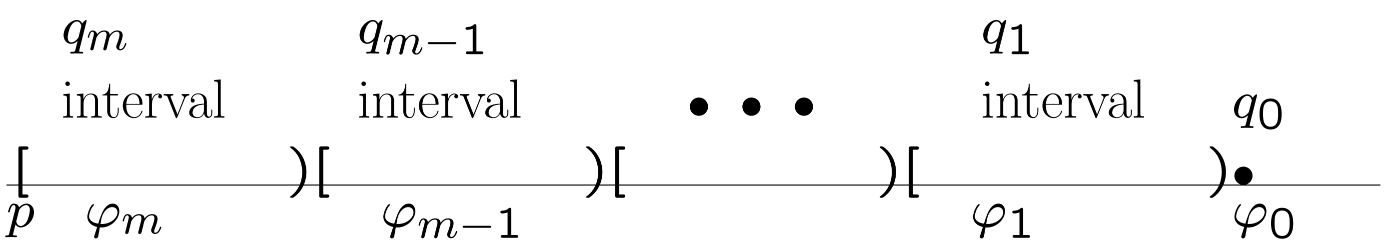
CS256/Winter 2009 Lecture #10

Zohar Manna

Nested Waiting-for Formulas



Rule nwait (nested waiting-for)

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

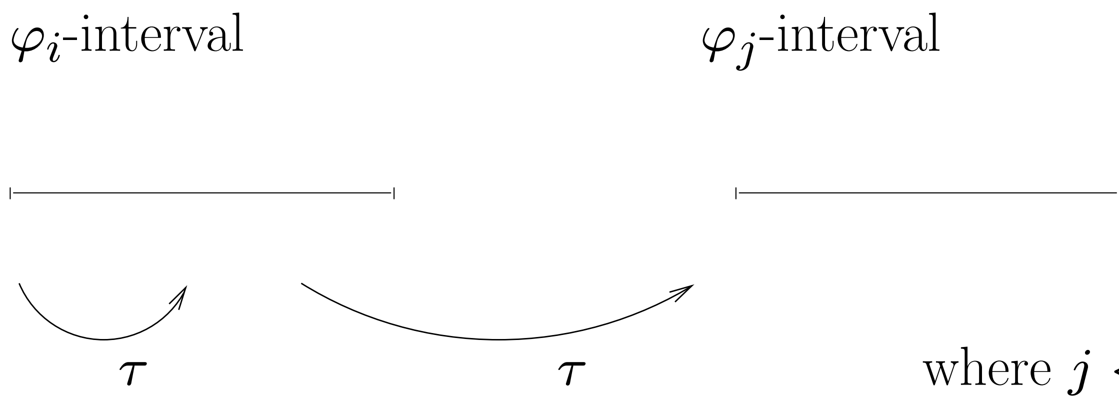
N1. $p \rightarrow \bigvee_{j=0}^{m} \varphi_j$

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

N3. $\{\varphi_i\}\mathcal{T}\left\{\bigvee_{j \leq i} \varphi_j\right\}$ for $i = 1, \ldots, m$

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

Nested Waiting-for Formulas (Cont'd)



where $j < i$

Premise N3 states that for each assertion φ_i, each transition $\tau \in \mathcal{T}$ either preserves φ_i or
leads to some φ_j, with $j < i$.

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

An example of a nested waiting-for formula is
1-bounded overtaking for MUX-PET1:

$$\underbrace{at\_\ell_3}_{p} \;\Rightarrow\; \underbrace{\neg at\_m_4}_{q_3} \;\mathcal{W}\; \underbrace{at\_m_4}_{q_2} \;\mathcal{W}\; \underbrace{\neg at\_m_4}_{q_1} \;\mathcal{W}\; \underbrace{at\_\ell_4}_{q_0}$$

It states that when process P_1 is at ℓ_3,
process P_2 can enter its critical section at most
once ahead of process P_1.

