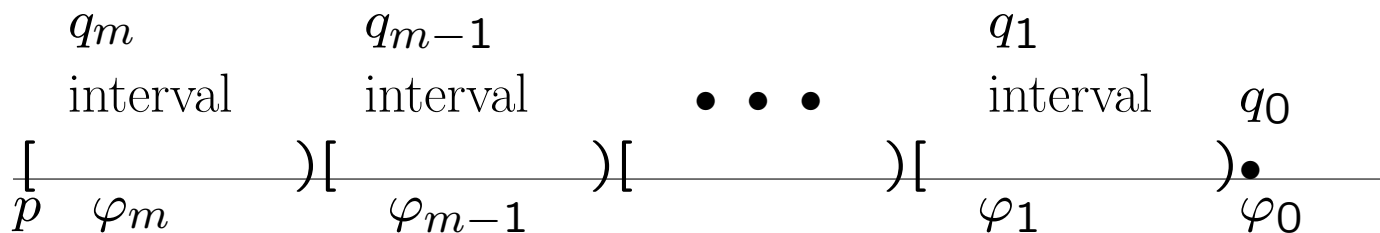


# CS256/Winter 2009 Lecture #10

Zohar Manna

## Nested Waiting-for Formulas



### Rule nwait (nested waiting-for)

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

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

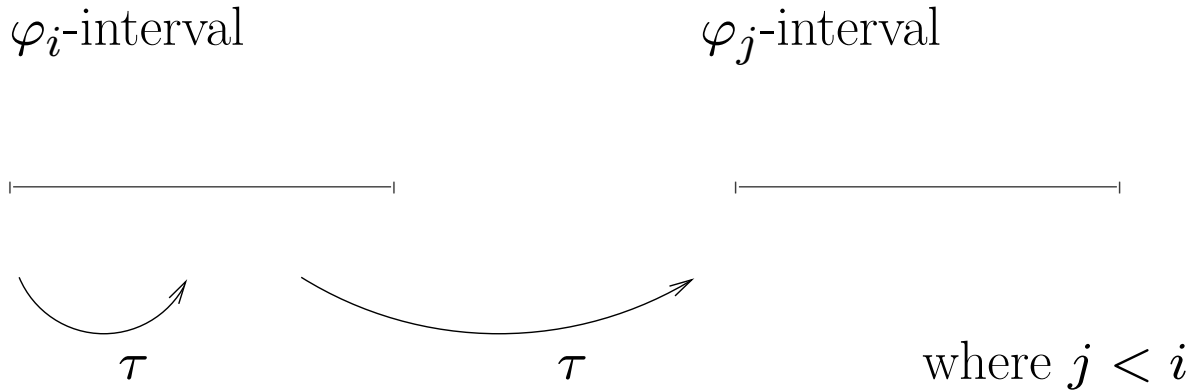
$$\text{N2. } \varphi_i \rightarrow q_i \quad \text{for } i = 0, 1, \dots, m$$

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

---


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

## Nested Waiting-for Formulas (Cont'd)



Premise N3 states that for each assertion  $\varphi_i$ , each transition  $\tau \in \mathcal{T}$  either preserves  $\varphi_i$  or leads to some  $\varphi_j$ , 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:

$$\underbrace{at\_l_3}_p \Rightarrow \underbrace{\neg at\_m_4}_{q_3} \mathcal{W} \underbrace{at\_m_4}_{q_2} \mathcal{W} \underbrace{\neg at\_m_4}_{q_1} \mathcal{W} \underbrace{at\_l_4}_{q_0}$$

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)

