

## CS256/Winter 2009 Lecture #6

Zohar Manna

### Chapter 1

Invariance: Proof Methods

For assertion  $q$   
and SPL program  $P$

show  $P \models \Box q$   
(i.e.,  $q$  is  $P$ -invariant)

## Proving Invariances

### Definitions

Recall:

- the variables of assertion:
  - free (flexible) system variables

$$V = Y \cup \{\pi\}$$

where  $Y$  are the program variables and  $\pi$  is the control variable

- quantified (rigid) specification variables
- $q'$  is the primed version of  $q$ , obtained by replacing each free occurrence of a system variable  $y \in V$  by its primed version  $y'$ .
- $\rho_\tau$  is the transition relation of  $\tau$ , expressing the relation holding between a state  $s$  and any of its  $\tau$ -successors  $s' \in \tau(s)$ .

## Verification Conditions

(proof obligations)

standard verification condition

For assertions  $\varphi, \psi$  and transition  $\tau$ ,

$\{\varphi\} \tau \{\psi\}$  (“Hoare triple”) stands for the state formula

$$\rho_\tau \wedge \varphi \rightarrow \psi'$$

“Verification condition (VC) of  $\varphi$  and  $\psi$  relative to transition  $\tau$ ”

$$\begin{array}{ccc} \varphi & \tau & \psi \\ | & \hline & & | \\ j & & j + 1 \end{array}$$

## Verification Conditions (Con't)

Example:

$$\rho_\tau: x \geq 0 \wedge y' = x + y \wedge x' = x$$

$$\varphi: y = 3 \quad \psi: y = x + 3$$

Then  $\{\varphi\} \tau \{\psi\}$ :

$$\underbrace{x \geq 0 \wedge y' = x + y \wedge x' = x}_{\rho_\tau} \wedge \underbrace{y = 3}_{\varphi} \rightarrow \underbrace{y' = x' + 3}_{\psi'}$$

## Verification Conditions (Con't)

- for  $\tau \in \mathcal{T}$  in  $P$

$$\{\varphi\} \tau \{\psi\}: \rho_\tau \wedge \varphi \rightarrow \psi'$$

“ $\tau$  leads from  $\varphi$  to  $\psi$  in  $P$ ”

- for  $\mathcal{T}$  in  $P$

$$\{\varphi\} \mathcal{T} \{\psi\}: \{\varphi\} \tau \{\psi\} \text{ for every } \tau \in \mathcal{T}$$

“ $\mathcal{T}$  leads from  $\varphi$  to  $\psi$  in  $P$ ”

**Claim** (Verification Condition)

If  $\{\varphi\} \tau \{\psi\}$  is  $P$ -state valid,

then every  $\tau$ -successor of a  $\varphi$ -state is a  $\psi$ -state.

## Verification Conditions (Con't)

### Special Cases

- while, conditional  $\rho_\tau: \rho_\tau^T \vee \rho_\tau^F$

$$\{\varphi\}\tau^T\{\psi\}: \rho_\tau^T \wedge \varphi \rightarrow \psi'$$

$$\{\varphi\}\tau^F\{\psi\}: \rho_\tau^F \wedge \varphi \rightarrow \psi'$$

---


$$\{\varphi\}\tau\{\psi\} : \{\varphi\}\tau^T\{\psi\} \wedge \{\varphi\}\tau^F\{\psi\}$$

- idle

$$\{\varphi\}\tau_I\{\varphi\}: \rho_{\tau_I} \wedge \varphi \rightarrow \varphi'$$

always valid, since

$$\rho_{\tau_I} \rightarrow v' = v \quad \text{for all } v \in V,$$

so  $\varphi' = \varphi$ .

