{cons}(x, y) = \text{cons}(z, w) \) if \( x = z \) and \( y = w \). You could possibly reuse this condition as the channel equivalence.

There is only the unit assertion.

Formalise and implement the transition constraint solver for this instance. Hint: Use unification.
2. Extend and implement the previous instance with the following condition.

\[ C' \ ::= \ L \neq L \]

3. Extend and implement the previous instance with the following data structure:

\[ L' \ ::= \ L \mid \text{car}(L') \mid \text{cdr}(L') \]

and the following conditions

\[ C'' \ ::= \ L' = L' \]

where equality is also modulo the following equations

\[
\begin{align*}
\text{car}(\text{cons}(x, y)) &= x \\
\text{cdr}(\text{cons}(x, y)) &= y
\end{align*}
\]

Hint: generate equations while unifying.

4. Extend and implement the previous instance with the following. Take \( A = P_{\text{fin}}(N) \), assertion composition to be the set union. Add a new condition for each natural number \( n \), time(\( n \)). It is entailed whenever there is \( i \) in the current assertion \( \Psi \) such that \( n \leq i \).

For example, the following process after two \( \tau \) transitions outputs the list \( \text{cons}(t_1, \text{cons}(t_2, \text{Nil})) \) via the channel \( c \).

\[
\text{case} \ \text{time}(2) : c(\text{cons}(t_1, \text{cons}(t_2, \text{Nil}))).0 \mid \tau.(\{\{1\}\} \mid \tau.(\{\{2\}\}))
\]