**local**  $y_1, y_2$ : **boolean** where  $y_1 = \text{F}, y_2 = \text{F}$   
 $s$  : **integer** where  $s = 1$

$l_0$  : **loop forever do**

$P_1 ::$   $\left[ \begin{array}{l} l_1 : \text{noncritical} \\ l_2 : (y_1, s) := (\text{T}, 1) \\ l_3 : \text{await } (\neg y_2) \vee (s \neq 1) \\ l_4 : \text{critical} \\ l_5 : y_1 := \text{F} \end{array} \right]$

||

$m_0$  : **loop forever do**

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

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

$$p: \quad \underline{at\_l_3}$$

$$\varphi_3: \quad at\_l_3 \wedge \underline{\neg at\_m_4} \wedge at\_m_3 \wedge s = 1$$

“ $P_2$  has priority over  $P_1$ ”

$$\varphi_2: \quad at\_l_3 \wedge \underline{at\_m_4}$$

$$\varphi_1: \quad at\_l_3 \wedge \underline{\neg at\_m_4} \wedge (at\_m_3 \rightarrow s = 2)$$

“ $P_1$  has priority over  $P_2$ ”

$$\varphi_0 = q_0: \quad \underline{at\_l_4}$$

or equivalently,

$$p: \quad at\_l_3$$

$$\varphi_3: \quad at\_l_3 \wedge at\_m_3 \wedge s = 1$$

$$\varphi_2: \quad at\_l_3 \wedge at\_m_4$$

$$\varphi_1: \quad at\_l_3 \wedge (at\_m_{0..2,5} \vee (at\_m_3 \wedge s = 2))$$

$$\varphi_0 = q_0: \quad at\_l_4$$

## Concatenation of waiting-for formulas

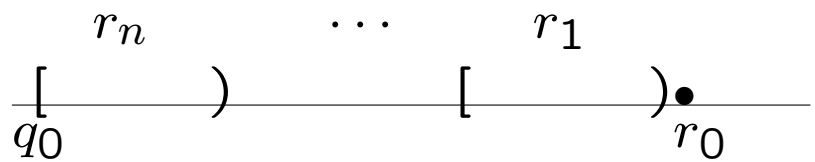
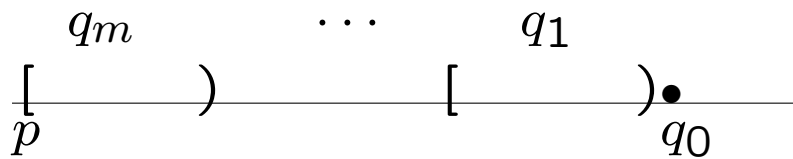
Rule CONC-W

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

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

---


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



## Collapsing of waiting-for formulas

### Rule COLL-W

For  $i > 0$

$$p \Rightarrow q_m \mathcal{W} \cdots \mathcal{W} q_{i+1} \mathcal{W} q_i \mathcal{W} \cdots \mathcal{W} q_0$$

---


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

$$\frac{p}{\left[ \frac{q_m}{\quad} \right) \cdots \left[ \frac{q_{i+1}}{\quad} \right) \left[ \frac{q_i}{\quad} \right) \cdots \left[ \frac{q_1}{\quad} \right) \bullet} q_0$$

$$\frac{p}{\left[ \frac{q_m}{\quad} \right) \cdots \left[ \frac{q_{i+1} \vee q_i}{\quad} \right) \cdots \left[ \frac{q_1}{\quad} \right) \bullet} q_0$$