## Verification Conditions (Con't)

### Substituted Form of Verification Condition

Transition relation can be written as

$$\rho_\tau: C_\tau \wedge (\overline{V}' = \overline{E})$$

where

$C_\tau$ : enabling condition

$\overline{V}'$ : primed variable list

$\overline{E}$ : expression list

- The substituted form of verification condition  $\{\varphi\}\tau\{\psi\}$ :

$$\boxed{C_\tau \wedge \varphi \rightarrow \psi[\overline{E}/\overline{V}]}$$

where

$\psi[\overline{E}/\overline{V}]$ : replace each variable  $v \in \overline{V}$   
in  $\psi$  by the corresponding  $e \in \overline{E}$

**Note:** No primed variables!

The substituted form of a verification condition is  $P$ -state valid iff the standard form is

## Verification Conditions (Con't)

Example:

$$\rho_\tau : x \geq 0 \wedge y' = x + y \wedge x' = x$$

$$\varphi : y = 3 \quad \psi : y = x + 3$$

Standard

$$\underbrace{x \geq 0 \wedge y' = x + y \wedge x' = x}_{\rho_\tau} \wedge \underbrace{y = 3}_{\varphi} \rightarrow \underbrace{y' = x' + 3}_{\psi'}$$

Substituted

$$\underbrace{x \geq 0}_{C_\tau} \wedge \underbrace{y = 3}_{\varphi} \rightarrow \underbrace{x + y = x + 3}_{\psi[\bar{E}/\bar{V}]}$$

## Verification Conditions (Con't)

Example:

$$\varphi : x = y \quad \psi : x = y + 1$$

$$\rho_\tau : \underbrace{x \geq 0}_{C_\tau} \wedge \underbrace{(x', y') = (x + 1, y)}_{\bar{V}'}$$

The substituted form of  $\{\varphi\}\tau\{\psi\}$  is

$$\underbrace{x \geq 0}_{C_\tau} \wedge \underbrace{x = y}_{\varphi} \rightarrow \underbrace{(x = y + 1)[(x + 1, y)/(x, y)]}_{\psi[\bar{E}/\bar{V}]}$$

or equivalently

$$x \geq 0 \wedge x = y \rightarrow x + 1 = y + 1$$

## Simplifying Control Expressions

$move(L_1, L_2): L_1 \subseteq \pi \wedge \pi' = (\pi - L_1) \cup L_2$

e.g., for  $L_1 = \{\ell_1\}, L_2 = \{\ell_2\}$

$move(\ell_1, \ell_2): \ell_1 \in \pi \wedge \pi' = (\pi - \{\ell_1\}) \cup \{\ell_2\}$

Consequences implied by  $move(L_1, L_2)$ :

- for every  $[\ell] \in L_1$   
 $at\_l = \text{T}$  (i.e.,  $[\ell] \in \pi$ )
- for every  $[\ell] \in L_2$   
 $at'_l = \text{T}$  (i.e.,  $[\ell] \in \pi'$ )
- for every  $[\ell] \in L_1 - L_2$   
 $at\_l = \text{T}$  (i.e.,  $[\ell] \in \pi$ ) and  
 $at'_l = \text{F}$  (i.e.,  $[\ell] \notin \pi'$ )
- for every  $\ell \notin L_1 \cup L_2$   
 $at'_l = at\_l$  (i.e.,  $[\ell] \in \pi, \pi'$  or  $[\ell] \notin \pi, \pi'$ )

## Proving invariance properties: $P \models \Box q$

We want to show that for every computation of  $P$

$\sigma : s_0, s_1, s_2, \dots$

assertion  $q$  holds in every state  $s_j, j \geq 0$ ,  
i.e.,  $s_j \models q$ .

### Recall:

A sequence  $\sigma : s_0, s_1, s_2, \dots$  is a computation if the following hold (from Chapter 0):

1. Initiality:  $s_0 \models \Theta$
2. Consecution: For each  $j \geq 0$ ,  
 $s_{j+1}$  is a  $\tau$ -successor of  $s_j$  for some  $\tau \in \mathcal{T}$   
( $s_{j+1} \in \tau(s_j)$ )
- 3, 4. Fairness conditions are respected.