Example: Program mux-pet1 (Fig. 3.4)

(Peterson's Algorithm for mutual exclusion)

$$
\begin{array}{llll}
\textbf{local} & y_1, y_2 : & \textbf{boolean} & \textbf{where } y_1 = \text{F}, y_2 = \text{F} \\
 & s \quad : & \textbf{integer} & \textbf{where } s = 1
\end{array}
$$

$$
P_1 :: \quad \ell_0 : \ \textbf{loop forever do} \left[\begin{array}{ll}
\ell_1 : & \textbf{noncritical} \\
\ell_2 : & (y_1,\ s) := (\text{T},\ 1) \\
\ell_3 : & \textbf{await } (\neg y_2) \vee (s \neq 1) \\
\ell_4 : & \textbf{critical} \\
\ell_5 : & y_1 := \text{F}
\end{array}\right]
$$

$$||$$

$$
P_2 :: \quad m_0 : \ \textbf{loop forever do} \left[\begin{array}{ll}
m_1 : & \textbf{noncritical} \\
m_2 : & (y_2,\ s) := (\text{T},\ 2) \\
m_3 : & \textbf{await } (\neg y_1) \vee (s \neq 2) \\
m_4 : & \textbf{critical} \\
m_5 : & y_2 := \text{F}
\end{array}\right]
$$

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

p: $\underline{at\_\ell_3}$

φ_3: $at\_\ell_3 \wedge \underline{\neg at\_m_4} \wedge at\_m_3 \wedge s = 1$
"P_2 has priority over P_1"

φ_2: $at\_\ell_3 \wedge \underline{at\_m_4}$

φ_1: $at\_\ell_3 \wedge \underline{\neg at\_m_4} \wedge (at\_m_3 \rightarrow s = 2)$
"P_1 has priority over P_2"

$\varphi_0 = q_0$: $\underline{at\_\ell_4}$

or equivalently,

p: at_ℓ_3

φ_3: $at\_\ell_3 \wedge at\_m_3 \wedge s = 1$

φ_2: $at\_\ell_3 \wedge at\_m_4$

φ_1: $at\_\ell_3 \wedge (at\_m_{0..2,5} \vee (at\_m_3 \wedge s = 2))$

$\varphi_0 = q_0$: at_ℓ_4

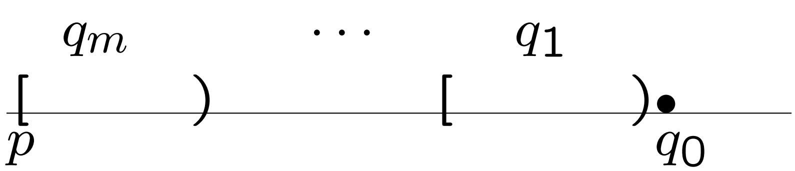
Concatenation of waiting-for formulas

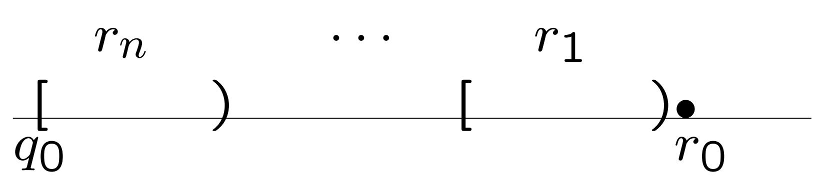
Rule CONC-W

$$p \Rightarrow q_m \, \mathcal{W} \cdots q_1 \, \mathcal{W} \, q_0$$

$$q_0 \Rightarrow r_n \, \mathcal{W} \cdots \mathcal{W} \, r_0$$

$$\overline{\qquad p \Rightarrow q_m \, \mathcal{W} \cdots \mathcal{W} \, q_1 \, \mathcal{W} \, r_n \, \mathcal{W} \cdots \mathcal{W} \, r_0 \qquad}$$



