Verification Diagrams

Up to now, we have presented the constituents of a proof by rules \textsc{chain}, \textsc{well}, or \textsc{distr-rank} by tables. An alternate presentation is provided by \textit{verification diagrams}. A verification diagram is a directed graph such that:

- Nodes contain labeled assertions, identifying \textit{helpful situations}.
- There exists a single node with no successors, called the \textit{terminal node}, and labeled by the \textit{goal assertion} \( q \).
- Every node has a distinguished edge departing from it, and labeled by a transition which is \textit{helpful} for this node. A node may have additional multiple unhelpful (indifferent) edges departing from it.

Diagrams differ by the rule they represent.
Chain Diagrams

It is required that

- The terminal node is labeled by $h_0 : q$.
- If there is an edge connecting node $h_i$ to node $h_j$, then $i > j$.

Assume that non-terminal node $h_i$ has the helpful transition $t_i$ which connects it to node $h_j$ and the unhelpful successors $h_{k_1}, \ldots, h_{k_n}$. This implies the following verification conditions:

C2. $h_i \land \rho_t \Rightarrow h'_i \lor h'_{k_1} \lor \cdots \lor h'_{k_n}$ For every $t \neq t_i$
C3. $h_i \land \rho_{t_i} \Rightarrow h'_j$
C4. $h_i \Rightarrow E_n(t_i)$

A chain diagram is defined to be $D$-valid if all the verification conditions associated with its nodes are $D$-valid.

Claim 8. If a verification diagram with nodes $h_0, \ldots, h_n$ is $D$-valid then so is the temporal formula

$$\bigvee_{i=0}^n h_i \Rightarrow \lozenge h_0$$

Corollary 9. If, in addition, we establish the $D$-validity of

$$p \Rightarrow \bigvee_{i=0}^n h_i \quad \text{and} \quad h_0 \Rightarrow q$$

then we can conclude

$$p \Rightarrow \lozenge q$$
Example: BAKERY-2

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

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

\[
\begin{array}{c}
h_0 : \text{at}_{\ell_4} \\
h_1 : \text{at}_{\ell_2} \\
h_2 : \text{at}_{\ell_3} \land \text{at}_{m_5} \\
h_3 : \text{at}_{\ell_3} \land \text{at}_{m_4} \\
h_4 : \text{at}_{\ell_3} \land \text{at}_{m_3} \land y_2 \leq y_1
\end{array}
\]
Encapsulation (Statecharts) Conventions

There are several conventions which make visual presentation more effective. We introduce compound nodes which may contains several internal nodes. The following graphical equivalences explain the conventions:

Departing Edges:

Arriving Edges:

Common Factors:
Encapsulated Verification Diagram for BAKERY-2

local \(y_1, y_2\) : natural initially \(y_1 = y_2 = 0\)

\[
P_1 :: \begin{cases} 
  l_0 : \text{loop forever do} \\
  l_1 : \text{Non-Critical} \\
  l_2 : y_1 := y_2 + 1 \\
  l_3 : \text{await} \ y_2 = 0 \lor y_1 < y_2 \\
  l_4 : \text{Critical} \\
  l_5 : y_1 := 0 
\end{cases} \quad \parallel \quad P_2 :: \begin{cases} 
  m_0 : \text{loop forever do} \\
  m_1 : \text{Non-Critical} \\
  m_2 : y_2 := y_1 + 1 \\
  m_3 : \text{await} \ y_1 = 0 \lor y_2 \leq y_1 \\
  m_4 : \text{Critical} \\
  m_5 : y_2 := 0 
\end{cases}
\]

\[
h_5 : \text{at-}l_2 \\
\rightarrow l_2
\]

\[
\text{at-}l_3 \\
h_4 : \text{at-}m_3 \land y_2 \leq y_1 \\
\rightarrow m_3
\]

\[
h_3 : \text{at-}m_4 \\
\rightarrow m_4
\]

\[
h_2 : \text{at-}m_5 \\
\rightarrow m_5
\]

\[
h_1 : y_2 = 0 \lor y_1 < y_2 \\
\rightarrow l_3
\]

\[
h_0 : \text{at-}l_4
\]
WELL Diagrams

Node $h_i$ contains also a ranking function $\delta_i$. It is required that $\delta_0 = 0$.

Assume that non-terminal node $h_i$ has the helpful transition $t_i$ which connects it to node $h_j$ and the unhelpful successors $h_{k_1}, \ldots, h_{k_n}$. This implies the following verification conditions:

W2. $h_i \wedge \rho_t \Rightarrow (h'_i \wedge \delta_i \geq \delta'_i) \vee (h'_{k_1} \wedge \delta_i \geq \delta'_{k_1}) \vee \cdots \vee (h'_{k_n} \wedge \delta_i \geq \delta'_{k_n})$ For every $t \neq t_i$

W3. $h_i \wedge \rho_{t_i} \Rightarrow h'_j \wedge \delta_i \geq \delta'_j$

W4. $h_i \Rightarrow En(t_i)$

A WELL diagram is defined to be $\mathcal{D}$-valid if all the verification conditions associated with its nodes are $\mathcal{D}$-valid.

