Model Checking I
alias
Reactive Systems Verification

Luca Tesei

MSc in Computer Science, University of Camerino

Topics

- Parallelism
- Interleaving operator for Program Graphs
- Mutual Exclusion
- Peterson Algorithm for Mutual Exclusion

Material

Reading:

Chapter 2 of the book, pages 39–47.

More:

The slides in the following pages are taken from the material of the course "Introduction to Model Checking" held by Prof. Dr. Ir. Joost-Pieter Katoen at Aachen University.

Interleaving for program graphs

Interleaving for program graphs

. . . for modeling parallel systems with
subprocesses communicating via shared variables

Interleaving for program graphs

program graph $\mathcal{P}_1$
$(Loc_1, \ldots, \hookrightarrow_1, \ldots)$

program graph $\mathcal{P}_2$
$(Loc_2, \ldots, \hookrightarrow_2, \ldots)$

interleaving operator

$$\mathcal{P}_1 \,|||\, \mathcal{P}_2 \;=\; (Loc_1 \times Loc_2, \ldots, \hookrightarrow, \ldots)$$

Interleaving for program graphs

program graph $\mathcal{P}_1$
$(Loc_1, \ldots, \hookrightarrow_1, \ldots)$

program graph $\mathcal{P}_2$
$(Loc_2, \ldots, \hookrightarrow_2, \ldots)$

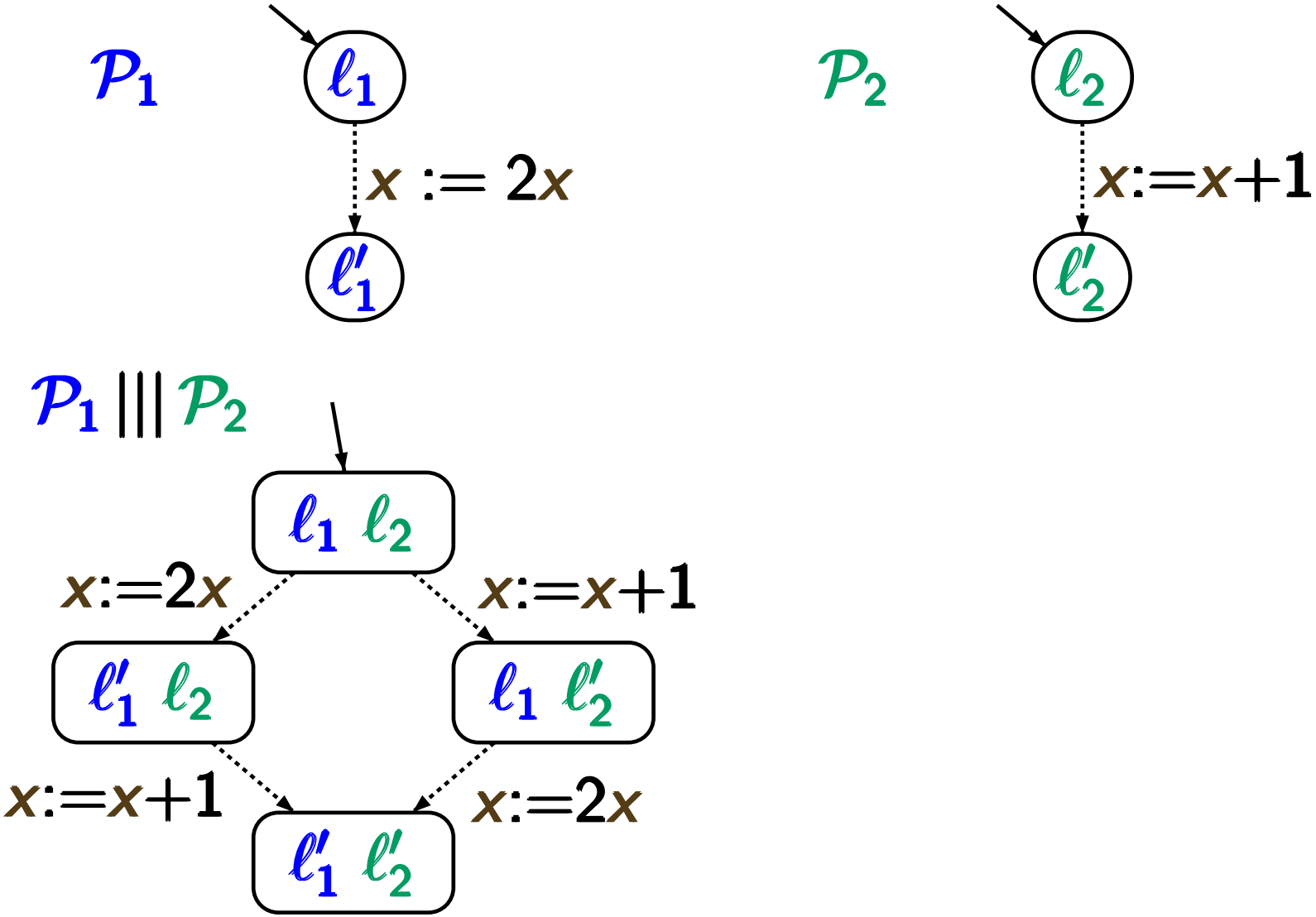
interleaving operator

$$\mathcal{P}_1 \,|||\, \mathcal{P}_2 \;=\; (Loc_1 \times Loc_2, \ldots, \hookrightarrow, \ldots)$$

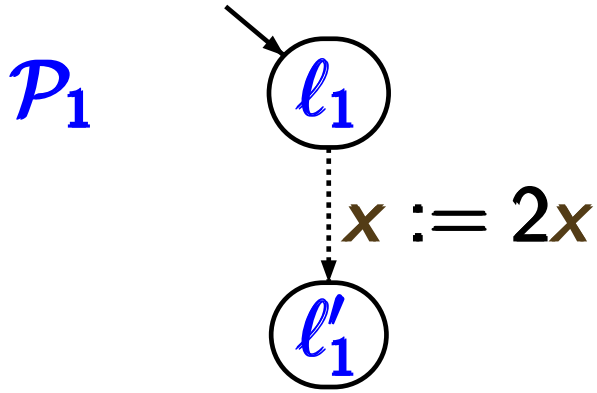
$$\frac{\ell_1 \overset{g\,:\,\alpha}{\hookrightarrow_1} \ell_1'}{\langle \ell_1, \ell_2 \rangle \overset{g\,:\,\alpha}{\hookrightarrow} \langle \ell_1', \ell_2 \rangle} \qquad \frac{\ell_2 \overset{g\,:\,\alpha}{\hookrightarrow_2} \ell_2'}{\langle \ell_1, \ell_2 \rangle \overset{g\,:\,\alpha}{\hookrightarrow} \langle \ell_1, \ell_2' \rangle}$$