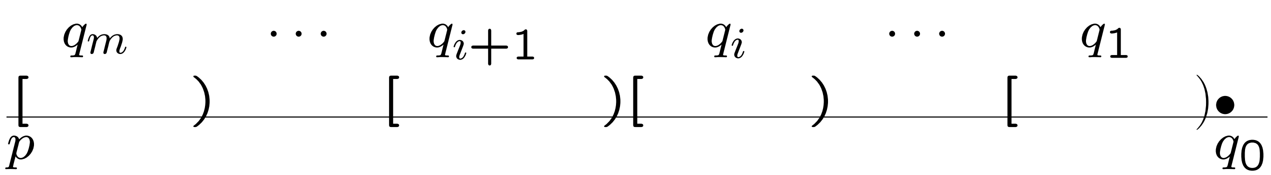


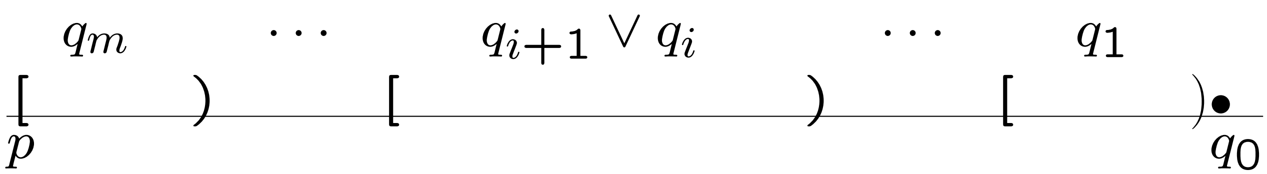
Collapsing of waiting-for formulas

Rule COLL-W

For $i > 0$

$$\frac{p \;\Rightarrow\; q_m \,\mathcal{W}\, \cdots \,\mathcal{W}\, q_{i+1} \,\mathcal{W}\, q_i \,\mathcal{W}\, \cdots \,\mathcal{W}\, q_0}{p \;\Rightarrow\; q_m \,\mathcal{W}\, \cdots \,\mathcal{W}\, (q_{i+1} \vee q_i) \,\mathcal{W}\, \cdots \,\mathcal{W}\, q_0}$$





Basic Verification Diagrams

A visual summary of verification proofs

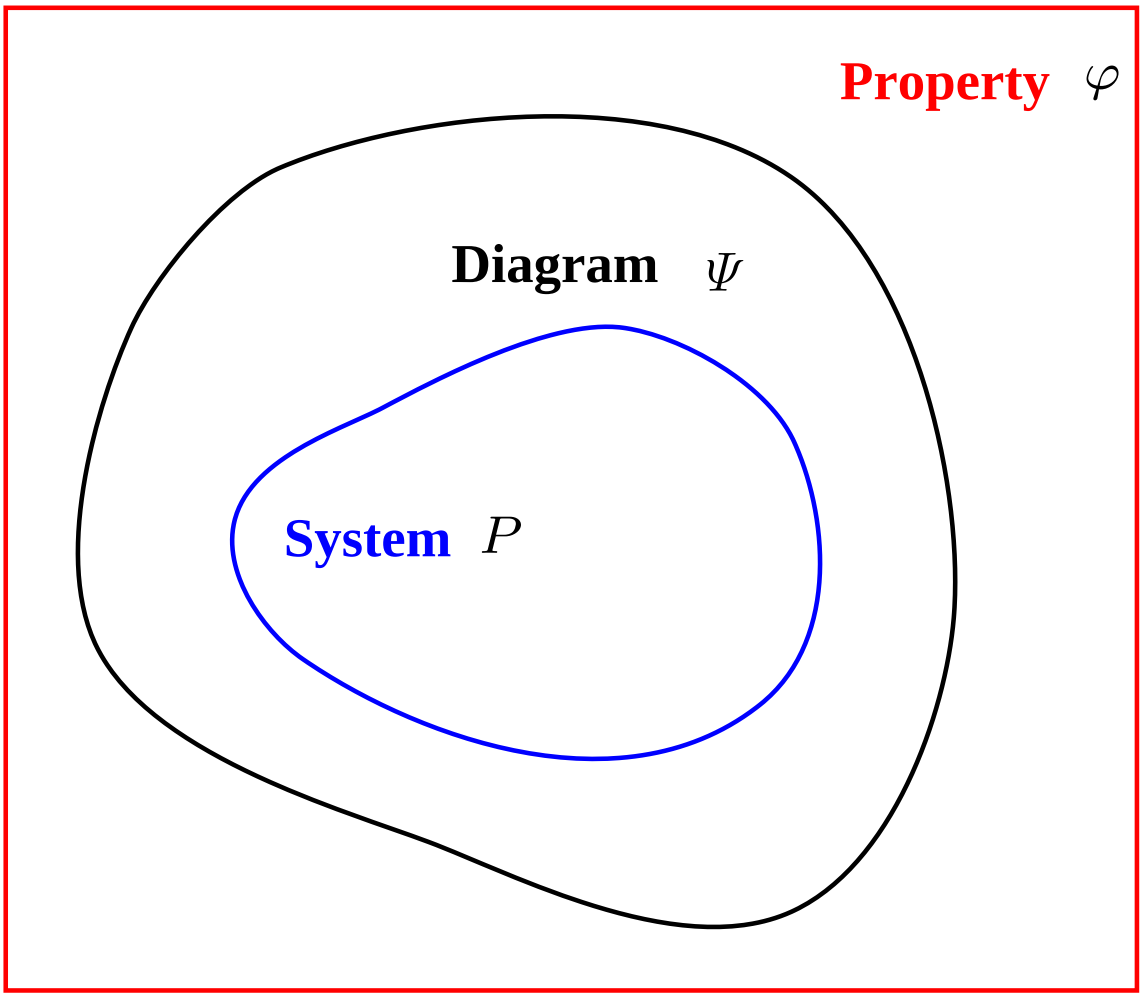
Verification Diagrams (VDs) allow a graphical representation of a proof of a temporal property.

