Recently, 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 verification diagrams. A verification diagram is a directed graph such that:

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

\begin{itemize}
    \item Every node has a distinguished edge departing from it.
    \item There exists a single node with no successors, called the terminal node, and labeled by the goal assertion \( q \).
    \item Nodes contain labeled assertions, identifying helpful situations.
\end{itemize}

Diagrams differ by the rule they represent.

- \textbf{Deductive Verification of Reactive Systems}, NYU, Fall, 2007
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:

Encapsulated Verification Diagram for BAKERY

Local

\[ y_1 = y_2 = 0 \]

Non-Critical

\[ y_1 = y_2 = 0 \]

Critical

\[ y_1 = 0 \]

\[ y_2 = 0 \]

Deductive Verification of Reactive Systems, NYU, Fall 2007

\[ \text{Critical} \]

\[ \text{Non-Critical} \]

\[ \text{Local} \]

\[ y_1 = y_2 \]

\[ y_1 = y_2 + 1 \]

\[ y_1 = 0 \]

\[ y_2 = 0 \]

\[ y_1 < y_2 \]

\[ y_2 < y_1 \]

\[ y_1 = 0 \]

\[ y_2 = 0 \]

\[ y_1 < y_2 \]

\[ y_2 < y_1 \]
A WELL diagram is defined to be $D$-valid if and only if all the verification conditions associated with its nodes are $D$-valid. The following are the verification conditions:

1. For every $i 
eq 0$, $(y_{i+1} > y_i) \lor (y_{i+1} = y_i)$
2. Assume that non-terminating node $t_I$ has the helpful transition $t_I$, which connects Node $y_i$ contains also a ranking function $y_i$. It is required that $y_0$ = 0.

Apply to Program \text{up-down}

\begin{align*}
P_1 : & \text{while } x = 0 \text{ do}
& \begin{cases}
0 = h = x : 0 = y \\
0 < h = h : y = 1
\end{cases}
\end{align*}

\begin{align*}
P_2 : & \text{while } y > 0 \text{ do}
& \begin{cases}
0 = h = x : 0 = y \\
0 = h = h : y = 0
\end{cases}
\end{align*}

\begin{align*}
0 = h = x : y
\end{align*}
Encapsulation Conventions Concerning Ranking

We adopt the additional conventions:

Diagrams for Parameterized Systems

To deal with parameterized systems, we introduce the inscription $i : [1 \ldots N]$ labeling a compound node. This is equivalent to having $N$ copies of this node, $[N : i] : [1 \ldots N]$. Assertions and transitions within the node may be parameterized by $i$.

In case a compound node has the transcription $g : f$, its top label contains the

In case node $h$ does not have an explicit ranking labeling, it is as though it had

In case node $m$ does not have an explicit ranking labeling, it is as though it had

Diagrams for Parameterized Systems

To deal with parameterized systems, we introduce the inscription $i : [1 \ldots N]$ labeling a compound node. This is equivalent to having $N$ copies of this node, $[N : i] : [1 \ldots N]$. Assertions and transitions within the node may be parameterized by $i$.

In case a compound node has the transcription $g : f$, its top label contains the

In case node $h$ does not have an explicit ranking labeling, it is as though it had

In case node $m$ does not have an explicit ranking labeling, it is as though it had

Diagrams for Parameterized Systems

To deal with parameterized systems, we introduce the inscription $i : [1 \ldots N]$ labeling a compound node. This is equivalent to having $N$ copies of this node, $[N : i] : [1 \ldots N]$. Assertions and transitions within the node may be parameterized by $i$.

In case a compound node has the transcription $g : f$, its top label contains the

In case node $h$ does not have an explicit ranking labeling, it is as though it had

In case node $m$ does not have an explicit ranking labeling, it is as though it had

Diagrams for Parameterized Systems

To deal with parameterized systems, we introduce the inscription $i : [1 \ldots N]$ labeling a compound node. This is equivalent to having $N$ copies of this node, $[N : i] : [1 \ldots N]$. Assertions and transitions within the node may be parameterized by $i$.

In case a compound node has the transcription $g : f$, its top label contains the

In case node $h$ does not have an explicit ranking labeling, it is as though it had

In case node $m$ does not have an explicit ranking labeling, it is as though it had

Diagrams for Parameterized Systems

To deal with parameterized systems, we introduce the inscription $i : [1 \ldots N]$ labeling a compound node. This is equivalent to having $N$ copies of this node, $[N : i] : [1 \ldots N]$. Assertions and transitions within the node may be parameterized by $i$.

In case a compound node has the transcription $g : f$, its top label contains the

In case node $h$ does not have an explicit ranking labeling, it is as though it had

In case node $m$ does not have an explicit ranking labeling, it is as though it had
Example: a Diagram for BAKERY

Apply to TOKEN-RING

Diagram for TOKEN-RING