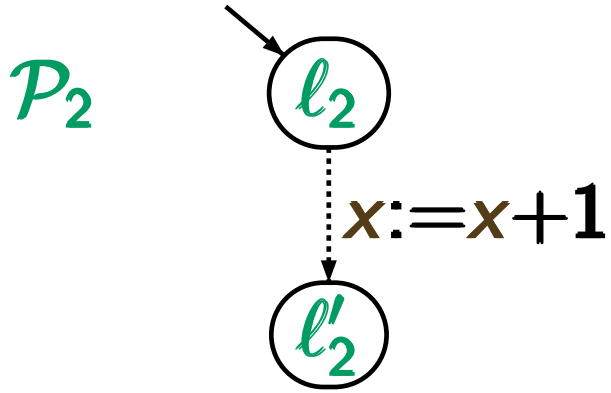
Example: interleaving for PG

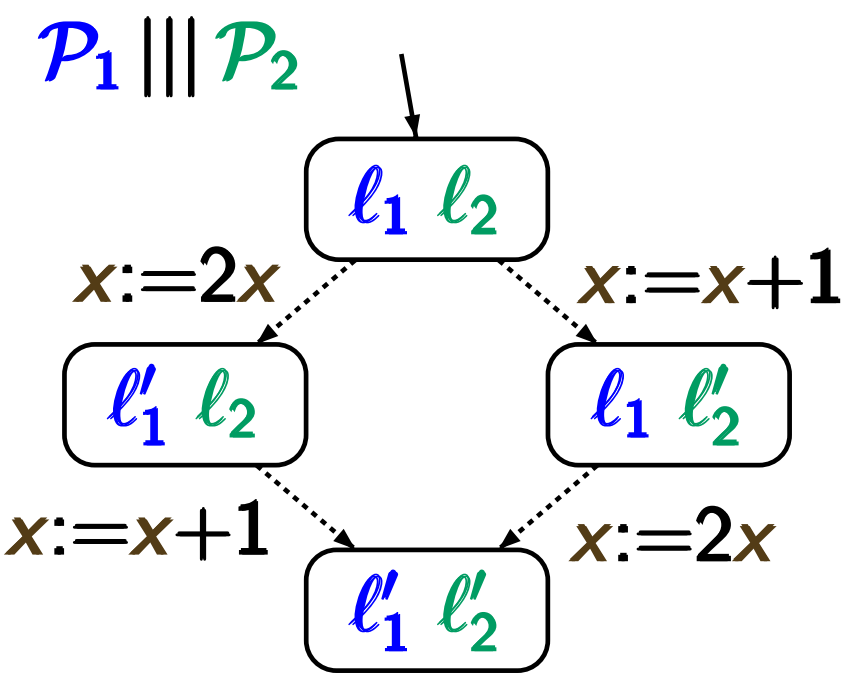
Example: interleaving for PG

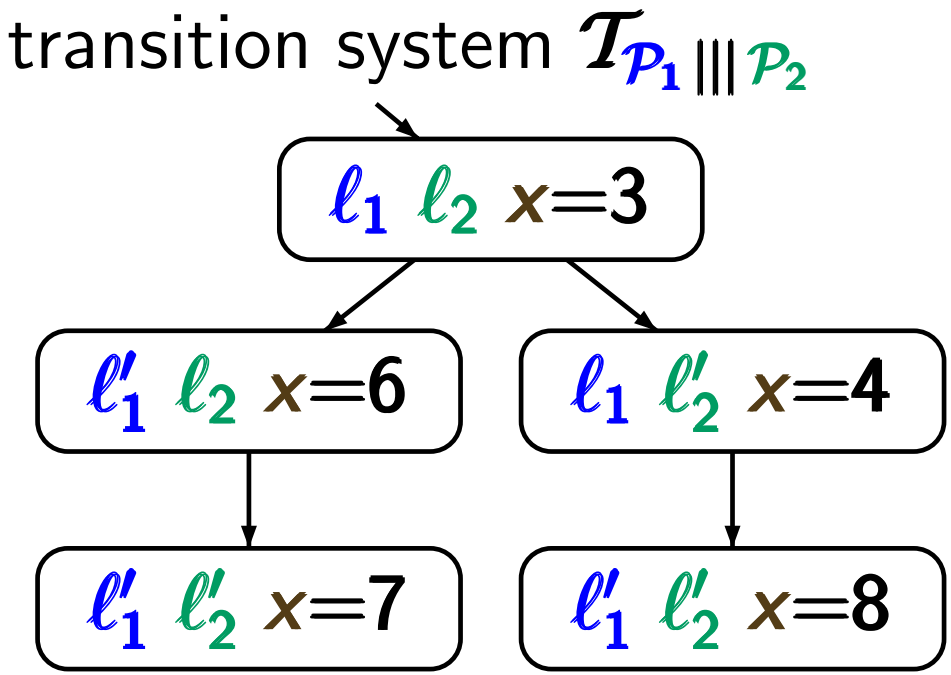


Example: interleaving for PG

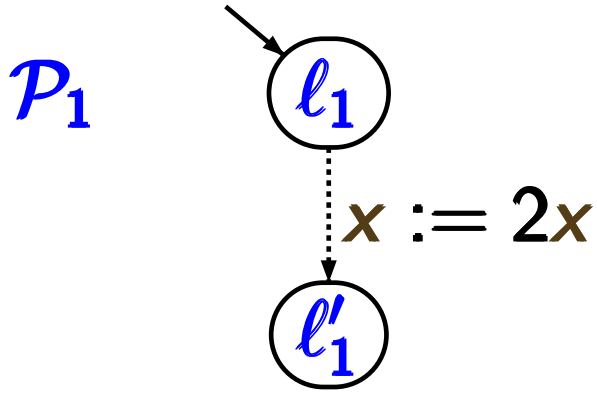


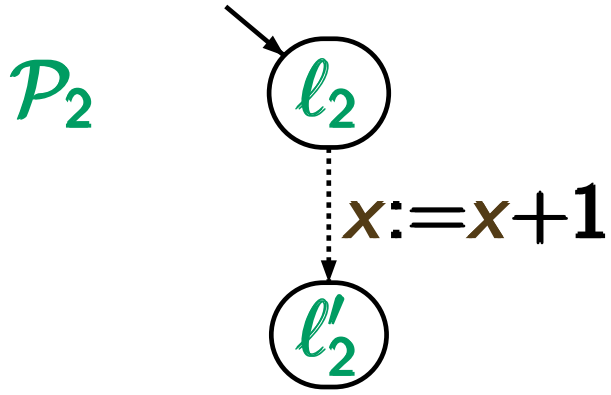