To prove φ is P-valid, find diagram Ψ such that:

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

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

Verification Diagrams (VDs)



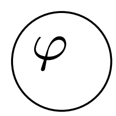
$\mathcal{L}(P) \subseteq \mathcal{L}(\Psi)$ proved by verification conditions.

$\mathcal{L}(\Psi) \subseteq \mathcal{L}(\varphi)$ follows from well-formedness of diagram.

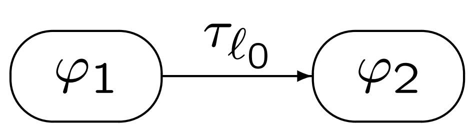
Verification Diagram (VD)

Directed labeled graph with

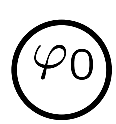
- Nodes – labeled by assertions



- Edges – labeled by names of transitions



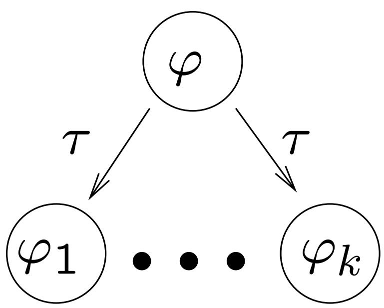
- Terminal Node ("goal") – no edges depart from it



Verification conditions (VCs)

VD provides a concise representation of sets of VCs:

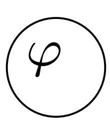
- The verification condition associated with a node labeled by φ and a transition τ is



$$\Rightarrow \{\varphi\}\ \tau\ \{\varphi \vee \varphi_1 \vee \ldots \vee \varphi_k\}$$

There is an implicit τ-edge connecting each φ-node to itself.

- Nonterminal node without outgoing edges



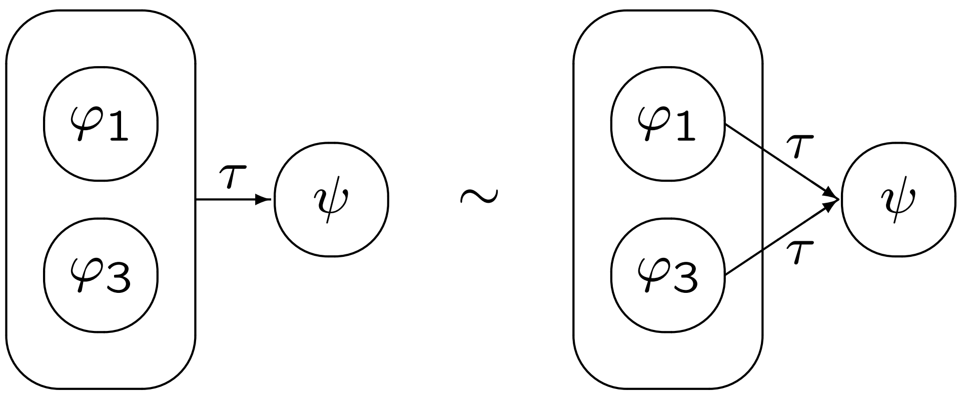
$$\Rightarrow \{\varphi\}\ \tau\ \{\varphi\}$$

Note: No verification conditions for terminal node.

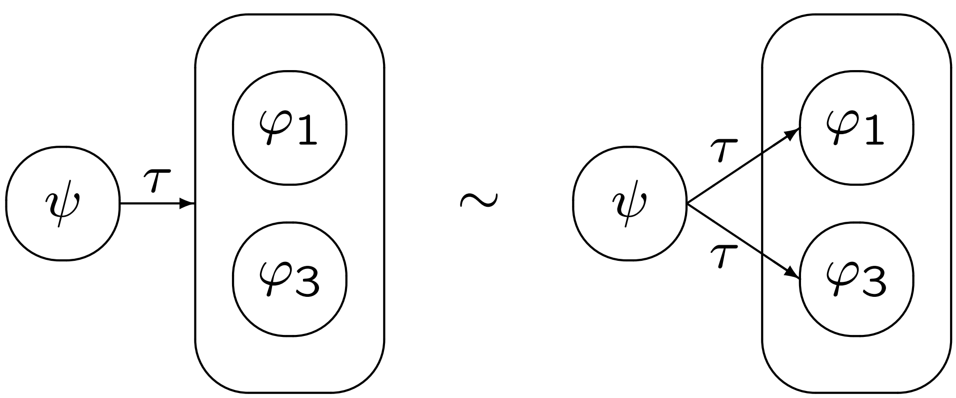
Definition: VD is *P*-valid iff all VCs associated with nodes in the diagram are *P*-state valid