## Basic Verification Diagrams

A visual summary of verification proofs

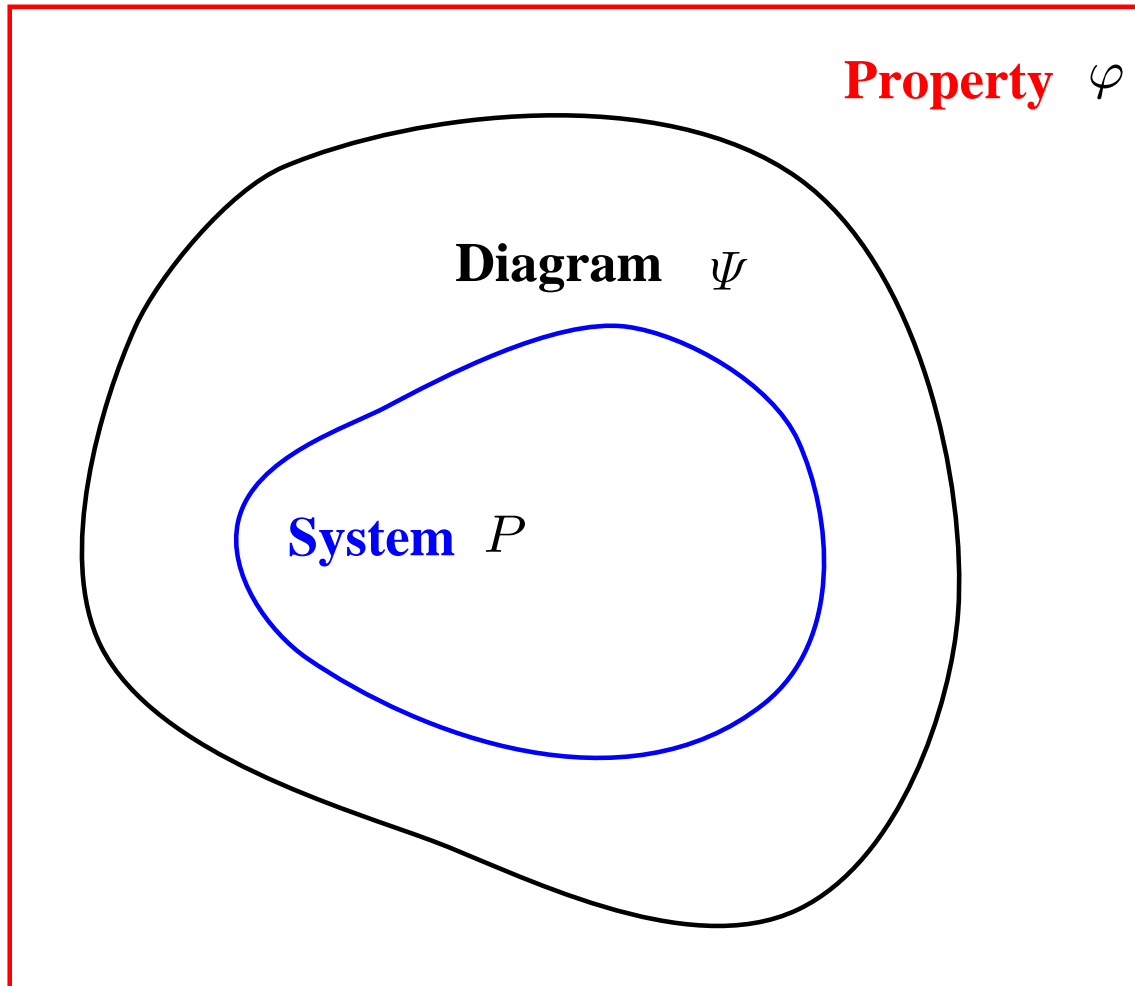
Verification Diagrams (VDs) allow a graphical representation of a proof of a temporal property.

To prove  $\varphi$  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 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.

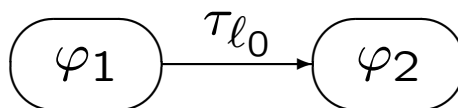
## Verification Diagram (VD)

Directed labeled graph with

- Nodes – labeled by assertions



- Edges – labeled by names of transitions



- Terminal Node (“goal”) – no edges depart

from it

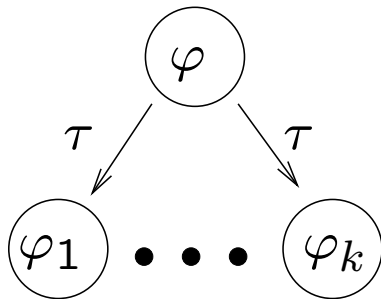


## Verification conditions (VCs)

VD provides a concise representation of sets of VCs:

- The verification condition associated with a node labeled by  $\varphi$  and a transition  $\tau$  is

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



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

- Nonterminal node without outgoing edges

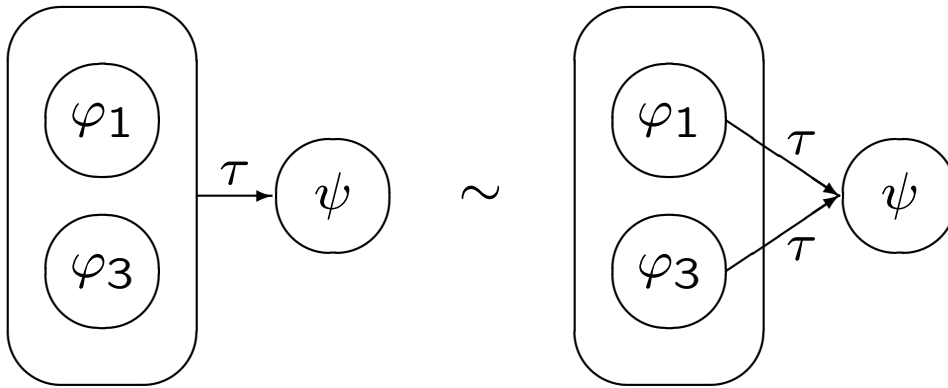
$$\textcircled{\varphi} \Rightarrow \{\varphi\} \tau \{\varphi\}$$

Note: No verification conditions for terminal node.

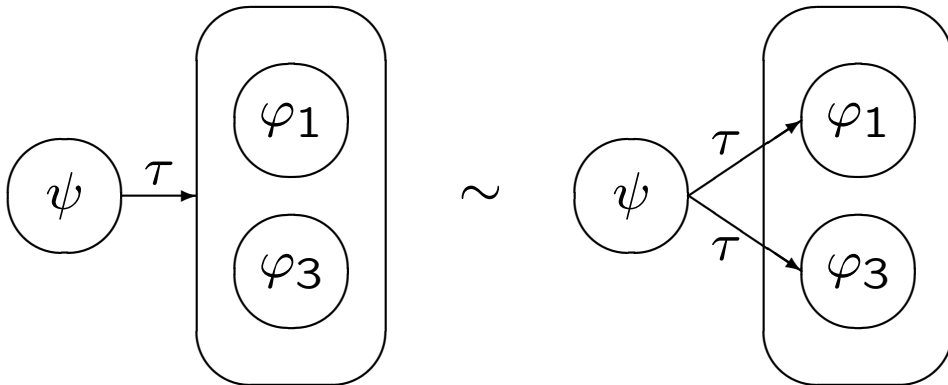
**Definition:** VD is  $P$ -valid iff all VCs associated with nodes in the diagram are  $P$ -state valid