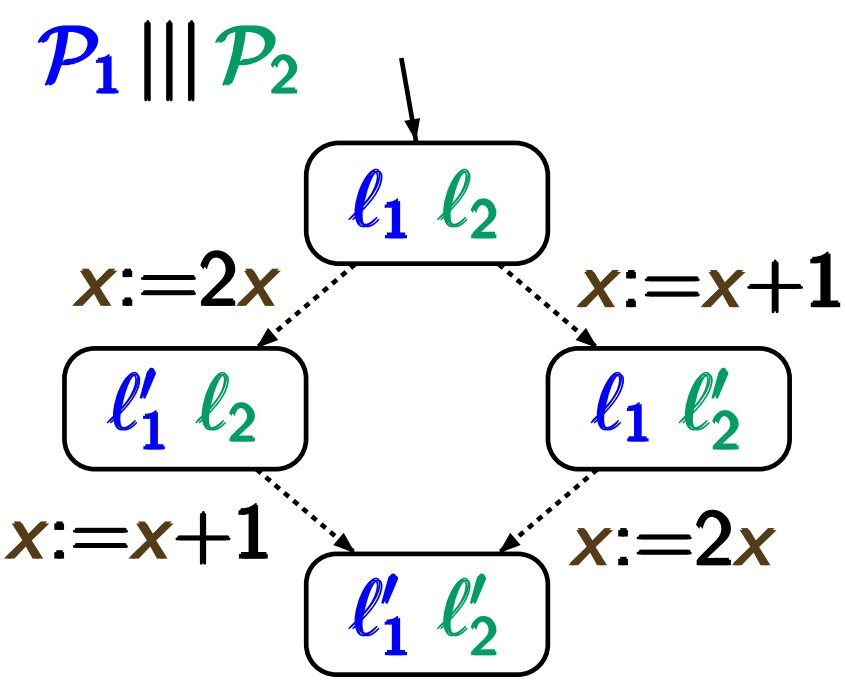


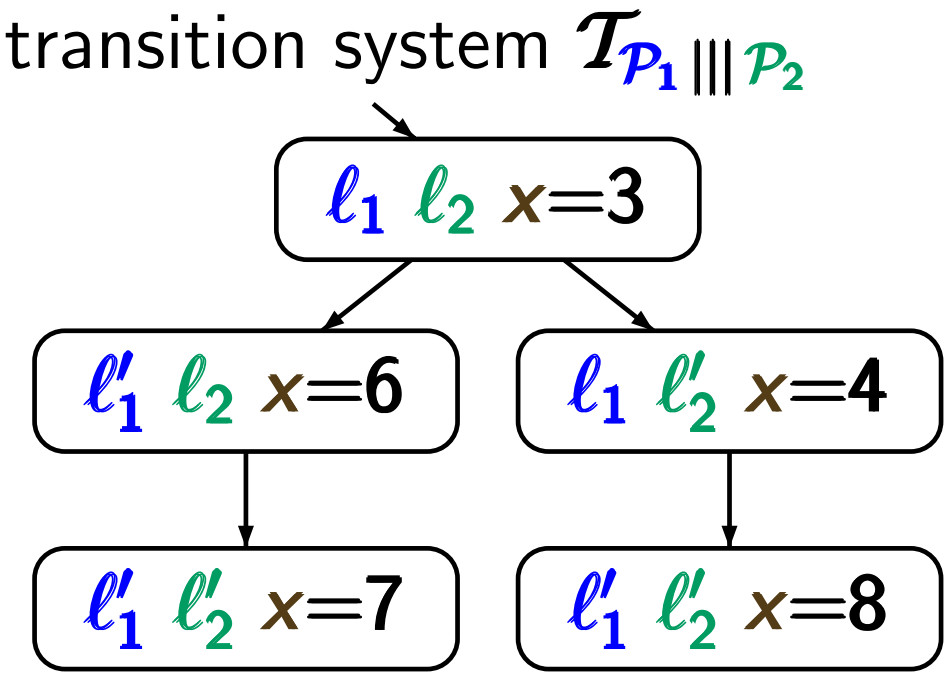


Example: interleaving for PG



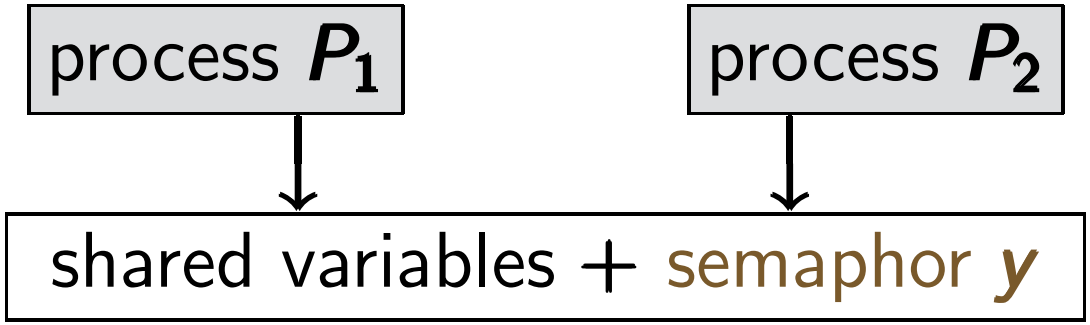




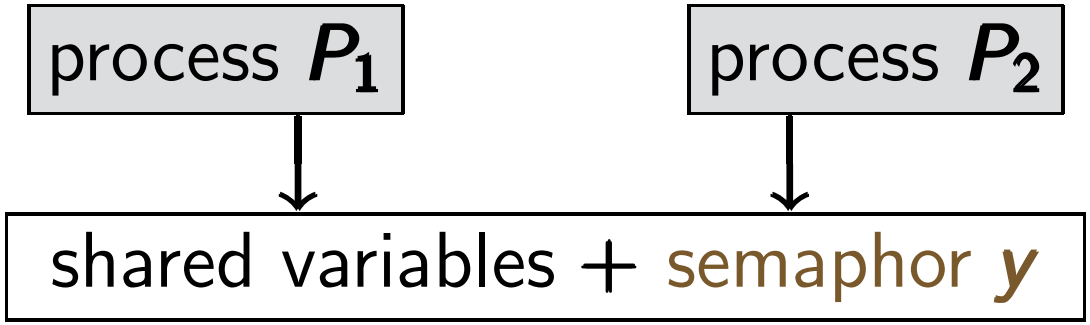


note: $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2} \neq \mathcal{T}_{\mathcal{P}_1} ||| \mathcal{T}_{\mathcal{P}_2}$