**Note:** Truth of *safety* properties over programs *does not* depend on fairness conditions.

## Proving invariance properties (Con't)

This definition suggests a way to prove invariance properties  $\Box q$ :

1. Base case:

Prove that  $q$  holds initially

$$\Theta \rightarrow q$$

i.e.,  $q$  holds at  $s_0$ .

2. Inductive step:

prove that  $q$  is preserved by all transitions

$$\underbrace{q \wedge \rho_\tau \rightarrow q'}_{\{q\}\tau\{q\}} \quad \text{for all } \tau \in \mathcal{T}$$

i.e., if  $q$  holds at  $s_j$ , then it holds at every  $\tau$ -successor  $s_{j+1}$ .

## Rule B-INV (basic invariance)

Show  $P \models \Box q$  (i.e.  $q$  is  $P$ -invariant)

For assertion  $q$ ,

$$\text{B1.} \quad P \models \Theta \rightarrow q$$

$$\text{B2.} \quad P \models \{q\} \mathcal{T} \{q\}$$

---


$$P \models \Box q$$

where B2 stands for

$$P \models \{q\} \tau \{q\} \quad \text{for every } \tau \in \mathcal{T}$$

- The rule states that if we can prove the  $P$ -state validity of  $\Theta \rightarrow q$  and  $\{q\}\mathcal{T}\{q\}$  then we can conclude that  $\Box q$  is  $P$ -valid.
- Thus the proof of a temporal property is reduced to the proof of  $1 + |\mathcal{T}|$  first-order verification conditions.

**Example 1: REQUEST-RELEASE**

**Example 1: request-release (Con't)**

local  $x$ : integer where  $x = 1$

$$\left[ \begin{array}{l} \ell_0 : \text{request } x \\ \ell_1 : \text{critical} \\ \ell_2 : \text{release } x \\ \ell_3 : \end{array} \right]$$

$$\Theta: x = 1 \wedge \pi = \{\ell_0\}$$

$$\mathcal{T}: \{\tau_I, \tau_{\ell_0}, \tau_{\ell_1}, \tau_{\ell_2}\}$$

Prove

$$P \models \underbrace{\square x \geq 0}_q$$

using B-INV.

$$\mathbf{B1:} \underbrace{x = 1 \wedge \pi = \{\ell_0\}}_{\Theta} \rightarrow \underbrace{x \geq 0}_q$$

holds since  $x = 1 \rightarrow x \geq 0$

**B2:**

$$\tau_{\ell_0}: \underbrace{x \geq 0}_q \wedge \underbrace{\text{move}(\ell_0, \ell_1) \wedge x > 0 \wedge x' = x - 1}_{\rho\tau_{\ell_0}} \rightarrow \underbrace{x' \geq 0}_{q'}$$

holds since  $x > 0 \rightarrow x - 1 \geq 0$

$$\tau_{\ell_1}: \underbrace{x \geq 0}_q \wedge \underbrace{\text{move}(\ell_1, \ell_2) \wedge x' = x}_{\rho\tau_{\ell_1}} \rightarrow \underbrace{x' \geq 0}_{q'}$$

holds since  $x \geq 0 \rightarrow x \geq 0$

$$\tau_{\ell_2}: \underbrace{x \geq 0}_q \wedge \underbrace{\text{move}(\ell_2, \ell_3) \wedge x' = x + 1}_{\rho\tau_{\ell_2}} \rightarrow \underbrace{x' \geq 0}_{q'}$$

holds since  $x \geq 0 \rightarrow x + 1 \geq 0$

**Example 1: request-release (Con't)**

**local**  $x$ : integer where  $x = 1$

$l_0$	request $x$
$l_1$	critical
$l_2$	release $x$
$l_3$	

We proved

$$P \models \square x \geq 0$$

using B-INV.