## Compound Nodes: Statecharts Conventions

- Departing edges

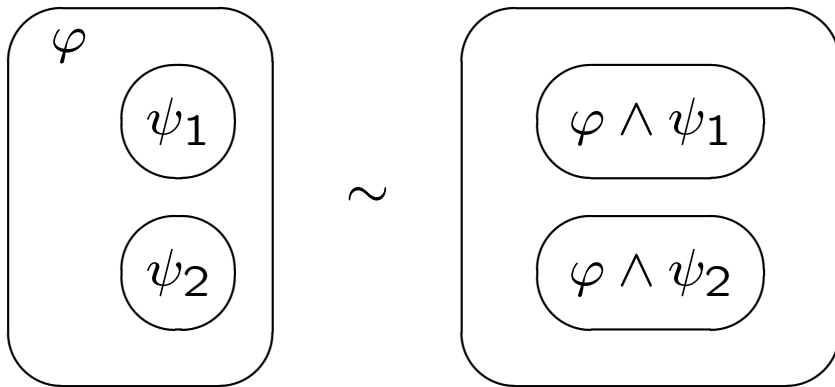


- Arriving edges



## Compound Nodes: Statecharts Conventions

- Common factors



## Classes of Diagrams

- Proofs of invariance properties

$$\square q$$

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

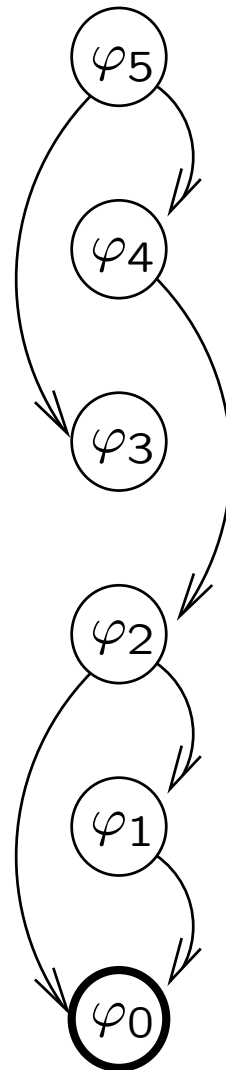
VDs with nodes  $\varphi_m, \dots, \varphi_0$  such that:

- weakly acyclic, i.e.,

if  $\varphi_i \longrightarrow \varphi_j$

then  $i \geq j$

- $\varphi_0$  is a terminal node



Claim (wait diagram):

A  $P$ -valid WAIT diagram establishes that

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

is  $P$ -valid.

If, in addition,

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

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

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)

1-bounded overtaking from  $\ell_3$

$$\psi: \underbrace{at_{\ell_3}}_p \Rightarrow$$
$$\underbrace{(\neg at_{m_4})}_{q_3} \mathcal{W} \underbrace{at_{m_4}}_{q_2} \mathcal{W} \underbrace{(\neg at_{m_4})}_{q_1} \mathcal{W} \underbrace{at_{\ell_4}}_{q_0}$$

Proof is summarized in WAIT diagram

(Fig 3.8)

**Example: Program mux-pet1 (Fig. 3.4)**  
(Peterson's Algorithm for mutual exclusion)

**local**  $y_1, y_2$ : **boolean** where  $y_1 = \text{F}, y_2 = \text{F}$   
 $s$  : **integer** where  $s = 1$

$l_0$  : **loop forever do**

$P_1$  ::  $\left[ \begin{array}{l} l_1 : \text{noncritical} \\ l_2 : (y_1, s) := (\text{T}, 1) \\ l_3 : \text{await } (\neg y_2) \vee (s \neq 1) \\ l_4 : \text{critical} \\ l_5 : y_1 := \text{F} \end{array} \right]$