Mutual exclusion with semaphore



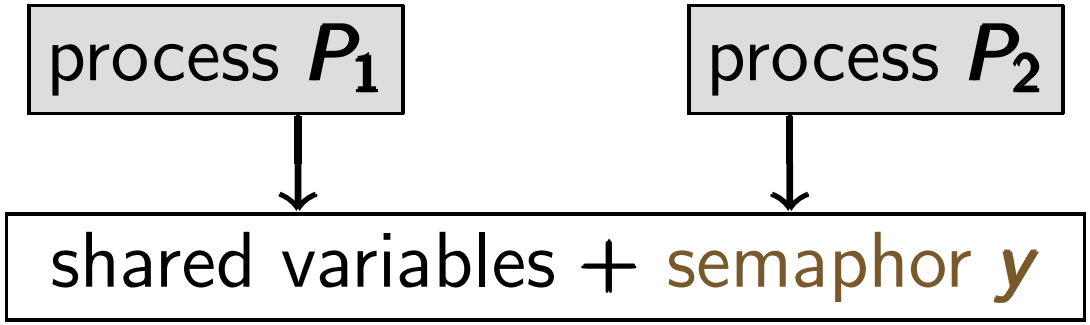
Mutual exclusion with semaphore



protocol for process P_i

```
LOOP FOREVER
  noncritical actions;
  AWAIT y > 0 DO
        y:=y−1
  OD
  critical actions;
  y:=y+1
END LOOP
```

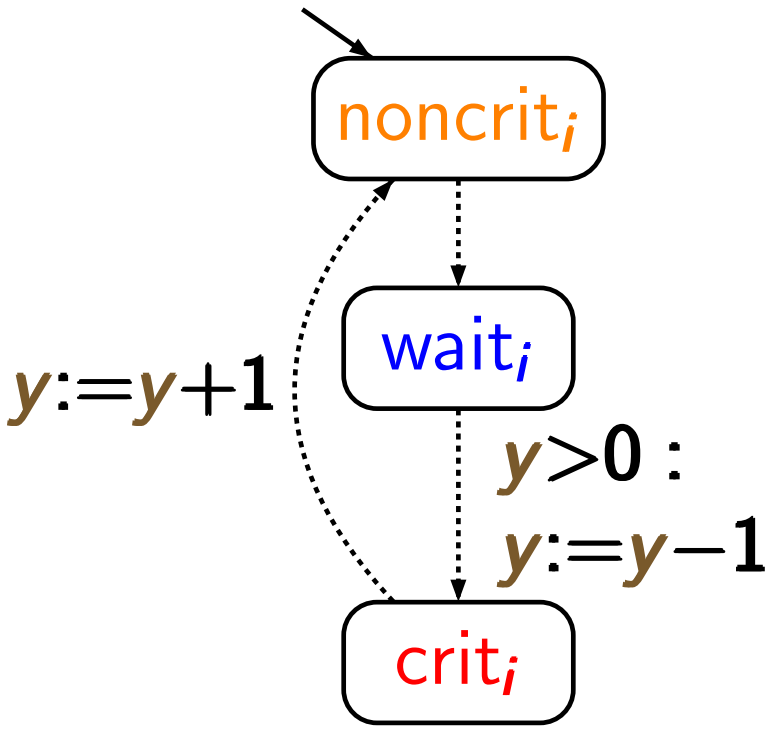
Mutual exclusion with semaphore



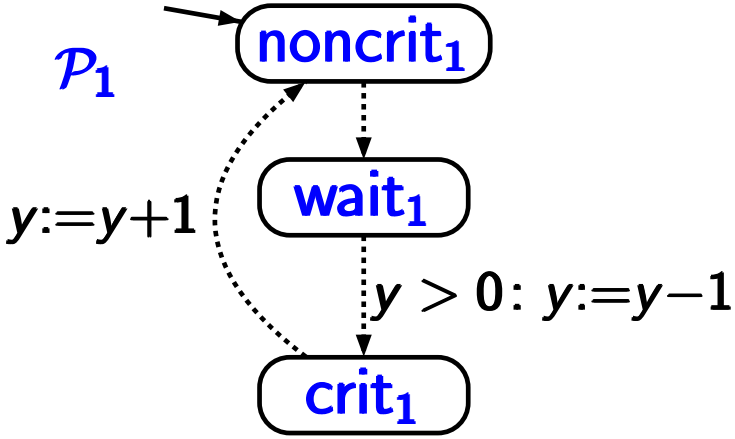
protocol for process P_i

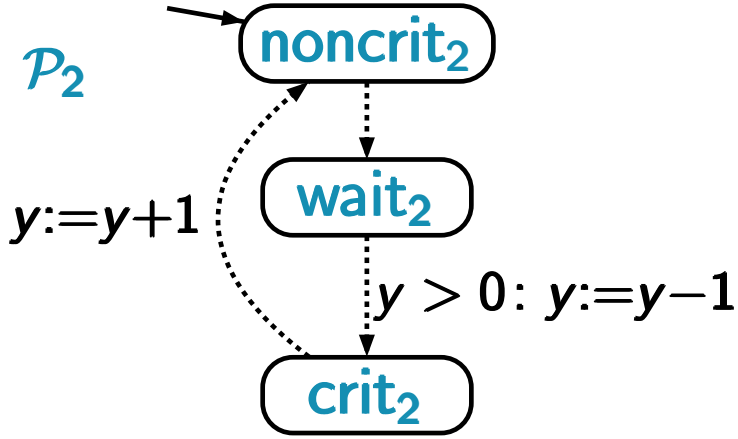
```
LOOP FOREVER
  noncritical actions;
  AWAIT y > 0 DO
        y:=y−1
  OD
  critical actions;
  y:=y+1
END LOOP
```

