



## $P$ -validity and $P$ -satisfiability problems (of $\varphi$ )

**$P$ -validity** problem (of  $\varphi$ )

**Main question:** given a finite-state program  $P$  and a formula  $\varphi$ , is  $\varphi$   $P$ -valid, that is, do all  $P$ -computations satisfy  $\varphi$ ?

**$P$ -satisfiability** problem (of  $\varphi$ )

**Main question:** given a finite-state program  $P$  and a formula  $\varphi$ , is there a  $P$ -computation which satisfies  $\varphi$ ?

To determine whether  $\varphi$  is  $P$ -valid, it suffices to employ an algorithm for deciding if there is a  $P$ -computation which satisfied  $\neg\varphi$ .

The algorithm for solving the  $P$ -satisfiability of  $\varphi$  makes use of the tableau for  $\varphi$   $T_\varphi$



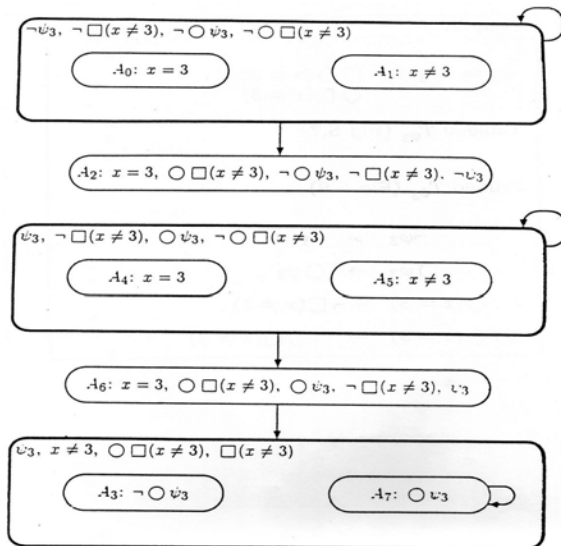






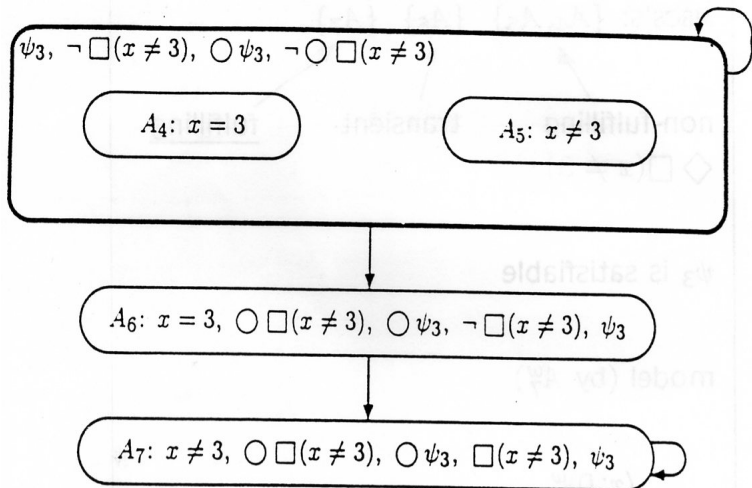


# The complete tableau

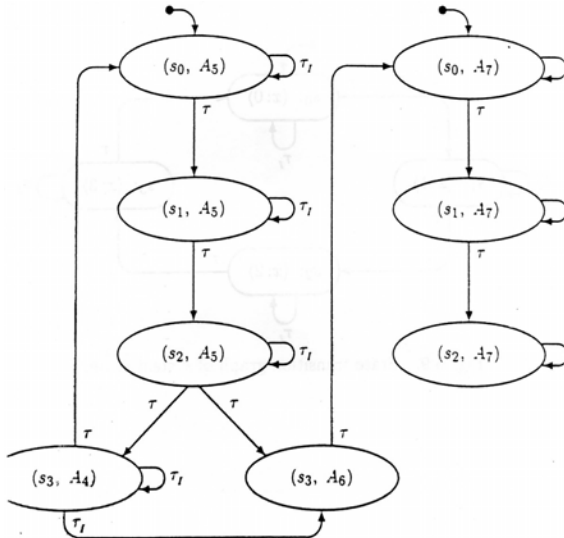




# The pruned tableau ( $T_{\psi_3}$ )



# The behavior graph $\mathcal{B}_{(LOOP, \psi_3)}$



# Paths in the behavior graph $\mathcal{B}_{(\mathcal{P}, \varphi)}$

## Proposition.

Let  $\varphi$  be an LTL formula.

The infinite sequence  $\pi : (s_0, A_0)(s_1, A_1) \dots$ , where  $(s_0, A_0)$  is an initial  $\varphi$ -node, is a path in  $\mathcal{B}_{(\mathcal{P}, \varphi)}$

if and only

- $\sigma_\pi : s_0 s_1 \dots$  is a run of  $P$  (computation less fairness)
- $\theta_\pi : A_0 A_1 \dots$  is a trail of  $T_\varphi$  over  $\sigma_\pi$  (for all  $j \geq 0$ ,  $A_j$  is consistent with  $s_j$ )

## Example.

In  $\mathcal{B}_{(LOOP, \psi_3)}$ , the path  $\pi : (s_0, A_5)(s_1, A_5)(s_2, A_5)(s_3, A_4))^\omega$  induces  $\sigma_\pi : (s_0 s_1 s_2 s_3)^\omega$  (run of LOOP) and  $\theta_\pi : (A_5 A_5 A_5 A_4)^\omega$  (trail of  $T_\varphi$  over  $\sigma_\pi$ )

