Nested Waiting-for Formulas

<table>
<thead>
<tr>
<th>$q_m$ interval</th>
<th>$q_{m-1}$ interval</th>
<th>$\cdots$</th>
<th>$q_1$ interval</th>
<th>$q_0$ interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p \varphi_m$</td>
<td>$\varphi_{m-1}$</td>
<td>$\cdots$</td>
<td>$\varphi_1$</td>
<td>$\varphi_0$</td>
</tr>
</tbody>
</table>

**Rule** *nwait* (nested waiting-for)

For assertions $p, q_0, q_1, \ldots, q_m$ and $\varphi_0, \varphi_1, \ldots, \varphi_m$

1. $p \rightarrow \bigvee_{j=0}^{m} \varphi_j$
2. $\varphi_i \rightarrow q_i$ for $i = 0, 1, \ldots, m$
3. $\{\varphi_i\}T \left\{ \bigvee_{j\leq i} \varphi_j \right\}$ for $i = 1, \ldots, m$

$p \Rightarrow q_m \, W \, q_{m-1} \, \cdots \, q_1 \, W \, q_0$

**Example: Program mux-pet1 (Fig. 3.4)**

An example of a nested waiting-for formula is

1-bounded overtaking for MUX-PET1:

\[
\begin{array}{c}
\overline{at_{\ell_3}} \\
\overline{\neg at_{m_{q_3}}} \, W \, at_{m_{q_2}} \, W \, \neg at_{m_{q_1}} \, W \, at_{\ell_4} \\
\end{array}
\]

It states that when process $P_1$ is at $\ell_3$, process $P_2$ can enter its critical section at most once ahead of process $P_1$. 

Premise N3 states that for each assertion $\varphi_i$, each transition $\tau \in T$ either preserves $\varphi_i$ or leads to some $\varphi_j$, with $j < i$. 

Nested Waiting-for Formulas (Cont’d)