Compound Nodes: Statecharts Conventions

- Departing edges

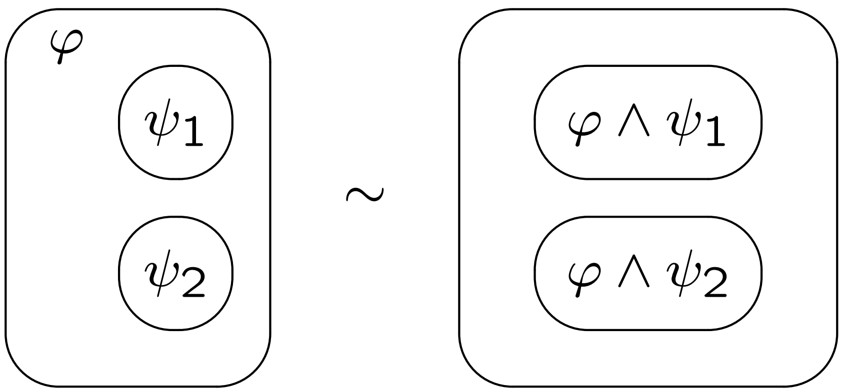


- Arriving edges



Compound Nodes: Statecharts Conventions

- Common factors



Classes of Diagrams

- Proofs of invariance properties

$$\Box\, q$$

are represented by INVARIANCE diagrams

- Proofs of precedence properties

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

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.,

  if $\varphi_i \longrightarrow \varphi_j$

  then $i \geq j$

- φ_0 is a terminal node

<u>`Claim` (wait diagram):</u>

A P-valid WAIT diagram establishes that

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

is P-valid.

If, <u>in addition</u>,

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

(N2) $\quad \varphi_i \;\rightarrow\; q_i \quad \text{for} \quad i = 0, 1, \ldots, m$

are P-state valid, then

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

is P-valid.

`Example:` Program MUX-PET1 (Fig 3.4)

$\mathtt{1}$-bounded overtaking from ℓ_3

$$\psi\colon\ \underbrace{at\_\ell_3}_{p} \Rightarrow$$

$$(\underbrace{\neg at\_m_4}_{q_3})\ \mathcal{W}\ \underbrace{at\_m_4}_{q_2}\ \mathcal{W}\ (\underbrace{\neg at\_m_4}_{q_1})\ \mathcal{W}\ \underbrace{at\_\ell_4}_{q_0}$$

Proof is summarized in WAIT diagram

(Fig 3.8)

Example: Program mux-pet1 (Fig. 3.4)

(Peterson's Algorithm for mutual exclusion)

$$
\begin{array}{llll}
\textbf{local} & y_1, y_2 : & \textbf{boolean} & \textbf{where } y_1 = \text{F}, y_2 = \text{F} \\
 & s \quad : & \textbf{integer} & \textbf{where } s = 1
\end{array}
$$

$$
P_1 :: \quad \begin{array}{l}
\ell_0 : \ \textbf{loop forever do} \\
\left[\begin{array}{ll}
\ell_1 : & \textbf{noncritical} \\
\ell_2 : & (y_1,\ s) := (\text{T},\ 1) \\
\ell_3 : & \textbf{await } (\neg y_2) \vee (s \neq 1) \\
\ell_4 : & \textbf{critical} \\
\ell_5 : & y_1 := \text{F}
\end{array}\right]
\end{array}
$$

