Proving Liveness Properties

The main liveness property we will be interested in is specified by the response formula

$$p \Rightarrow \Diamond q$$

claiming that every $p$ is eventually followed by a $q$.

Rule **ABS-CHAIN**
For justice requirements $J_1, \ldots, J_m$, and assertions $p, q = h_0, h_1, \ldots, h_m$

C1. \[ p \Rightarrow \bigvee_{j=0}^{m} h_j \]

For $i = 1, \ldots, m$

C2. \[ h_i \land \rho \Rightarrow (h_i' \land \neg J_i) \lor \bigvee_{j<i} h_j' \]

\[ p \Rightarrow \Diamond q \]
**Soundness of Rule** ABS-CHAIN

**Claim 5.** Rule ABS-CHAIN is sound for proving the response property $p \Rightarrow \diamondsuit q$.

**Proof** Assume that the premises of rule ABS-CHAIN are valid. Let $\sigma : s_0, s_1, \ldots$ be a computation of $D$ and let $p$ hold at position $j$. We have to show that there exists a position $k \geq j$ such that $q$ holds at position $k$.

Assume to the contrary, that no position beyond $j$ satisfies $q$. By premise C1, state $s_j$ must satisfy $h_i$, for some $i \geq 0$. Since $q$ never holds beyond $j$, we must have $i > 0$. Let us denote by $i_j$ the index $i > 0$ such that $h_i$ holds at state $s_j$. By premise C2, the successor state $s_{j+1}$ must also satisfy $h_i$, for some $i$, $0 \leq i \leq i_j$. Denote this index by $i_{j+1}$. By argument similar to the above, $i_{j+1} > 0$. In this way we proceed to establish an infinite sequence of indices $i_j \geq i_{j+1} \geq \cdots$ where, for each $k \geq j$, $i_k > 0$ and $s_k \models h_{i_k}$. Since this is an infinite non-increasing sequence, there must exist an index $n$, such that $i_n = i_{n+1} = \cdots$.

By premise C2, we can have $i_k = i_{k+1}$ only if $s_{k+1} \models \neg J_{i_k}$. Thus, justice requirement $J_{i_n}$ is never satisfied beyond position $n$. It follows that $\sigma$ is not a computation, contrary to our original assumption.

We conclude that there must exists a position $k \geq j$ satisfying $q$. \qed
**Rule CHAIN for Programs**

For the case that the considered FDS is derived from a program, it is possible to present a concrete version of rule CHAIN which utilizes the fact that all justice requirements are derived from transitions (statements). Recall that, in such a system, any justice requirement is the negation of an enabling condition of some transition. In the following, we denote the enabling condition of transition $t_i$ by $En(t_i)$.

<table>
<thead>
<tr>
<th>Rule CHAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>For just transitions $t_1, \ldots, t_m$, and assertions $p, q = h_0, h_1, \ldots, h_m$</td>
</tr>
<tr>
<td>C1. $p \Rightarrow \bigvee_{j=0}^{m} h_j$</td>
</tr>
<tr>
<td>For $i = 1, \ldots, m$</td>
</tr>
<tr>
<td>C2. $h_i \land \rho_t \Rightarrow \bigvee_{j \leq i} h'_j$ For every $t \neq t_i$</td>
</tr>
<tr>
<td>C3. $h_i \land \rho_{t_i} \Rightarrow \bigvee_{j &lt; i} h'_j$</td>
</tr>
<tr>
<td>C4. $h_i \Rightarrow En(t_i)$</td>
</tr>
<tr>
<td>$p \Rightarrow \Diamond q$</td>
</tr>
</tbody>
</table>

It can be shown that rule CHAIN is a special case of rule ABS-CHAIN.
Apply to BAKERY-2

\[\text{local } y_1, y_2 : \text{natural initially } y_1 = y_2 = 0\]

\[P_1 :: \begin{cases} \ell_0 : \text{loop forever do} \\
\ell_1 : \text{Non-Critical} \\
\ell_2 : y_1 := y_2 + 1 \\
\ell_3 : y_2 = 0 \lor y_1 < y_2 \\
\ell_4 : \text{Critical} \\
\ell_5 : y_1 := 0 \end{cases} \parallel P_2 :: \begin{cases} m_0 : \text{loop forever do} \\
m_1 : \text{Non-Critical} \\
m_2 : y_2 := y_1 + 1 \\
m_3 : y_1 = 0 \lor y_2 \leq y_1 \\
m_4 : \text{Critical} \\
m_5 : y_2 := 0 \end{cases}\]

The desired response properties for program BAKERY-2 are individual accessibility for the two processes

\[\psi_1 : \text{at} - \ell_2 \Rightarrow \Diamond \text{at} - \ell_4,\]

\[\psi_2 : \text{at} - m_2 \Rightarrow \Diamond \Diamond \text{at} - m_4\]