||

$m_0$  : **loop forever do**

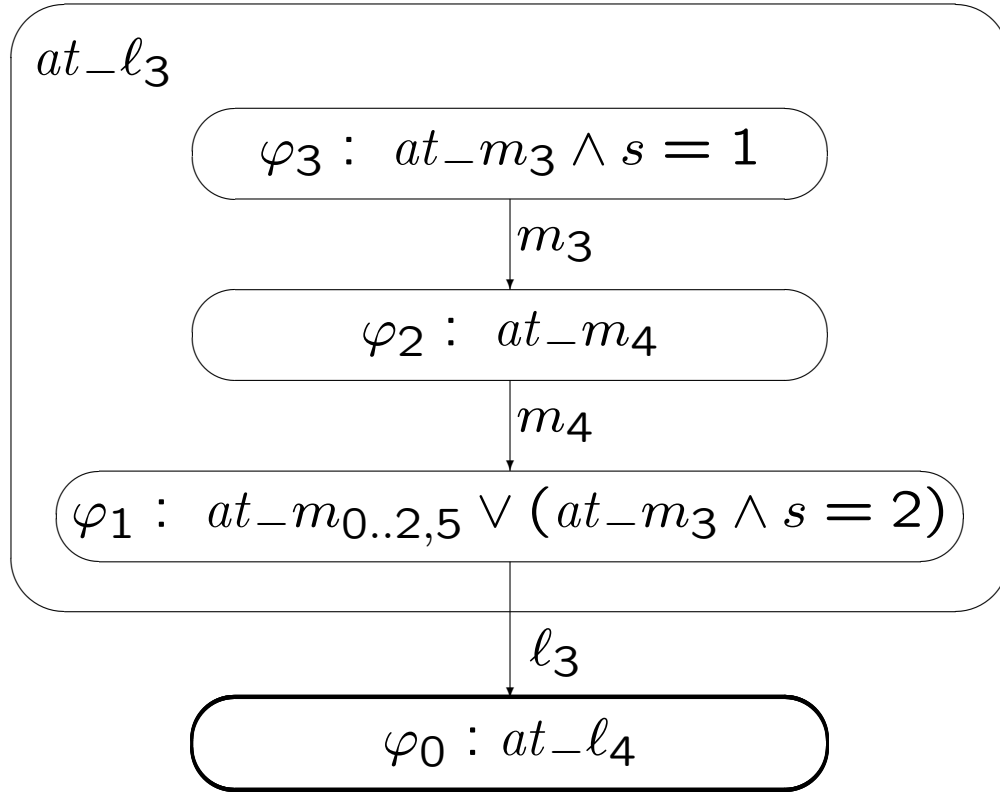
$P_2$  ::  $\left[ \begin{array}{l} m_1 : \text{noncritical} \\ m_2 : (y_2, s) := (\text{T}, 2) \\ m_3 : \text{await } (\neg y_1) \vee (s \neq 2) \\ m_4 : \text{critical} \\ m_5 : y_2 := \text{F} \end{array} \right]$

Example: Program MUX-PET1 (Con't)

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

$$\psi: \underbrace{at_{-l_3}}_p \Rightarrow$$

$$\underbrace{(\neg at_{-m_4})}_{q_3} \mathcal{W} \underbrace{at_{-m_4}}_{q_2} \mathcal{W} \underbrace{(\neg at_{-m_4})}_{q_1} \mathcal{W} \underbrace{at_{-l_4}}_{q_0}$$



Example: Program MUX-PET1 (Con't)

Associated VCs

- From  $\varphi_3$

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

$$\underbrace{\dots}_{\varphi_3} \wedge \underbrace{\dots \wedge at'_{-m_4}}_{\rho_{m_3}} \rightarrow \underbrace{\dots}_{\varphi'_3} \vee \underbrace{at'_{-m_4}}_{\varphi'_2}$$

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

for all non- $m_3$  transitions.

But since we are  $at_{-l_3}$ ,  $at_{-m_3}$ , check only  $l_3$ .

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

$$\underbrace{at\_m_3 \wedge \dots \wedge s = 1}_{\varphi_3} \wedge \underbrace{\dots \wedge ((\neg y_2) \vee (s \neq 1))}_{\rho l_3} \rightarrow \underbrace{\dots}_{\varphi'_3}$$

Recall that by  $\chi_2$ ,  $at\_m_3 \rightarrow y_2$ .

- From  $\varphi_2$ 

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

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

- From  $\varphi_1$ 

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

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

They are  $P$ -state valid

[not state-valid - require invariants  $\chi_0, \dots, \chi_4$ ]

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,

$$\underbrace{at\_l_3}_p \rightarrow \bigvee_{j=0}^3 \varphi_j$$

$$\varphi_0 \rightarrow \underbrace{at\_l_4}_{q_0} \quad \varphi_1 \rightarrow \underbrace{\neg at\_m_4}_{q_1}$$

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

are  $P$ -state valid.