## $P$ -satisfiability of $\varphi$ by path

### Proposition.

Let  $\varphi$  be an LTL formula.

There exists a  $P$ -computation which satisfies  $\varphi$   
if and only if

there is an infinite path  $\pi$  in  $\mathcal{B}_{(\mathcal{P}, \varphi)}$ , starting from an initial  $\varphi$ -node, such that

- $\sigma_\pi$  is a fair run (computation)
- $\theta_\pi$  is a fulfilling trail over  $\sigma_\pi$

### Example.

The trail  $\theta_\pi : (A_5 A_5 A_5 A_4)^\omega$  is not fulfilling (both atoms  $A_5$  and  $A_4$  include  $\Diamond \Box (x \neq 3)$  and  $\neg \Box (x \neq 3)$ ).

# Adequate subgraphs

Given a behavior graph  $\mathcal{B}_{(\mathcal{P}, \varphi)}$ ,

- node  $(s', A')$  is a  $\tau$ -**successor** of node  $(s, A)$  if  $\mathcal{B}_{(\mathcal{P}, \varphi)}$  contains a  $\tau$ -edge connecting  $(s, A)$  to  $(s', A')$
- transition  $\tau$  is **enabled** on node  $(s, A)$  if it is enabled on state  $s$

Given a subgraph  $S \subseteq \mathcal{B}_{(\mathcal{P}, \varphi)}$ ,

- transition  $\tau$  is **taken** in  $S$  if there exist two nodes  $(s, A)$  and  $(s', A')$  in  $S$  such that  $(s', A')$  is a  $\tau$ -successor of  $(s, A)$
- $S$  is **just** (resp., **compassionate**) if every just (resp., compassionate) transition  $\tau \in \mathcal{J}$  (resp.,  $\tau \in \mathcal{C}$ ) is either taken in  $S$  or is disabled on some nodes (resp., all nodes) in  $S$
- $S$  is **fair** if it is both just and compassionate
- $S$  is **fulfilling** if every promising formula is fulfilled by an atom  $A$  such that  $(s, A) \in S$  for some state  $s$
- $S$  is **adequate** if it is fair and fulfilling