Now we want to prove

$$P \models \square \underbrace{(at\_l_1 \rightarrow x = 0)}_q$$

**Example 1: request-release (Con't)**

Attempted proof:

$$\mathbf{B1:} \underbrace{x = 1 \wedge \pi = \{l_0\}}_{\Theta} \rightarrow \underbrace{(at\_l_1 \rightarrow x = 0)}_q$$

holds since  $\pi = \{l_0\} \rightarrow at\_l_1 = \text{F}$

$$\begin{aligned} \mathbf{B2:} & \{q\} \tau_{l_0} \{q\} \\ & \underbrace{at\_l_1 \rightarrow x = 0}_q \wedge \underbrace{move(l_0, l_1) \wedge x > 0 \wedge x' = x - 1}_{\rho \tau_{l_0}} \\ & \rightarrow \underbrace{at'_l_1 \rightarrow x' = 0}_{q'} \end{aligned}$$

We have  $move(l_0, l_1) \rightarrow at\_l_1 = \text{F}, at'_l_1 = \text{T}$

BUT

$$(\text{F} \rightarrow x = 0) \wedge x > 0 \wedge x' = x - 1 \rightarrow (\text{T} \rightarrow x' = 0)$$

Cannot prove: not state-valid

What is the problem?

We need a stronger rule.

## Strategies for invariance proofs

### Rule B-INV (basic invariance)

For assertion  $q$ ,

$$\text{B1. } P \models \Theta \rightarrow q$$

$$\text{B2. } P \models \{q\} \mathcal{T} \{q\}$$

---

$$P \models \square q$$

- $q$  is inductive if B1 and B2 are (state) valid
- By rule B-INV,  
every inductive assertion  $q$  is  $P$ -invariant
- The converse is not true

**Example:** In REQUEST-RELEASE

$$at\_l_1 \rightarrow x = 0$$

is  $P$ -invariant, but not inductive

### Rule B-INV(Con't)

The problem is:

“The invariant is not inductive”

i.e., it is not strong enough to be preserved by all transitions.

Another way to look at it is to observe that

$$\{q\} \tau_{l_0} \{q\}$$

is not state valid, but it is  $P$ -state valid, i.e., it is true in all  $P$ -accessible states, since in all  $P$ -accessible states

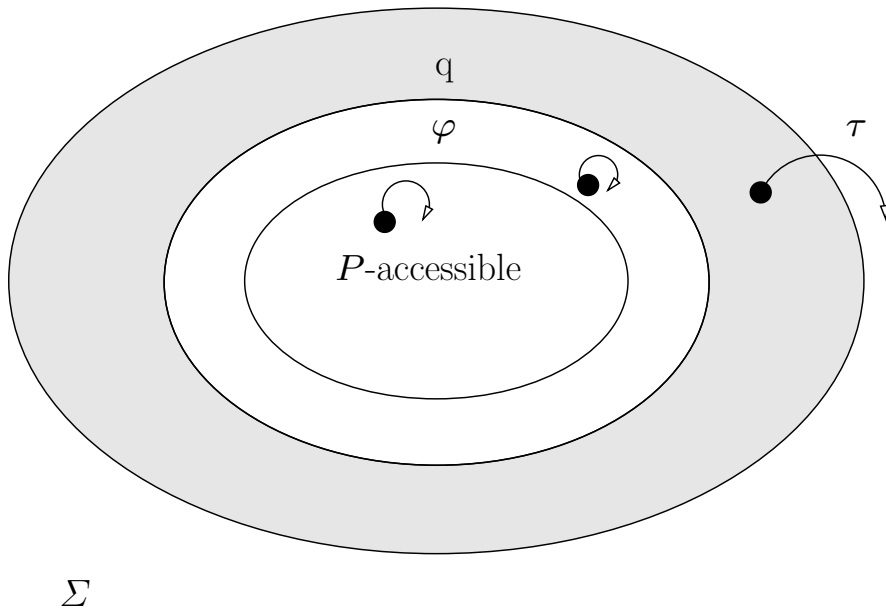
$$x = 1 \text{ when at location } l_0.$$

This suggests two strategies to overcome this problem:

- strengthening
- incremental proof

## Strategy 1: Strengthening

Find a stronger assertion  $\varphi$  that is inductive and implies the assertion  $q$  we want to prove.