Let us present a heuristic by which we can systematically derive the auxiliary constructs required by rule CHAIN, in order to prove property \(\psi_1\). Thus, we consider the case that \(p = \text{at} - \ell_2\) and \(q = \text{at} - \ell_4\).
Identifying $t_1$ and $h_1$

Recalling that the condition $f \land \rho_t \to g$ can be rewritten as $f \to \text{pre}(t, g)$, we can summarize premises C2–C4 for the case $i = 1$ into the single implication

$$\text{Imp}(h_0) : \quad h \to \neg h_0 \land \text{En}(t) \land \text{pre}(t, h_0) \land \bigwedge_{\tau \neq t} \text{pre}(\tau, h \lor h_0)$$

where we take $h_0 = q = at_{-\ell_4}$. The conjunct $\neg h_0$ has been added in order to guarantee that all the $h_i$’s will be exclusive.

$\text{Imp}(h_0)$ can be viewed as an inequality with the unknown $h$. For a given $t$, we can try to solve such an inequality by forming the iteration sequence

$$\begin{align*}
\psi_0 &= \neg h_0 \land \text{En}(t) \land \text{pre}(t, h_0), \\
\psi_1 &= \psi_0 \land \bigwedge_{\tau \neq t} \text{pre}(\tau, \psi_0 \lor h_0), \\
\psi_2 &= \psi_1 \land \bigwedge_{\tau \neq t} \text{pre}(\tau, \psi_1 \lor h_0), \\
&\vdots
\end{align*}$$

until it converges.
**Computation of \( h_1 \) Continued**

Let us form the recommended iteration sequence for the case that \( h_0 = at_\ell_4 \) and \( t = \ell_3 \). We obtain

\[
\psi_0 = \left[ \begin{array}{l}
\pi_1 = 3 \land (y_2 = 0 \lor y_1 < y_2) \\
En(\ell_3) \land \frac{\text{pre}(\ell_3, at_\ell_4)}{at_\ell_3} \land (y_2 = 0 \lor y_1 < y_2)
\end{array} \right]
\]

In principle, we should now compute \( \psi_1 = \psi_0 \land \frac{\text{pre}(\tau, \psi_0 \lor h_0)}{\tau \neq \ell_3} \). However, since we can show that every transition different from \( \ell_3 \) preserves \( \psi_0 \), this computation will produce an assertion equivalent to \( \psi_0 \). Thus, the iteration sequence converges in a single step, and produce \( t_1 = \ell_3 \) and

\[
h_1 : at_\ell_3 \land (y_2 = 0 \lor y_1 < y_2)
\]

Why did we choose \( t = \ell_3 \)?

We can try different transitions. However, the computation shows that \( \neg h_0 \land En(t) \land \text{pre}(t, h_0) \) for any \( t \neq \ell_3 \) yields 0 (the empty assertion). Therefore, \( t_1 = \ell_3 \) is the only helpful transition which yields a non-trivial \( h_1 \).
Proceeding to $t_2$ and $h_2$

Once we identified $h_2$, the search for $h_2$ can be based on a solution of the implication $\text{Imp}(h_0 \lor h_1)$ for an appropriately chosen $t = t_2$. Repeating the specified procedure, we end up computing the following sequence of $h_i$ and $t_i$:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$t_i$</th>
<th>$h_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>−</td>
<td>at$_-$l$_4$</td>
</tr>
<tr>
<td>1</td>
<td>l$_3$</td>
<td>at$_-$l$_3$ $\land$ (y$_2$ = 0 $\lor$ y$_1$ &lt; y$_2$)</td>
</tr>
<tr>
<td>2</td>
<td>m$_5$</td>
<td>at$_-$l$<em>3$ $\land$ at$</em>-$m$_5$</td>
</tr>
<tr>
<td>3</td>
<td>m$_4$</td>
<td>at$_-$l$<em>3$ $\land$ at$</em>-$m$_4$</td>
</tr>
<tr>
<td>4</td>
<td>m$_3$</td>
<td>at$_-$l$<em>3$ $\land$ at$</em>-$m$_3$ $\land$ (y$_1$ = 0 $\lor$ y$_2$ ≤ y$_1$)</td>
</tr>
<tr>
<td>5</td>
<td>l$_2$</td>
<td>at$_-$l$_2$</td>
</tr>
</tbody>
</table>

In the computation of this table, we made free use of the relevant invariants which correlate the values of $y_1$ and $y_1$ to the locations of the processes, i.e.

$\Box (y_1 = 0 \leftrightarrow \text{at}_-$l$_0..2)$ and $\Box (y_2 = 0 \leftrightarrow \text{at}_-$m$_0..2)$