$$||$$

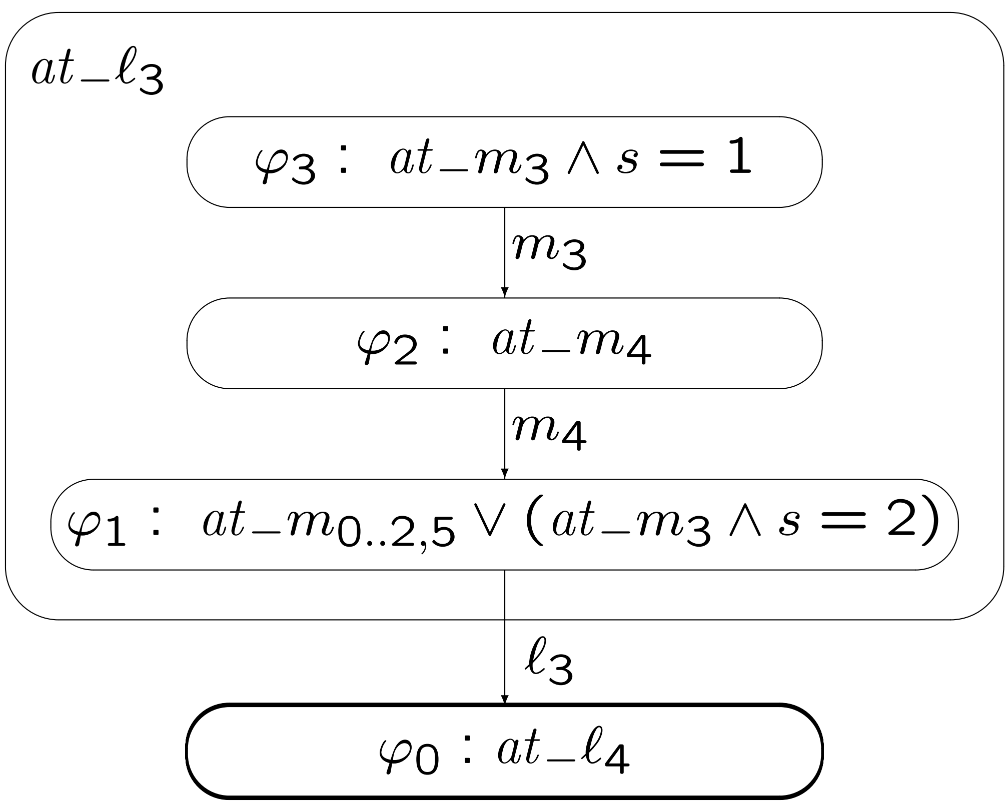
$$
P_2 :: \quad \begin{array}{l}
m_0 : \ \textbf{loop forever do} \\
\left[\begin{array}{ll}
m_1 : & \textbf{noncritical} \\
m_2 : & (y_2,\ s) := (\text{T},\ 2) \\
m_3 : & \textbf{await } (\neg y_1) \vee (s \neq 2) \\
m_4 : & \textbf{critical} \\
m_5 : & y_2 := \text{F}
\end{array}\right]
\end{array}
$$

Example: Program MUX-PET1 (Con't)

WAIT diagram (Fig. 3.8)
(**1**-bounded overtaking from ℓ_3)

ψ: $\underbrace{at\_\ell_3}_{p} \Rightarrow$

$$\underbrace{(\neg at\_m_4)}_{q_3} \ \mathcal{W} \ \underbrace{at\_m_4}_{q_2} \ \mathcal{W} \ \underbrace{(\neg at\_m_4)}_{q_1} \ \mathcal{W} \ \underbrace{at\_\ell_4}_{q_0}$$



<u>`Example:`</u> Program MUX-PET1 (Con't)

<u>Associated VCs</u>

- From φ_3

  $\{\varphi_3\}\ m_3\ \{\varphi_3 \vee \varphi_2\}$

  $$\underbrace{\ldots}_{\varphi_3} \wedge \underbrace{\ldots \wedge at'\_m_4}_{\rho_{m_3}} \rightarrow \underbrace{\ldots}_{\varphi_3'} \vee \underbrace{at'\_m_4}_{\varphi_2'}$$

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

  for all non-m_3 transitions.
  But since we are at_ℓ_3, at_m_3, check only ℓ_3.

$\{\varphi_3\}\ \ell_3\ \{\varphi_3\}$ holds, since

$$\underbrace{at\_m_3 \wedge \ldots \wedge s = 1}_{\varphi_3} \wedge \underbrace{\ldots \wedge ((\neg y_2) \vee (s \neq 1))}_{\rho_{\ell_3}} \rightarrow \underbrace{\ldots}_{\varphi_3'}$$

Recall that by χ_2, $at\_m_3 \rightarrow y_2$.

- From φ_2

  $\{\varphi_2\}\ m_4\ \{\varphi_2 \vee \varphi_1\}$

  $\{\varphi_2\}\ \overline{m_4}\ \{\varphi_2\}$

- From φ_1

  $\{\varphi_1\}\ \ell_3\ \{\varphi_1 \vee \varphi_0\}$

  $\{\varphi_1\}\ \overline{\ell_3}\ \{\varphi_1\}$

They are P-state valid
[not state-valid - require invariants $\chi_0, \ldots, \chi_4$]

Therefore,
WAIT diagram is valid over MUX-PET1

Example: Program MUX-PET1 (Con't)

Therefore,

$$\bigvee_{i=0}^{3} \varphi_i \Rightarrow \varphi_3 \ \mathcal{W} \ \varphi_2 \ \mathcal{W} \ \varphi_1 \ \mathcal{W} \ \varphi_0$$

is valid over MUX-PET1.