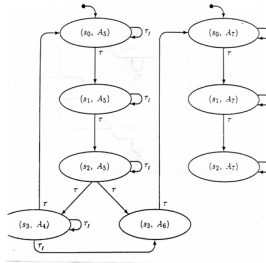
# Adequate strongly connected subgraphs and satisfiability

## Proposition.

A finite-state program  $P$  has a computation  $\sigma$  which satisfies  $\varphi$  if and only if the behavior graph  $\mathcal{B}_{(P, \varphi)}$  has an adequate strongly connected subgraph

**Example** (LOOP and  $\psi_3 : \Diamond \Box (x \neq 3)$ ).

The behavior graph  $\mathcal{B}_{(LOOP, \psi_3)}$  has no adequate subgraphs.



## Example (cont'd)

Let us check the maximal strongly connected subgraphs (MSCS):

- $\{(s_0, A_5), (s_1, A_5), (s_2, A_5), (s_3, A_4)\}$  is fair, but not fulfilling ( $\psi_3$  belongs to both  $A_4$  and  $A_5$ , and it promises  $\Box(x \neq 3)$ , but  $\Box(x \neq 3) \notin A_4, A_5$ )
- $\{(s_0, A_7)\}$ ,  $\{(s_1, A_7)\}$ , and  $\{(s_2, A_7)\}$  are fulfilling, but not fair (they are not just with respect to transition  $\tau$ )
- $\{(s_3, A_6)\}$  is neither fair (it is not just with respect to  $\tau$ ) nor fulfilling (it is transient)

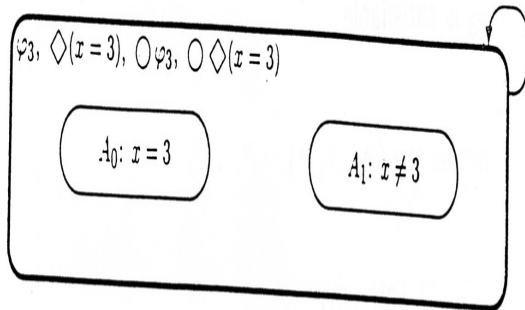
Hence, there are **no adequate subgraphs** in  $\mathcal{B}_{(LOOP, \psi_3)}$ .

By the last proposition, it follows that LOOP has no computation that satisfies  $\psi_3 : \Diamond\Box(x \neq 3)$

## Another example

**Example** (LOOP and  $\phi_3 (= \neg \psi_3) : \Box \Diamond (x = 3)$ ).

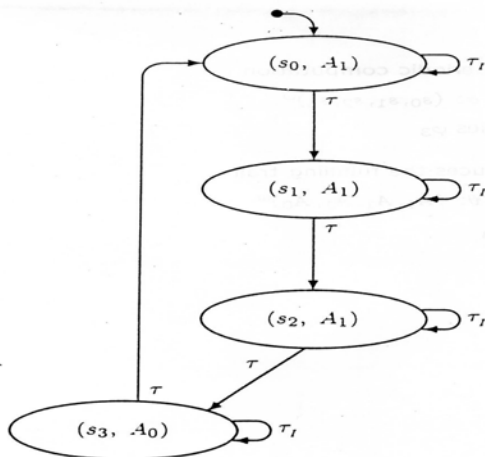
The pruned tableau is the following one:





## Another example (cont'd)

The behavior graph  $\mathcal{B}_{(LOOP, \phi_3)}$  is the following one:



## Another example (cont'd)

The subgraph  $S = \{(s_0, A_1), (s_1, A_1), (s_2, A_1), (s_3, A_0)\}$  is an **adequate subgraph**, as it is both fair ( $\tau$  is taken in  $S$ ) and fulfilling ( $\Diamond(x = 3)$  belongs to both  $A_0$  and  $A_1$ , but  $x = 3$  belongs to  $A_0$ )

By the last proposition, it follows that LOOP has computation that satisfies  $\phi_3 : \Box\Diamond(x = 3)$

The periodic computation  $\sigma : (s_0s_1s_2s_3)^\omega$  satisfies  $\phi$ .