Therefore,

$$\psi: at\_l_3 \Rightarrow (\neg at\_m_4) \mathcal{W} at\_m_4 \mathcal{W} (\neg at\_m_4) \mathcal{W} at\_l_4$$

is valid over MUX-PET1

## Invariance Diagrams

VDs with no terminal nodes (cycles OK)

Claim (invariance diagram):

A  $P$ -valid INVARIANCE diagram establishes that

$$\bigvee_{j=1}^m \varphi_j \Rightarrow \square \left( \bigvee_{j=1}^m \varphi_j \right)$$

is  $P$ -valid.

If, in addition,

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

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

are  $P$ -state valid, then

$$\boxed{\square q}$$

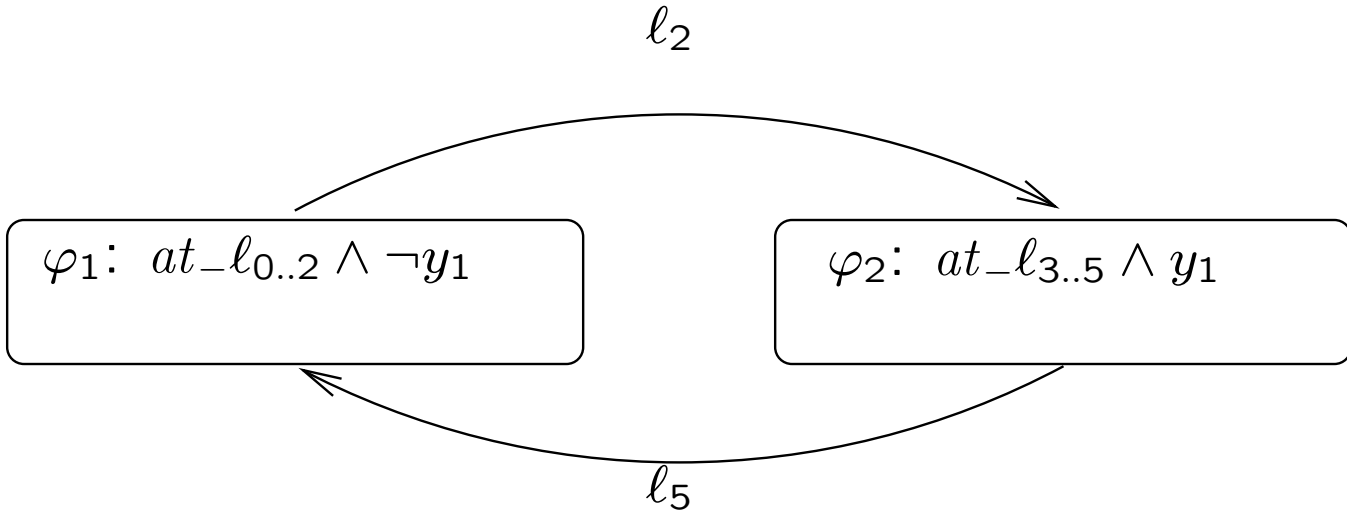
is  $P$ -valid

Example: Program MUX-PET1 (Fig 3.4)

Establish

$$\boxed{\underbrace{\square (y_1 \leftrightarrow at_{-l_{3..5}})}_q}$$

INVARIANCE diagram  
valid for program MUX-PET1



because

$$\begin{array}{ll} \{\varphi_1\} l_2 \{\varphi_1 \vee \varphi_2\} & \{\varphi_1\} \bar{l}_2 \{\varphi_1\} \\ \{\varphi_2\} l_5 \{\varphi_2 \vee \varphi_1\} & \{\varphi_2\} \bar{l}_5 \{\varphi_2\} \end{array}$$