In addition,

$$\underbrace{at\_\ell_3}_{p} \rightarrow \bigvee_{j=0}^{3} \varphi_j$$

$$\varphi_0 \rightarrow \underbrace{at\_\ell_4}_{q_0} \qquad \varphi_1 \rightarrow \underbrace{\neg at\_m_4}_{q_1}$$

$$\varphi_2 \rightarrow \underbrace{at\_m_4}_{q_2} \qquad \varphi_3 \rightarrow \underbrace{\neg at\_m_4}_{q_3}$$

are P-state valid.

Therefore,

$$\psi: \ at\_\ell_3 \ \Rightarrow$$
$$(\neg at\_m_4) \ \mathcal{W} \ at\_m_4 \ \mathcal{W} \ (\neg at\_m_4) \ \mathcal{W} \ at\_\ell_4$$

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,

$$\text{(I1)} \quad \Theta \rightarrow \bigvee_{j=1}^{m} \varphi_j$$

$$\text{(I2)} \quad \bigvee_{j=1}^{m} \varphi_j \rightarrow q$$

are P-state valid, then

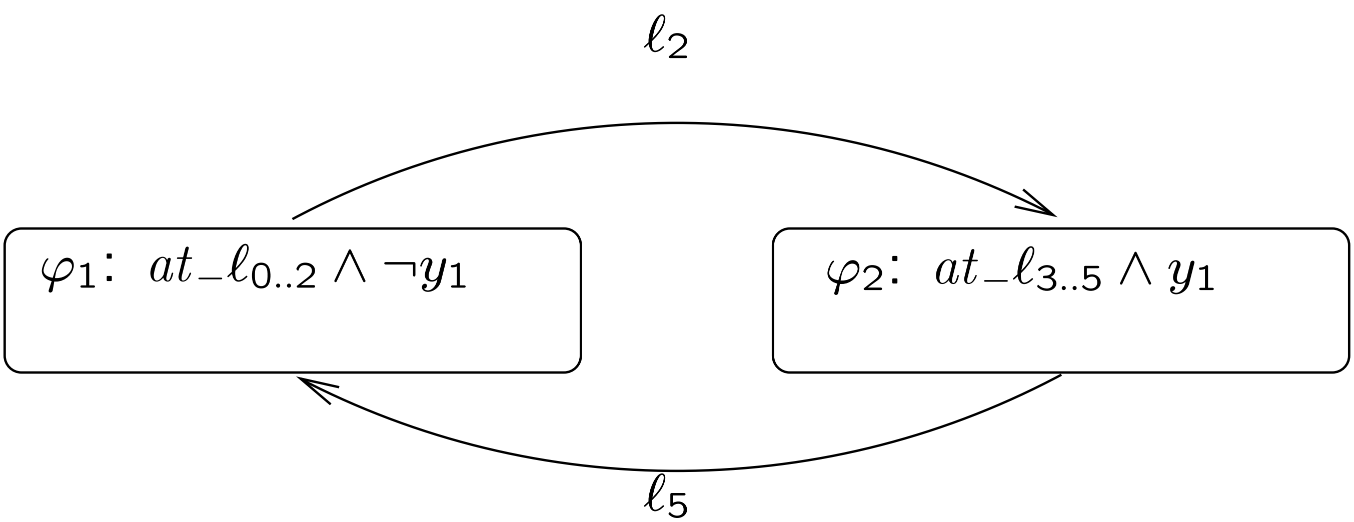
$$\boxed{\Box\, q}$$

is P-valid

`Example:` Program MUX-PET1 (Fig 3.4)

Establish $\boxed{\Box \underbrace{(y_1 \leftrightarrow at\_\ell_{3..5})}_{q}}$

INVARIANCE diagram
valid for program MUX-PET1



because

$$\{\varphi_1\}\, \ell_2\, \{\varphi_1 \lor \varphi_2\} \qquad \{\varphi_1\}\, \overline{\ell_2}\, \{\varphi_1\}$$

$$\{\varphi_2\}\, \ell_5\, \{\varphi_2 \lor \varphi_1\} \qquad \{\varphi_2\}\, \overline{\ell_5}\, \{\varphi_2\}$$