It induces the fulfilling trail  $\theta : (A_1A_1A_1A_0)^\omega$  in  $T_\phi$ .

## How to find adequate subgraphs?

Checking MSCS is **not enough**:

$$S' \subset S$$

$S'$  just **implies**  $S$  just

$S'$  fulfilled **implies**  $S$  fulfilled

but

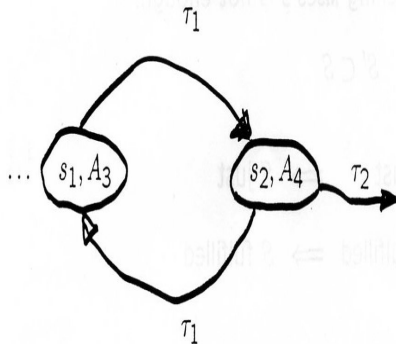
$S'$  compassionate **do not imply**  $S$  compassionate

Therefore, it is possible that  $S$  is not adequate, but  $S'$  is adequate

# A counterexample

$S'$  compassionate **do not imply**  $S$  compassionate

Let  $\tau_2$  belong to the set of compassionate transitions.  $\tau_2$  is enabled on  $s_2$  and disabled on  $s_1$



## A counterexample (cont'd)

The strongly connected subgraph  $S' = \{(s_1, A_3)\}$  is compassionate ( $\tau_2$  is disabled on all the states in this subgraph)

The strongly connected subgraph  $S = \{(s_1, A_3)(s_2, A_4)\}$ , that includes  $S'$ , is not compassionate ( $\tau_2$  is enabled on  $(s_2, A_4)$ , but it is not taken in  $S$ )

# Algorithm ADEQUATE-SUB

## Algorithm **ADEQUATE-SUB**

- accepts as input a strongly connected subgraph  $S$  and returns as output a strongly connected subgraph  $S' \subseteq S$

If  $S' = \emptyset$  -  $S$  contains no adequate subgraphs

Otherwise -  $S'$  is an adequate strongly connected subgraph in  $S$

- Notation:

$EN(\tau, S)$  - the set of all nodes  $(s, A)$  in  $S$  on which  $\tau$  is enabled

## Algorithm ADEQUATE-SUB (cont'd)

**Algorithm** ADEQUATE-SUB checks for adequate subgraphs

**recursive function** adequate-sub( $S$ : SCS) **returns** SCS

**if**  $S$  is not fulfilling **then return**  $\emptyset$  – failure

**if**  $S$  is not just **then return**  $\emptyset$  – failure

**if**  $S$  is compassionate **then return**  $S$  – success

–  $S$  is fulfilling and just but not compassionate. Let  $T \subseteq \mathcal{C}$

– be the set of all compassionate transitions that are not taken

– in  $S$ . Clearly,  $EN(T, S) \neq \emptyset$ .

**let**  $U = S - EN(T, S)$ . Decompose  $U$  into MSCSs  $U_1 \dots U_k$ .

**let**  $V = \emptyset, i = 1$

**while**  $V = \emptyset$  and  $i \leq k$  **do**

**let**  $V = \text{adequate} - \text{sub}(U_i); i := i + 1$

**end-while**

**return**  $V$

## An example: the system LOOP+

The system **LOOP+**

Initially  $x = 0$

LTL formula  $\psi_3 : \Diamond \Box (x \neq 3)$

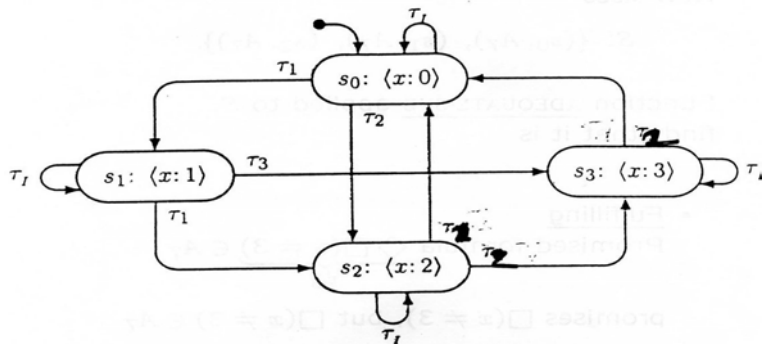
Transitions: (i) the idling transition  $\tau_I$ ; (ii)  $\mathcal{J} = \{\tau_1, \tau_2\}$ ; (iii)  
 $\mathcal{C} = \{\tau_3\}$