program graph $\mathcal{P}_i$

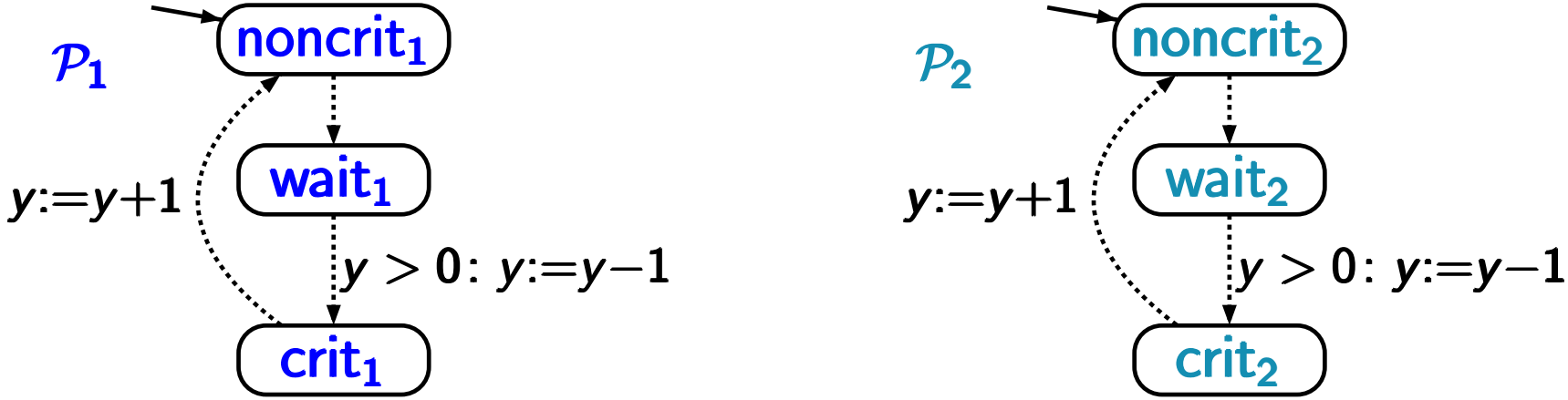


Mutual exclusion with semaphore



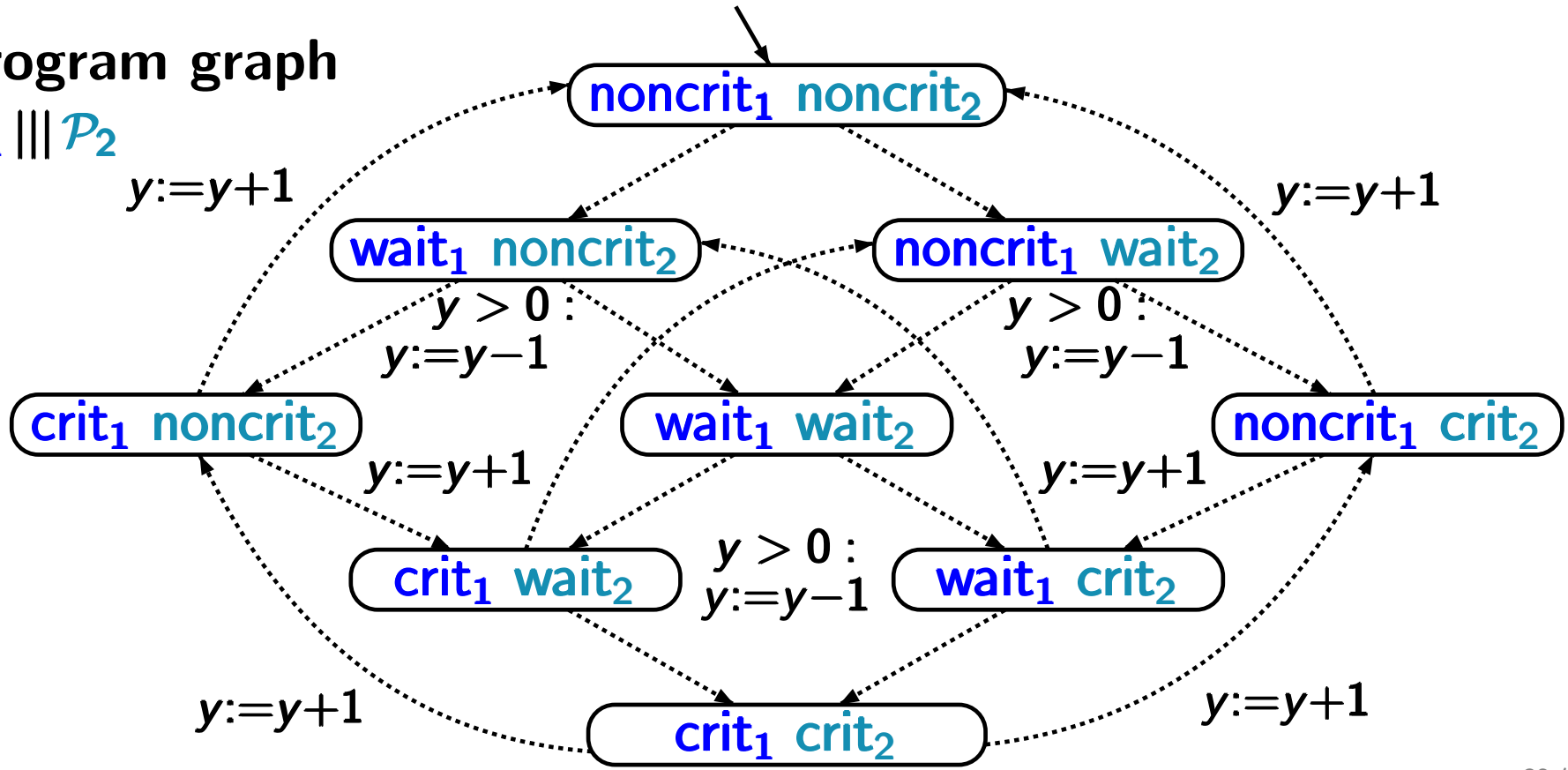


Mutual exclusion with semaphore