$\varphi_i$-interval $\varphi_j$-interval

$\tau$ $\tau$

where $j < i$
Example: Program mux-pet1 (Fig. 3.4)
(Peterson’s Algorithm for mutual exclusion)

```plaintext
local y1, y2: boolean where y1 = F, y2 = F
s : integer where s = 1

m3 : await (¬y1) ∨ (s ≠ 2)
m4 : critical
m5 : y2 := F

P1 ::
\[
\begin{align*}
\ell_0 & : \text{loop forever do} \\
\ell_1 & : \text{noncritical} \\
\ell_2 & : (y_1, s) := (T, 1) \\
\ell_3 & : \text{await } (¬y_2) ∨ (s ≠ 1) \\
\ell_4 & : \text{critical} \\
\ell_5 & : y_1 := F
\end{align*}
\]

P2 ::
\[
\begin{align*}
m_0 & : \text{loop forever do} \\
m_1 & : \text{noncritical} \\
m_2 & : (y_2, s) := (T, 2) \\
m_3 & : \text{await } (¬y_1) ∨ (s ≠ 2) \\
m_4 & : \text{critical} \\
m_5 & : y_2 := F
\end{align*}
\]

With the following strengthenings all premises of rule NWAIT become state-valid.

\[
p: \quad \overline{at_{-\ell 3}}
\]
\[
\varphi_3: \quad at_{-\ell 3} ∧ ¬at_{-m_4} ∧ at_{-m_3} ∧ s = 1 \quad "P_2 \text{ has priority over } P_1"
\]
\[
\varphi_2: \quad at_{-\ell 3} ∧ at_{-m_4}
\]
\[
\varphi_1: \quad at_{-\ell 3} ∧ ¬at_{-m_4} ∧ (at_{-m_3} → s = 2) \quad "P_1 \text{ has priority over } P_2"
\]
\[
\varphi_0 = q_0: \quad at_{-\ell 4}
\]

or equivalently,

\[
p: \quad \overline{at_{-\ell 3}}
\]
\[
\varphi_3: \quad at_{-\ell 3} ∧ at_{-m_3} ∧ s = 1
\]
\[
\varphi_2: \quad at_{-\ell 3} ∧ at_{-m_4}
\]
\[
\varphi_1: \quad at_{-\ell 3} ∧ (at_{-m_0..2,5} ∨ (at_{-m_3} ∧ s = 2))
\]
\[
\varphi_0 = q_0: \quad at_{-\ell 4}
\]

Concatenation of waiting-for formulas

**Rule CONC-W**
\[
p ⇒ q_m W \cdots W q_1 W q_0 \quad \Rightarrow \quad q_0 ⇒ r_n W \cdots W r_0
\]
\[
p ⇒ q_m W \cdots W q_1 W r_n W \cdots W r_0
\]

Collapsing of waiting-for formulas

**Rule COLL-W**

For \( i > 0 \)
\[
p ⇒ q_m W \cdots W q_{i+1} W q_i W \cdots W q_0
\]
\[
p ⇒ q_m W \cdots W (q_{i+1} ∨ q_i) W \cdots W q_0
\]

\[
q_m \cdots q_{i+1} q_i \cdots q_1
\]

\[
q_m \cdots q_{i+1} ∨ q_i \cdots q_1
\]
Basic Verification Diagrams
A visual summary of verification proofs

Verification Diagrams (VDs) allow a graphical representation of a proof of a temporal property.

To prove $\varphi$ is $P$-valid, find diagram $\Psi$ such that:

$$\mathcal{L}(P) \subseteq \mathcal{L}(\Psi) \subseteq \mathcal{L}(\varphi)$$

i.e., every $P$-computation $\sigma$ is a $\Psi$-sequence and every $\Psi$-sequence $\sigma$ is a model of $\varphi$ (satisfies $\sigma \models \varphi$).

Verification Diagram (VD)
Directed labeled graph with

- **Nodes** – labeled by assertions
  
  \[ \varphi \]

- **Edges** – labeled by names of transitions
  
  \[ \varphi_1 \xrightarrow{\tau_{e_0}} \varphi_2 \]

- **Terminal Node** (“goal”) – no edges depart from it
  
  \[ \varphi_0 \]

Verification conditions (VCs)
VD provides a concise representation of sets of VCs:

- The verification condition associated with a node labeled by $\varphi$ and a transition $\tau$ is
  
  $$\Rightarrow \{ \varphi \} \tau \{ \varphi \lor \varphi_1 \lor \ldots \lor \varphi_k \}$$

  There is an implicit $\tau$-edge connecting each $\varphi$-node to itself.

- Nonterminal node without outgoing edges
  
  $$\Rightarrow \{ \varphi \} \tau \{ \varphi \}$$

  **Note:** No verification conditions for terminal node.

**Definition:** VD is $P$-valid iff all VCs associated with nodes in the diagram are $P$-state valid.
Compound Nodes: Statecharts Conventions

- Departing edges

- Arriving edges

Classes of Diagrams

- Proofs of invariance properties

\[ \Box q \]

are represented by **INVARIA\textsc{NCE}** diagrams

- Proofs of precedence properties

\[ p \Rightarrow q_m \lor q_{m-1} \lor \cdots \lor q_1 \lor q_0 \]

are represented by **WAIT** diagrams

- Proofs of response properties

\[ p \Rightarrow \diamond q \]

are represented by **CHAIN** and **RANK** diagrams (Vol. III)

Wait Diagrams

VDs with nodes \( \varphi_m, \ldots, \varphi_0 \) such that:

- weakly acyclic, i.e.,

  \[
  \text{if } \varphi_i \rightarrow \varphi_j \text{ then } i \geq j