Thus

$$\varphi_1 \lor \varphi_2 \Rightarrow \Box(\varphi_1 \lor \varphi_2)$$

Also,

$$\text{(I1)}\ \underbrace{at\_\ell_0 \ \wedge\ \neg y_1 \ \wedge\ \cdots}_{\Theta} \ \rightarrow$$

$$\underbrace{at\_\ell_{0..2} \ \wedge\ \neg y_1}_{\varphi_1} \ \vee\ \underbrace{\cdots}_{\varphi_2}$$

$$\text{(I2)}\ \underbrace{at\_\ell_{0..2} \ \wedge\ \neg y_1}_{\varphi_1} \ \vee\ \underbrace{at\_\ell_{3..5} \ \wedge\ y_1}_{\varphi_2} \ \rightarrow$$

$$\underbrace{y_1 \ \leftrightarrow\ at\_\ell_{3..5}}_{q}$$

are state-valid

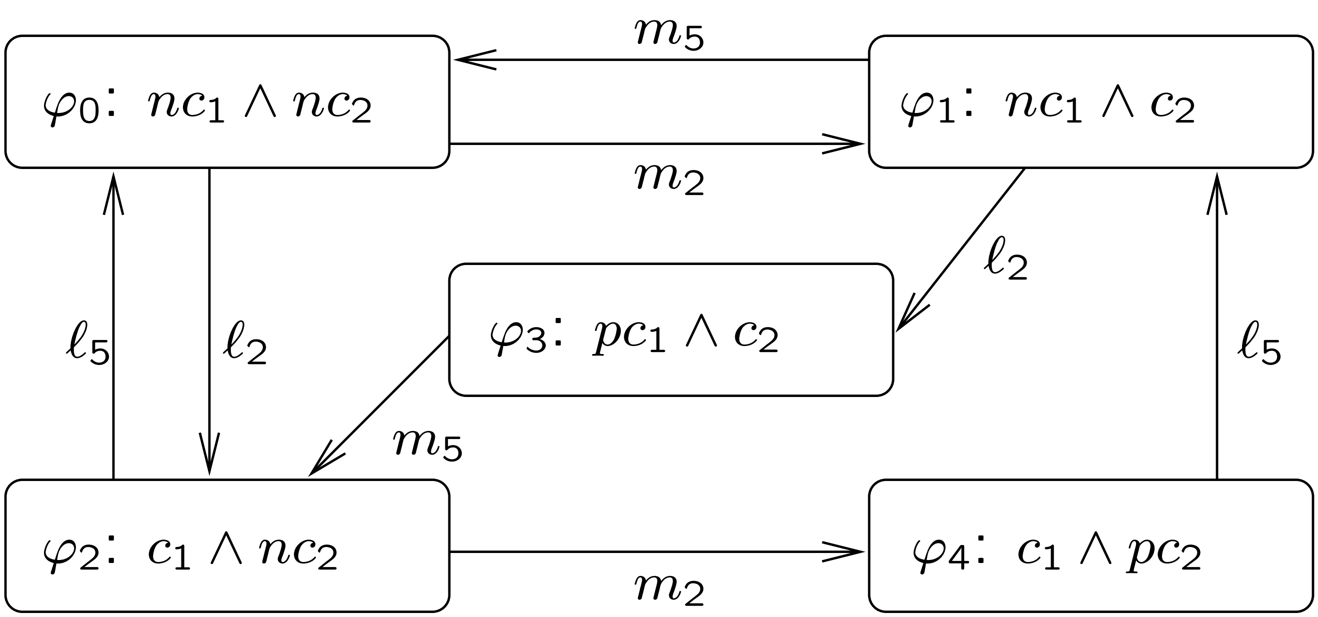
Therefore

$$\boxed{\square\, \underbrace{(y_1 \ \leftrightarrow\ at\_\ell_{3..5})}_{q}}$$

is P-valid.

Example: Program MUX-PET1 (Fig. 3.4)

Establish $\boxed{\square \neg(at\_\ell_4 \wedge at\_m_4)}$



non-critical: nc_1: $at\_\ell_{0..2}$

nc_2: $at\_m_{0..2}$

critical: c_1: $at\_\ell_{3..5} \wedge \neg y_2$

c_2: $at\_m_{3..5} \wedge \neg y_1$

pre-critical: pc_1: $at\_\ell_3 \wedge s = 1 \wedge y_2$

pc_2: $at\_m_3 \wedge s = 2 \wedge y_1$