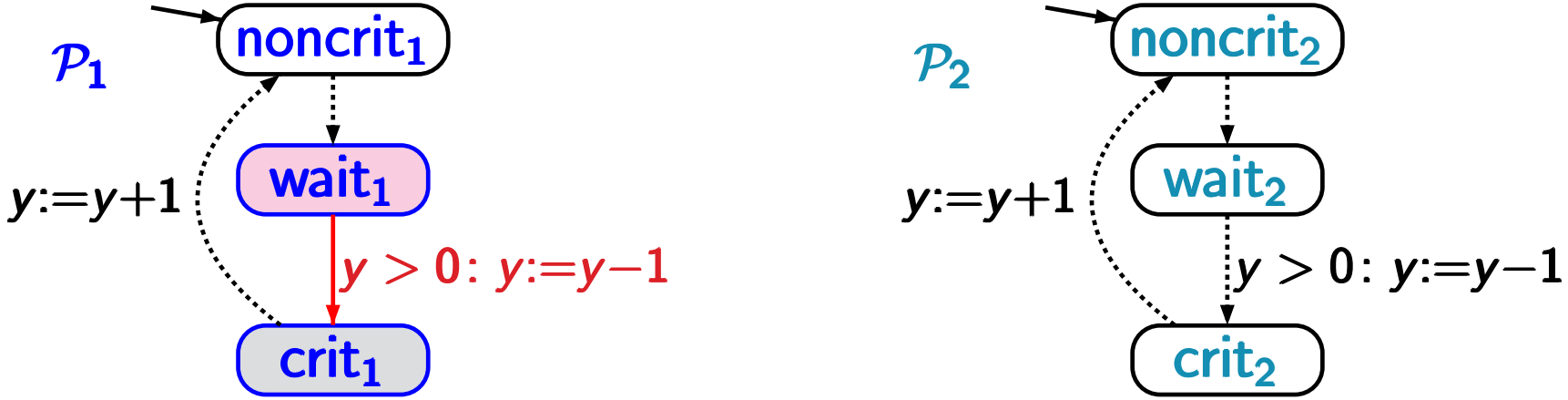


program graph

$\mathcal{P}_1 \,|||\, \mathcal{P}_2$

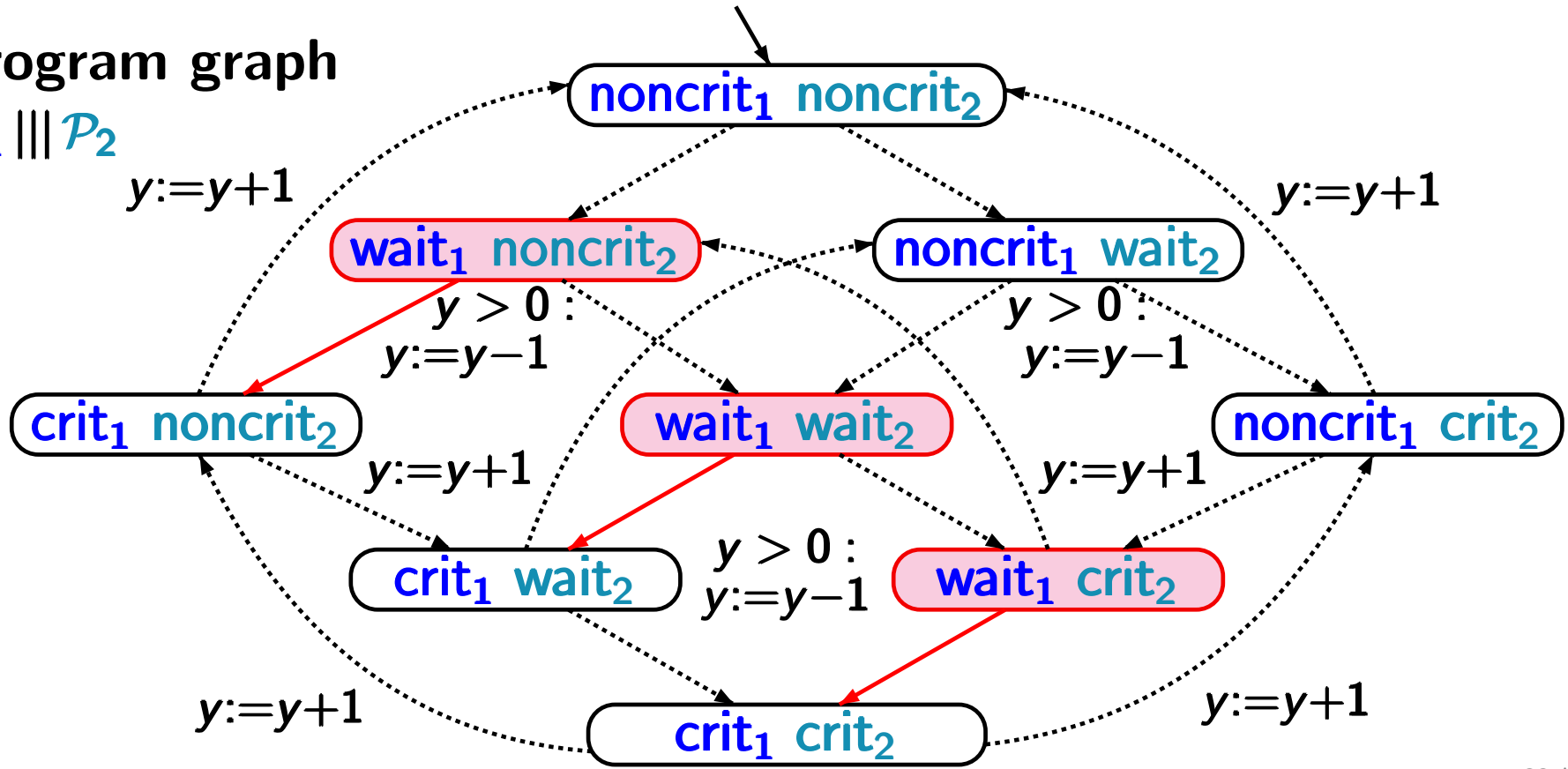


Mutual exclusion with semaphore

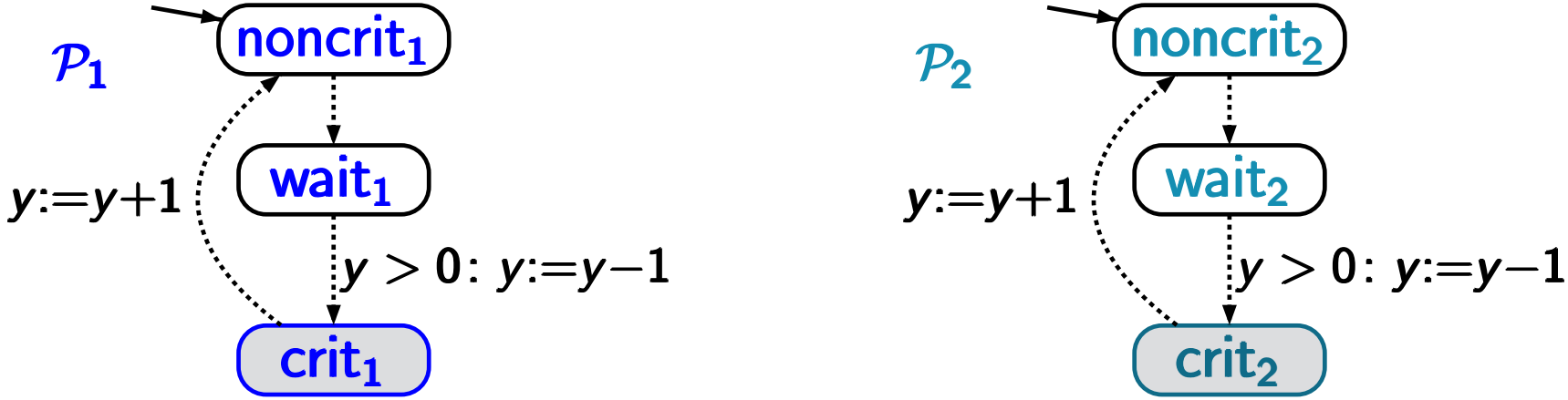