  \]

- \( \varphi_0 \) is a terminal node
Claim (wait diagram):

A $P$-valid wait diagram establishes that

$$\bigvee_{j=0}^{m} \varphi_j \Rightarrow \varphi_m W \varphi_{m-1} \cdots \varphi_1 W \varphi_0$$

is $P$-valid.

If, in addition,

(N1) $p \rightarrow \bigvee_{j=0}^{m} \varphi_j$

(N2) $\varphi_i \rightarrow q_i$ for $i = 0, 1, \ldots, m$

are $P$-state valid, then

$$p \Rightarrow q_m W q_{m-1} \cdots q_1 W q_0$$

is $P$-valid.

10-17

Example: Program mux-pet1 (Fig. 3.4)

(Peterson’s Algorithm for mutual exclusion)

local $y_1, y_2$: boolean where $y_1 = F, y_2 = F$

$s$: integer where $s = 1$

$\ell_0$: loop forever do

$P_1::$

$$\begin{cases}
\ell_1: & \text{noncritical} \\
\ell_2: & (y_1, s) := (T, 1) \\
\ell_3: & \text{await } (\neg y_2) \lor (s \neq 1) \\
\ell_4: & \text{critical} \\
\ell_5: & y_1 := F
\end{cases}$$

$\ell_0$

$\ell_0$

$\ell_0$

$\ell_0$

$\ell_0$

$\ell_0$

$\ell_0$

$\ell_0$

$\ell_0$

$\ell_0$

$\ell_0$

10-19

Example: Program MUX-PET1 (Con’t)

WAIT diagram (Fig. 3.8)

(1-bounded overtaking from $\ell_3$)

$\psi: \begin{array}{l}
at_{\ell_3} \\
\quad \vdash \\
\psi_3: at_{m_3} \land s = 1 \\
\quad \vdash \\
\psi_2: at_{m_4} \\
\quad \vdash \\
\psi_1: at_{m_0, 2,5} \lor (at_{m_3} \land s = 2) \\
\quad \vdash \\
\ell_3 \\
\quad \vdash \\
\varphi_0: at_{\ell_4}
\end{array}$

10-20
Example: Program MUX-PET1 (Con’t)

Associated VCs

- From $\varphi_3$
  $\{\varphi_3\} m_3 \{\varphi_3 \lor \varphi_2\}$
  $\cdots \land \alpha \lor m_4 \alpha \to \varphi_3 \lor m_4 \beta$

  $\{\varphi_3\} m_3 \{\varphi_3\}$

  for all non-$m_3$ transitions.

  But since we are at $-\ell_3$, at $-m_3$, check only $\ell_3$.

Therefore, $\psi$: $\alpha \Rightarrow \varphi_3 W \varphi_2 W \varphi_1 W \varphi_0$ is valid over MUX-PET1.

In addition,

$$\begin{align*}
\text{at}_-\ell_3 & \vdash \bigvee_{j=0}^{3} \varphi_j \\
\varphi_0 & \Rightarrow \text{at}_-\ell_4
\end{align*}$$

are $P$-state valid.

Therefore,

$$\psi: \text{at}_-\ell_3 \Rightarrow (\neg \text{at}_-m_4) W \text{at}_-m_4 W (\neg \text{at}_-m_4) W \text{at}_-\ell_4$$

is valid over MUX-PET1.

Invariance Diagrams

VDs with no terminal nodes (cycles OK)

Claim (invariance diagram):

A $P$-valid INVARianCE diagram establishes that

$$\bigvee_{j=1}^{m} \varphi_j \Rightarrow \Box(\bigvee_{j=1}^{m} \varphi_j)$$

is $P$-valid.

If, in addition,

$$\begin{align*}
\Theta & \Rightarrow \bigvee_{j=1}^{m} \varphi_j \\
\bigvee_{j=1}^{m} \varphi_j & \Rightarrow q
\end{align*}$$

are $P$-state valid, then

$$\Box q$$

is $P$-valid.
Example: Program MUX-PET1 (Fig 3.4)

Establish \( \square (y_1 ↔ at_{-\ell 3.5}) \)

INVARINACE diagram valid for program MUX-PET1

\( \varphi_1: at_{-\ell 0.2} ∧ \neg y_1 \)
\( \varphi_2: at_{-\ell 3.5} ∧ y_1 \)

because
\[
\{ \varphi_1 \} \ell_2 \{ \varphi_1 ∨ \varphi_2 \} \quad \{ \varphi_1 \} \ell_5 \{ \varphi_1 \} \\
\{ \varphi_2 \} \ell_5 \{ \varphi_2 ∨ \varphi_1 \} \quad \{ \varphi_2 \} \ell_5 \{ \varphi_2 \}
\]

Thus
\[ \varphi_1 ∨ \varphi_2 \Rightarrow \square (\varphi_1 ∨ \varphi_2) \]

Also,
\[
(11) \quad at_{-\ell 0} ∧ \neg y_1 ∧ \cdots \Rightarrow \\
\quad \begin{array}{c}
\quad at_{-\ell 0.2} ∧ \neg y_1 ∨ \\
\quad \varphi_1 \quad \varphi_2
\end{array}
\]
\[
(12) \quad at_{-\ell 0.2} ∧ \neg y_1 ∨ at_{-\ell 3.5} ∧ y_1 \Rightarrow \\
\quad y_1 ↔ at_{-\ell 3.5}
\]

are state-valid

Therefore
\[ \square (y_1 ↔ at_{-\ell 3.5}) \]

is P-valid.

Example: Program MUX-PET1 (Fig. 3.4)

Establish \( \square \neg (at_{-\ell 4} ∧ at_{-m4}) \)

\( \varphi_0: nc_1 ∧ nc_2 \)
\( \varphi_1: nc_1 ∧ c_2 \)
\( \varphi_2: c_1 ∧ nc_2 \)
\( \varphi_3: pc_1 ∧ c_2 \)
\( \varphi_4: c_1 ∧ pc_2 \)

non-critical: \( nc_1: at_{-\ell 0.2} \)
\( nc_2: at_{-m0.2} \)

critical: \( c_1: at_{-\ell 3.5} ∧ \neg y_2 \)
\( c_2: at_{-m3.5} ∧ \neg y_1 \)

pre-critical: \( pc_1: at_{-\ell 3} ∧ s = 1 ∧ y_2 \)
\( pc_2: at_{-m3} ∧ s = 2 ∧ y_1 \)