Claim 10. If a WELL verification diagram with nodes $h_0, \ldots, h_n$ is $\mathcal{D}$-valid then so is the temporal formula

$$\bigvee_{i=0}^{n} h_i \Rightarrow \lozenge h_0$$

Corollary 11. If, in addition, we establish the $\mathcal{D}$-validity of

$$p \Rightarrow \bigvee_{i=0}^{n} h_i \quad \text{and} \quad h_0 \Rightarrow q$$

then we can conclude

$$p \Rightarrow \lozenge q$$
Apply to Program \( \text{UP-DOWN} \)

\( x, y : \text{natural initially} \quad x = y = 0 \)

\[
P_1 :: \begin{cases} 
\ell_0 : \quad \text{while } x = 0 \text{ do} \\
[\ell_1 : \quad y := y + 1] \\
\ell_2 : \quad \text{while } y > 0 \text{ do} \\
[\ell_3 : \quad y := y - 1] \\
\ell_4 
\end{cases} \quad \parallel \quad P_2 :: \begin{cases} 
m_0 : \quad x := 1 \\
m_1
\end{cases}
\]

\( h_5 : at_{\ell_0,1} \land at_{m_0}, \quad \delta : 3 \)

\( at_{m_1} \land x = 1 \)

\( h_4 : at_{\ell_1}, \quad \delta : 2 \)

\( h_3 : at_{\ell_0}, \quad \delta : 1 \)

\( h_2 : at_{\ell_2}, \quad \delta : (0, y, 2) \)

\( h_1 : at_{\ell_3} \land y > 0, \quad \delta : (0, y, 1) \)

\( h_0 : at_{\ell_4}, \quad \delta : 0 \)
Encapsulation Conventions Concerning Ranking

We adopt the additional conventions:

- In case node $h_i$ does not have an explicit ranking labeling, it is as though it had the label $\delta : i$.

- In case a compound node has the transcription $\delta : f$ at its top left corner, the factor $f$ is added as a left lexicographic component to all the rankings of the contained nodes.

\[ h_5 : at_{-\ell_0,1} \land at_{-m_0} \]

\[ at_{-m_1} \land x = 1 \]

\[ h_4 : at_{-\ell_1} \]

\[ h_3 : at_{-\ell_0} \]

\[ \delta : (0, y) \]

\[ h_2 : at_{-\ell_2} \]

\[ h_1 : at_{-\ell_3} \land y > 0 \]

\[ h_0 : at_{-\ell_4} \]

\[ \delta_5 : 5 \]

\[ \delta_4 : 4 \]

\[ \delta_3 : 3 \]

\[ \delta_2 : (0, y, 2) \]

\[ \delta_1 : (0, y, 1) \]

\[ \delta_0 : 0 \]
Diagrams for Parameterized Systems

To deal with parameterized systems, we introduce the inscription $\lambda i : [1..N]$ labeling a compound node. This is equivalent to having $N$ copies of this node, one for each value of $i \in [1..N]$. Assertions and transitions within the node may be parameterized by $i$. 
Example: a Diagram for BAKERY

\[
\begin{align*}
N & : \text{natural where } N > 0 \\
y & : \text{array}[1..N] \text{ of natural where } y = 0 \\
\ell_0 & : \text{loop forever do} \\
\ell_1 & : \text{Non-critical} \\
\ell_2 & : y[i] := \max(y[1], \ldots, y[N]) + 1 \\
\ell_3 & : \text{await } \forall j \neq i : y[j] = 0 \lor y[i] < y[j] \\
\ell_4 & : \text{Critical} \\
\ell_5 & : y[i] := 0
\end{align*}
\]

\[
\begin{align*}
\lambda i : [1..N] : \mu(i) \land \text{at}_-\ell_3[i], & \quad \delta : (3, y[z] - y[i]) \\
\ell_5[i] & \quad \ell_3[i] \\
\ell_5[i] & \quad \ell_4[i] \\
\ell_5[i] & \quad \ell_1[i] \\
\ell_5[i] & \quad \ell_0[i]
\end{align*}
\]
Apply to TOKEN-RING

local $\alpha$: array[1..N] of boolean where $\alpha[1] = 1$, $a[2] = \cdots = a[N] = 0$

$S ::$

\[
l_0 : \text{loop forever do} \\
    l_1 : \text{request } \alpha[i] \\
    l_2 : \text{if } at-m_2[i] \text{ then} \\
    \quad [l_3 : \text{await } at-m_4[i]] \\
    l_4 : \text{release } \alpha[i \oplus 1]
\]

$P[i] ::$

\[
N \parallel \{i = 1, \ldots, N\}
\]

$C ::$

\[
m_0 : \text{loop forever do} \\
    m_1 : \text{Non-critical} \\
    m_2 : \text{await } at-l_3[i] \\
    m_3 : \text{Critical} \\
    m_4 : \text{await } \neg at-l_3[i]
\]
Diagram for **TOKEN-RING**

\[ \lambda i : [1..N] : at\_m_2[i], \quad \delta : \Delta(i, z) \]

\[ h_7 : at\_l_4[i \ominus 1] \]

\[ \ell_4[i \ominus 1] \]

\[ \alpha[i] \]

\[ h_6 : at\_l_0[i] \]

\[ \ell_0[i] \]

\[ h_5 : at\_l_1[i] \]

\[ \ell_1[i] \]

\[ h_4 : at\_l_2[i] \]

\[ \ell_2[i] \]

\[ at\_l_3[i] \]

\[ h_3 : at\_m_2[i] \]

\[ m_2[i] \]

\[ h_2 : at\_m_3[i] \]

\[ m_3[i] \]

\[ h_1 : at\_m_4[i] \]

\[ \ell_3[i] \]

\[ h_0 : at\_m_3[z] \]