program graph

$\mathcal{P}_1 \,|||\, \mathcal{P}_2$

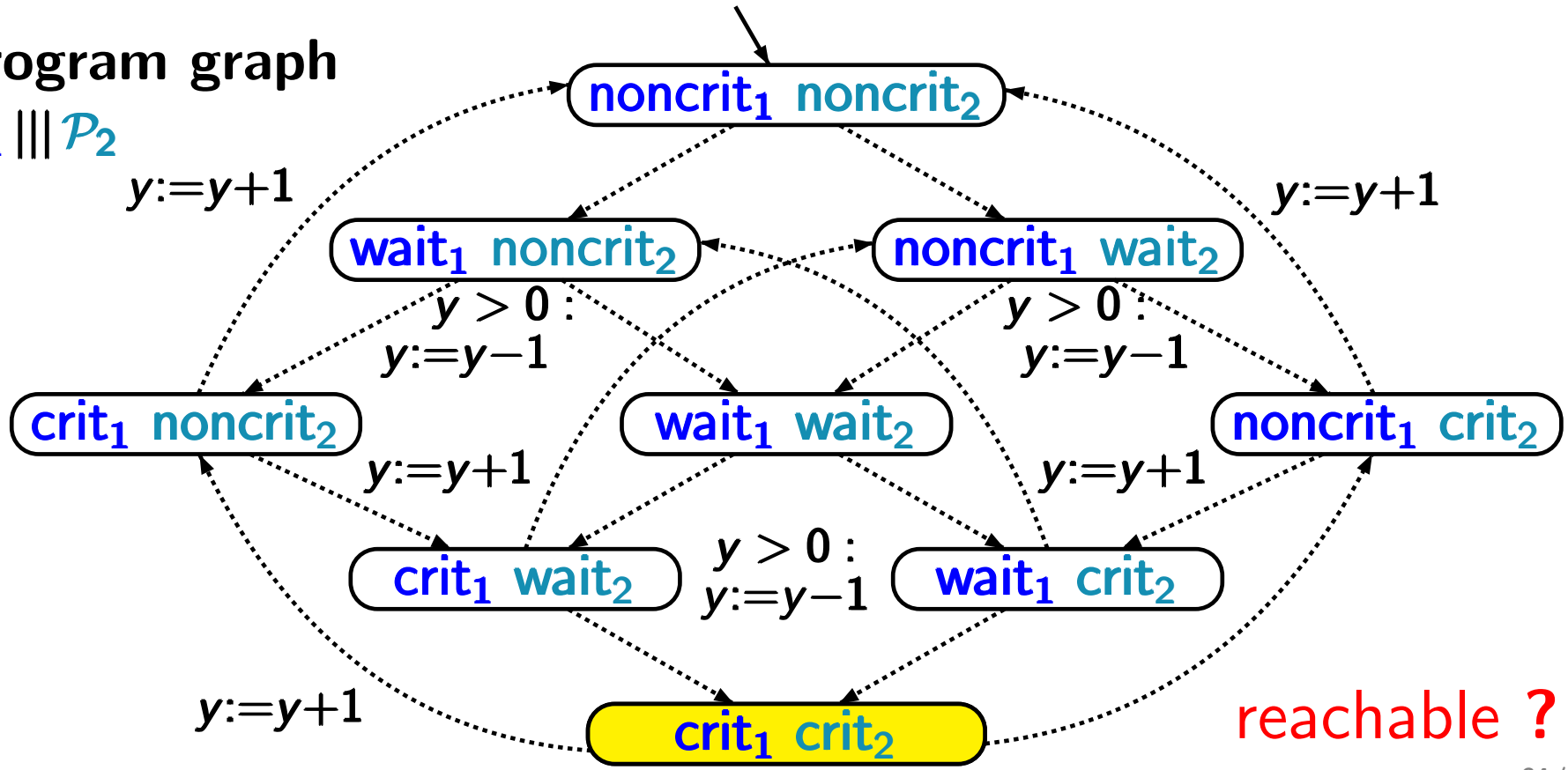


Mutual exclusion with semaphore



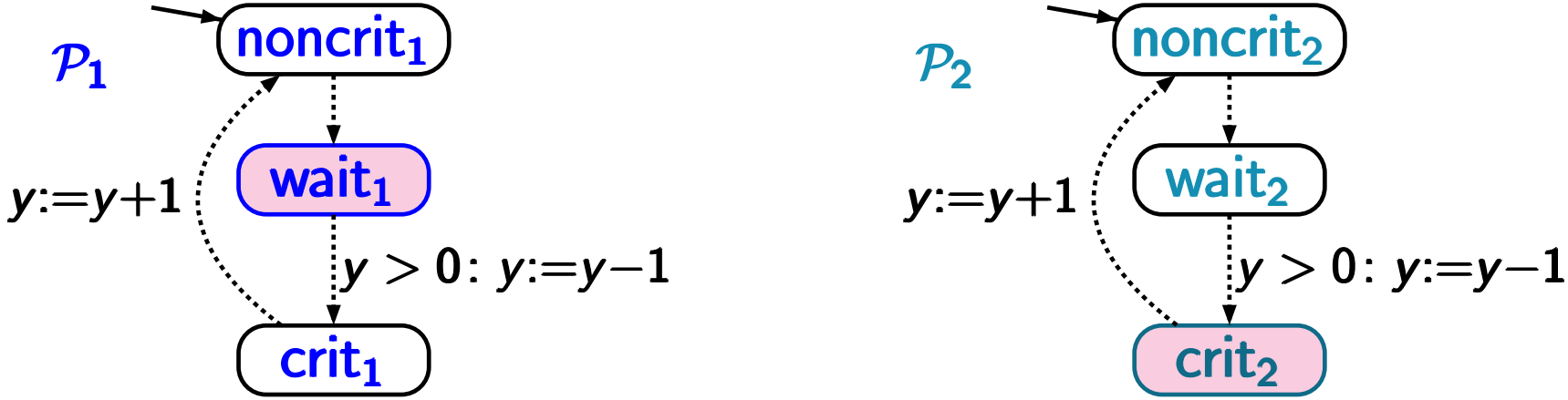
program graph

$\mathcal{P}_1 \,|||\, \mathcal{P}_2$