In Chapter 2 it will be shown that there always exists such an assertion  $\varphi$ .

## Strategy 1: Strengthening (Con't)

Example:

To show

$$\square \underbrace{(at\_l_1 \rightarrow x = 0)}_q$$

strengthen  $q$  to

$$\varphi : (at\_l_1 \rightarrow x = 0) \wedge (at\_l_0 \rightarrow x = 1)$$

and show

$$\square \underbrace{(at\_l_1 \rightarrow x = 0) \wedge (at\_l_0 \rightarrow x = 1)}_\varphi$$

by rule B-INV.

### Strategy 1: Strengthening (Con't)

The strengthening strategy relies on the following rule, MON-I, which, combined with B-INV leads to the general invariance rule INV.

#### Rule MON-I (Monotonicity)

For assertions  $q_1, q_2$ ,

$$\frac{P \models \Box q_1 \quad P \models q_1 \rightarrow q_2}{P \models \Box q_2}$$

### Strategy 1: Strengthening (Con't)

Rule INV (general invariance)

For assertions  $q, \varphi$

$$\text{I1.} \quad P \models \varphi \rightarrow q$$

$$\text{I2.} \quad P \models \Theta \rightarrow \varphi$$

$$\text{I3.} \quad P \models \{\varphi\} \mathcal{T} \{\varphi\}$$

$$\frac{}{P \models \Box q}$$

## Strategy 1: Strengthening (Con't)

**Soundness:** If we manage to prove  $\Box q$  using the INV rule for some program  $P$ , is  $q$  really an invariant for the program?

We can prove that this is indeed the case. So INV rule is *sound*.

**Completeness:** What if  $q$  is an invariant for a program  $P$  but there is **no** way of proving it under the INV rule?

We can prove that this never happens. There always exists an appropriate  $\varphi$ . In other words INV rule is *complete*.

Motivation:

$$P \models \Box \varphi \quad (\text{by I2 and I3})$$

$$P \models \varphi \rightarrow q \quad (\text{by I1})$$

Therefore,

$$P \models \Box q \quad (\text{by MON-I})$$

i.e., this rule requires that  $\Box \varphi$  holds and  $\varphi$  implies  $q$ , then  $\Box q$  can be concluded to hold by monotonicity.

## Control Invariants

Some control invariants that can always be used (without mentioning them)

- **CONFLICT:**

for labels  $l_i, l_j$  that are in conflict (i.e., not  $\sim_L$ , not parallel):

$$\square \neg(at_{-l_i} \wedge at_{-l_j})$$

- **SOMEWHERE:**

for the set of labels  $\mathcal{L}_i$  in a top-level process:

$$\square \bigvee_{\ell \in \mathcal{L}_i} at_{-\ell}$$

- **EQUAL:**

for labels  $l, m$ , s.t.  $l \sim_L m$ :

$$\square (at_{-l} \leftrightarrow at_{-m})$$

## Control Invariants (Con't)

- **PARALLEL:**

for substatement  $[S_1 || S_2]$ :

$$\square (in_{-S_1} \leftrightarrow in_{-S_2})$$

i.e., if control is in  $S_1$  it must also be in  $S_2$  and vice versa.