New MSCS:  $S = \{(s_0, A_7), (s_1, A_7), (s_2, A_7)\}$

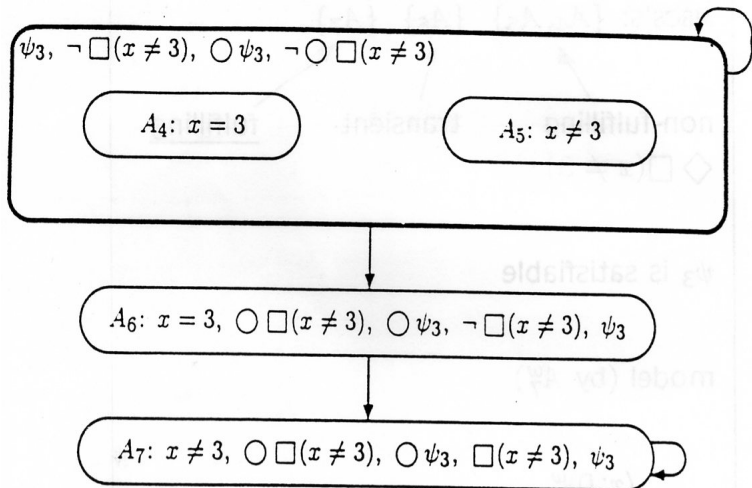
In the next transparencies, we respectively provide the state-transition graph  $G_{\text{LOOP+}}$ , the pruned tableau  $T_{\psi_3}$ , and the behavior graph  $\mathcal{B}_{(\text{LOOP+}, \psi_3)}$ .



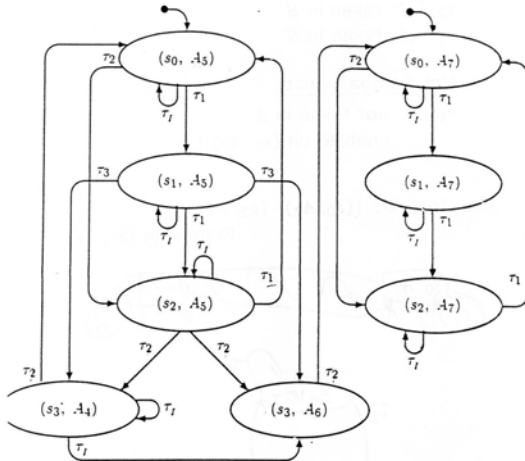
# The state-transition graph of system LOOP+ ( $G_{LOOP+}$ )



# The pruned tableau ( $T_\psi$ )



# The behavior graph $\mathcal{B}_{(\mathcal{L}OOP+, \psi)}$



# Application of the function ADEQUATE-SUB

Function **ADEQUATE-SUB** applied to  $S$  finds that it is

- **fulfilling**: the formula  $\psi_3 : \Diamond \Box (x \neq 3)$ , that promises  $\Box (x \neq 3)$ , belongs to  $A_7$ , but  $\Box (x \neq 3)$  belongs to  $A_7$  as well
- **just**:  $\tau_2 \in \mathcal{J}$  is taken in  $S$  and  $\tau_1 \in \mathcal{J}$  is taken in  $S$
- **not compassionate**:  $\tau_3 \in \mathcal{C}$  is not taken in  $S$ , but it is enabled on  $(s_1, A_7)$

Construct  $U : \{(s_0, A_7), (s_2, A_7)\}$  by removing  $(s_1, A_7)$