Thus

$$\varphi_1 \vee \varphi_2 \Rightarrow \square(\varphi_1 \vee \varphi_2)$$

Also,

$$(I1) \underbrace{at_{-l_0} \wedge \neg y_1 \wedge \dots}_{\Theta} \rightarrow \underbrace{at_{-l_{0..2}} \wedge \neg y_1}_{\varphi_1} \vee \underbrace{\dots}_{\varphi_2}$$

$$(I2) \underbrace{at_{-l_{0..2}} \wedge \neg y_1}_{\varphi_1} \vee \underbrace{at_{-l_{3..5}} \wedge y_1}_{\varphi_2} \rightarrow \underbrace{y_1 \leftrightarrow at_{-l_{3..5}}}_q$$

are state-valid

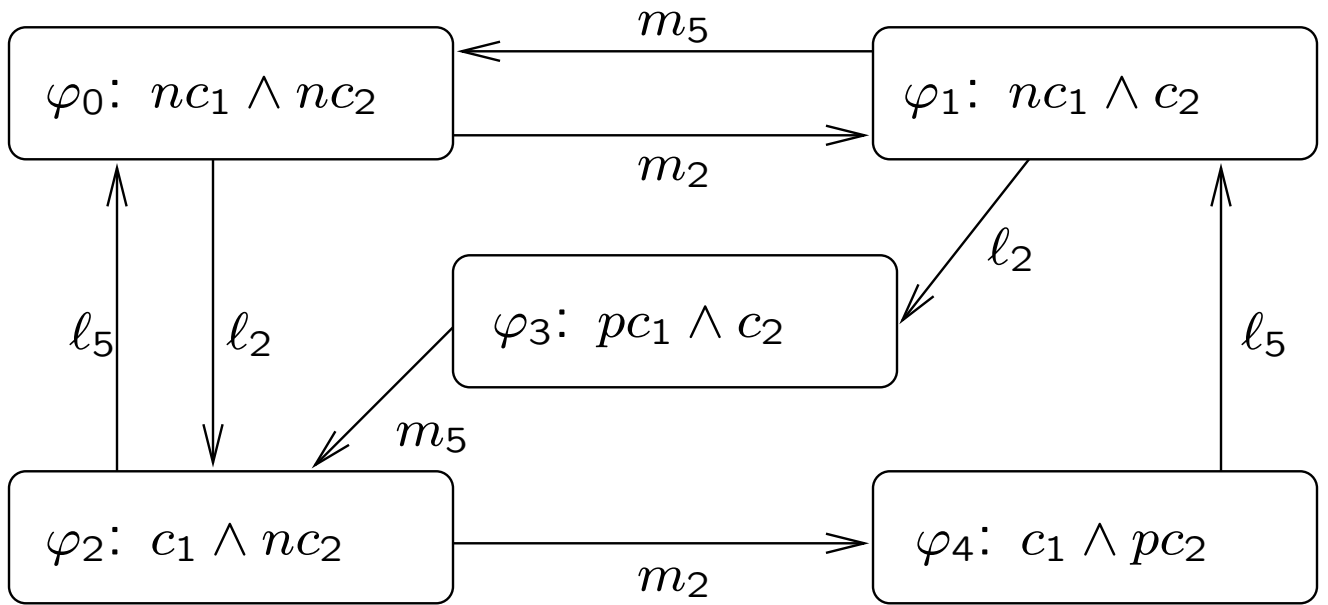
Therefore

$$\boxed{\square \underbrace{(y_1 \leftrightarrow at_{-l_{3..5}})}_q}$$

is  $P$ -valid.

Example: Program MUX-PET1 (Fig. 3.4)

Establish  $\square \neg(at_{-l_4} \wedge at_{-m_4})$



non-critical:  $nc_1: at_{-l_{0..2}}$

$nc_2: at_{-m_{0..2}}$

critical:  $c_1: at_{-l_{3..5}} \wedge \neg y_2$

$c_2: at_{-m_{3..5}} \wedge \neg y_1$

pre-critical:  $pc_1: at_{-l_3} \wedge s = 1 \wedge y_2$

$pc_2: at_{-m_3} \wedge s = 2 \wedge y_1$