**Example:**

Using the invariant **CONFLICT**,

$$move(\ell_2, \ell_3) \quad \text{implies} \quad \begin{array}{l} l_0 \notin \pi, l_1 \notin \pi, l_3 \notin \pi \\ l_0 \notin \pi', l_1 \notin \pi', l_2 \notin \pi' \end{array}$$

## Strategy 1: Strengthening (Con't)

Example:

We proposed the strengthened invariant

$$\varphi : (at_{-l_0} \rightarrow x = 1) \wedge (at_{-l_1} \rightarrow x = 0)$$

Consider  $\{\varphi\} \tau_{l_0} \{\varphi\}$ :

$$\underbrace{(at_{-l_0} \rightarrow x = 1) \wedge (at_{-l_1} \rightarrow x = 0)}_{\varphi} \wedge$$

$$\underbrace{move(l_0, l_1) \wedge x > 0 \wedge x' = x - 1}_{\rho_{\tau_{l_0}}}$$

$$\rightarrow \underbrace{(at'_{-l_0} \rightarrow x' = 1) \wedge (at'_{-l_1} \rightarrow x' = 0)}_{\varphi'}$$

$move(l_0, l_1)$  implies  $l_0 \in \pi, l_1 \notin \pi, l_1 \in \pi', l_0 \notin \pi'$

Therefore

$$(T \rightarrow x = 1) \wedge (F \rightarrow \dots) \wedge \dots \wedge x' = x - 1 \wedge \dots$$

$$\rightarrow (F \rightarrow \dots) \wedge (T \rightarrow x' = 0)$$

holds.

## Strategy 1: Strengthening (Con't)

Example (Con't):

Consider  $\{\varphi\} \tau_{l_2} \{\varphi\}$ :

$$\underbrace{(at_{-l_0} \rightarrow x = 1) \wedge (at_{-l_1} \rightarrow x = 0)}_{\varphi} \wedge$$

$$\underbrace{move(l_2, l_3) \wedge x' = x + 1}_{\rho_{\tau_{l_2}}}$$

$$\rightarrow \underbrace{(at'_{-l_0} \rightarrow x' = 1) \wedge (at'_{-l_1} \rightarrow x' = 0)}_{\varphi'}$$

$move(l_2, l_3)$  implies  $l_3 \in \pi'$

and by CONFLICT invariants  $l_0, l_1 \notin \pi'$ .

Therefore

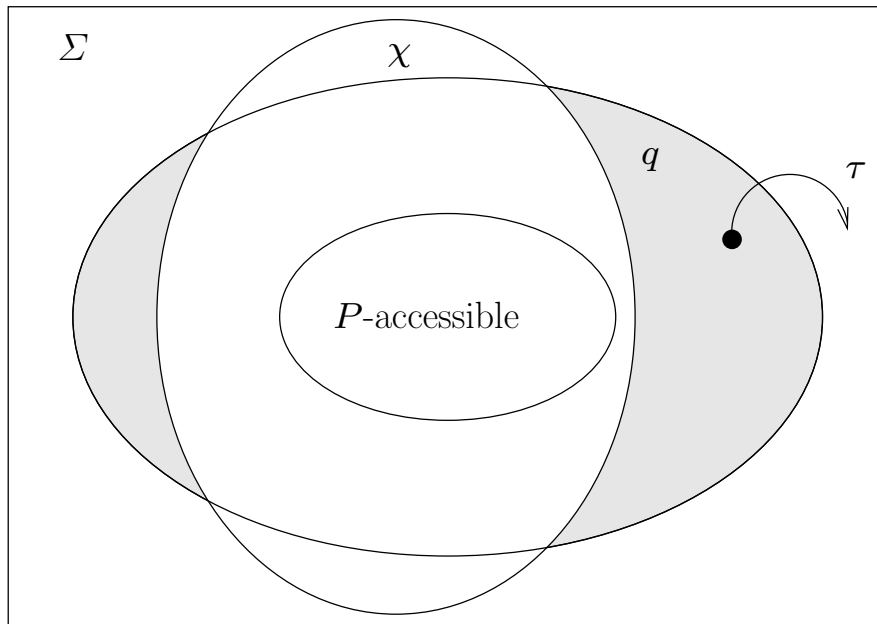
$$\dots \wedge \dots \rightarrow (F \rightarrow x' = 1) \wedge (F \rightarrow x' = 0)$$

holds.

$\{\varphi\} \tau_{l_2} \{\varphi\}$  is not state-valid,  
but it is  $P$ -state valid. Why?

## Strategy 2: Incremental proof

Use previously proven invariances  $\chi$  to exclude parts of the state space from consideration.



## Strategy 2: Incremental proof (Con't)

**Example:**

To show

$$\square \underbrace{(at\_l_1 \rightarrow x = 0)}_q$$

prove first (separately) by rule B-INV

$$\square \underbrace{(at\_l_0 \rightarrow x = 1)}_\chi,$$

then show

$$\square \underbrace{(at\_l_1 \rightarrow x = 0)}_q$$

by rule B-INV, but add the conjunct

$$at\_l_0 \rightarrow x = 1$$

to the antecedent of all verification conditions.

(**Example** continues...)

## Strategy 2: Incremental proof (Con't)

**Example:** (cont'd)

e.g., to show  $\{\chi \wedge q\} \tau_{\ell_0} \{q\}$ , prove

$$\underbrace{at\_l_0 \rightarrow x = 1}_{\chi} \wedge \underbrace{at\_l_1 \rightarrow x = 0}_{q} \wedge$$

$$\underbrace{move(\ell_0, \ell_1) \wedge x > 0 \wedge x' = x - 1}_{\rho_{\tau_{\ell_0}}}$$

$$\rightarrow \underbrace{at\_l_1 \rightarrow x' = 0}_{q'}$$

## Strategy 2: Incremental proof (Con't)

In an incremental proof we use previously proven properties to eliminate parts of the state space (non  $P$ -accessible states) from consideration, relying on the following rules:

**Rule SV-PSV:** from state validities to  $P$ -state validities

For assertions  $q_1, q_2$  and  $\chi$ ,

$$P \models \Box \chi$$

$$P \models \chi \wedge q_1 \rightarrow q_2$$

---


$$P \models \Box(q_1 \rightarrow q_2)$$

**Rule I-CON:** Conjunction

For assertions  $q_1$  and  $q_2$ ,

$$P \models \Box q_1$$

$$P \models \Box q_2$$

---


$$P \models \Box(q_1 \wedge q_2)$$

## Strategy 2: Incremental proof (Con't)

Example: Program MUX-SEM  
(mutual exclusion by semaphores)

local  $y$ : integer where  $y = 1$

$$P_1 :: \left[ \begin{array}{l} \ell_0: \text{loop forever do} \\ \left[ \begin{array}{l} \ell_1: \text{noncritical} \\ \ell_2: \text{request } y \\ \ell_3: \text{critical} \\ \ell_4: \text{release } y \end{array} \right] \end{array} \right] \parallel P_2 :: \left[ \begin{array}{l} m_0: \text{loop forever do} \\ \left[ \begin{array}{l} m_1: \text{noncritical} \\ m_2: \text{request } y \\ m_3: \text{critical} \\ m_4: \text{release } y \end{array} \right] \end{array} \right]$$

Prove mutual exclusion

$$\square \underbrace{\neg(at_{\ell_3} \wedge at_{m_3})}_q$$

Program MUX-SEM (Con't)

$$3 \text{ steps: } \square \underbrace{(y \geq 0)}_{\varphi_1}$$

$$\square \underbrace{(at_{\ell_{3,4}} + at_{m_{3,4}} + y = 1)}_{\varphi_2}$$

$$\square \underbrace{\neg(at_{\ell_3} \wedge at_{m_3})}_p$$

where  $F = 0, T = 1$ .

$$\text{Let } \pi_\ell: \pi \cap \{\ell_0, \dots, \ell_4\}$$

$$\pi_m: \pi \cap \{m_0, \dots, m_4\}$$

By control invariants (CONFLICT, SOMEWHERE and PARALLEL)

$$|\pi_\ell| = |\pi_m| = 1$$

Program MUX-SEM (Con't)

Step 1:  $\square(\underbrace{y \geq 0}_{\varphi_1})$

by rule B-INV

$$\text{B1. } \underbrace{\pi = \{l_0, m_0\} \wedge y = 1}_{\Theta} \rightarrow \underbrace{y \geq 0}_{\varphi_1}$$

$$\text{B2. } \rho_\tau \wedge y \geq 0 \rightarrow y' \geq 0$$

check only  $l_2, l_4, m_2, m_4$   
 (“ $y$ -modifiable transitions”)

Program MUX-SEM (Con't)

$$l_2: \underbrace{\text{move}(l_2, l_3) \wedge y > 0 \wedge y' = y-1}_{\rho_\tau} \wedge \underbrace{y \geq 0}_{\varphi} \rightarrow \underbrace{y' \geq 0}_{\varphi'}$$

holds since  $y > 0 \rightarrow y-1 \geq 0$

$$l_4: \underbrace{\text{move}(l_4, l_0) \wedge y' = y+1}_{\rho_\tau} \wedge \underbrace{y \geq 0}_{\varphi} \rightarrow \underbrace{y' \geq 0}_{\varphi'}$$

holds since  $y \geq 0 \rightarrow y+1 \geq 0$ .

Similarly for  $m_2, m_4$ .

Program MUX-SEM (Con't)

Step 2:

$$\square(\underbrace{at_{-l_{3,4}} + at_{-m_{3,4}} + y = 1}_{\varphi_2})$$

by rule B-INV

$$B1. \underbrace{\pi = \{l_0, m_0\} \wedge y = 1}_{\Theta} \rightarrow$$

$$\underbrace{\underbrace{at_{-l_{3,4}}}_0 + \underbrace{at_{-m_{3,4}}}_0 + \underbrace{y}_1 = 1}_{\varphi_2}$$

Program MUX-SEM (Con't)

$$B2. \rho_\tau \wedge \varphi_2 \rightarrow \varphi'_2$$

$$\rho_{\ell_0} \wedge 0 + at_{-m_{3,4}} + y = 1 \rightarrow 0 + at_{-m_{3,4}} + y = 1$$

$$\rho_{\ell_1} \wedge 0 + at_{-m_{3,4}} + y = 1 \rightarrow 0 + at_{-m_{3,4}} + y = 1$$

$$\rho_{\ell_2} \wedge 0 + at_{-m_{3,4}} + y = 1 \rightarrow 1 + at_{-m_{3,4}} + (y-1) = 1$$

$$\rho_{\ell_3} \wedge 1 + at_{-m_{3,4}} + y = 1 \rightarrow 1 + at_{-m_{3,4}} + y = 1$$

$$\rho_{\ell_4} \wedge 1 + at_{-m_{3,4}} + y = 1 \rightarrow \underbrace{0}_{at'_{-l_{3,4}}} + \underbrace{at_{-m_{3,4}}}_{at'_{-m_{3,4}}} + \underbrace{(y+1)}_{y'} = 1$$

Program MUX-SEM (Con't)

**Step 3:** Show  $P \models \square \underbrace{\neg(at_{-l_3} \wedge at_{-m_3})}_q$

• By I-CON

$$P \models \square \varphi_1, P \models \square \varphi_2$$

---

$$P \models \square(\varphi_1 \wedge \varphi_2)$$

• By MON-I

$$P \models \square(\varphi_1 \wedge \varphi_2)$$

$$P \models \underbrace{y \geq 0}_{\varphi_1} \wedge \underbrace{at_{-l_{3,4}} + at_{-m_{3,4}} + y = 1}_{\varphi_2}$$
$$\rightarrow \underbrace{\neg(at_{-l_3} \wedge at_{-m_3})}_q$$

---

$$P \models \square \underbrace{\neg(at_{-l_3} \wedge at_{-m_3})}_q$$