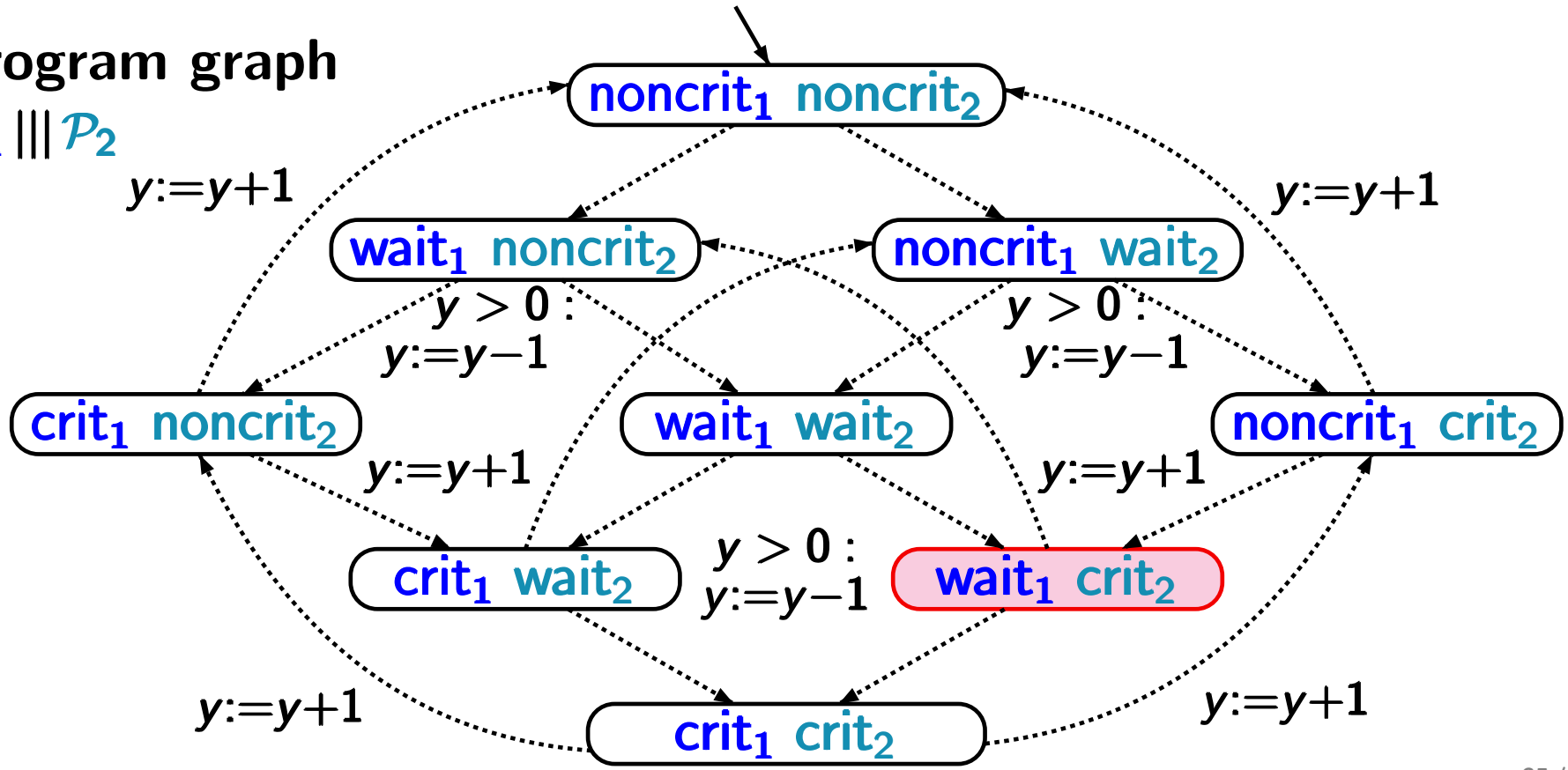
reachable ?

Mutual exclusion with semaphore

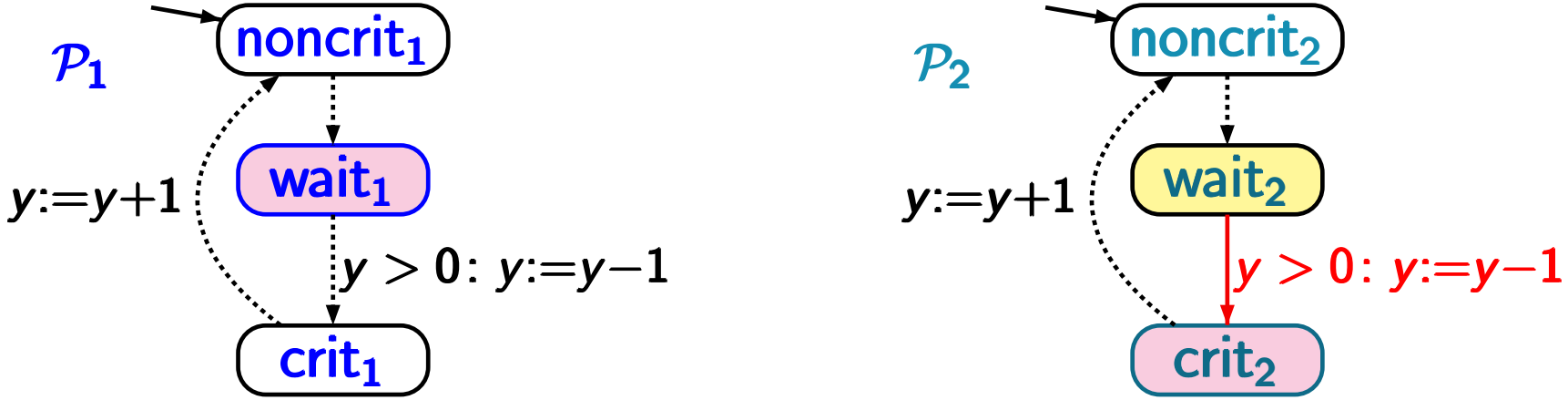


program graph

$\mathcal{P}_1 \,|||\, \mathcal{P}_2$

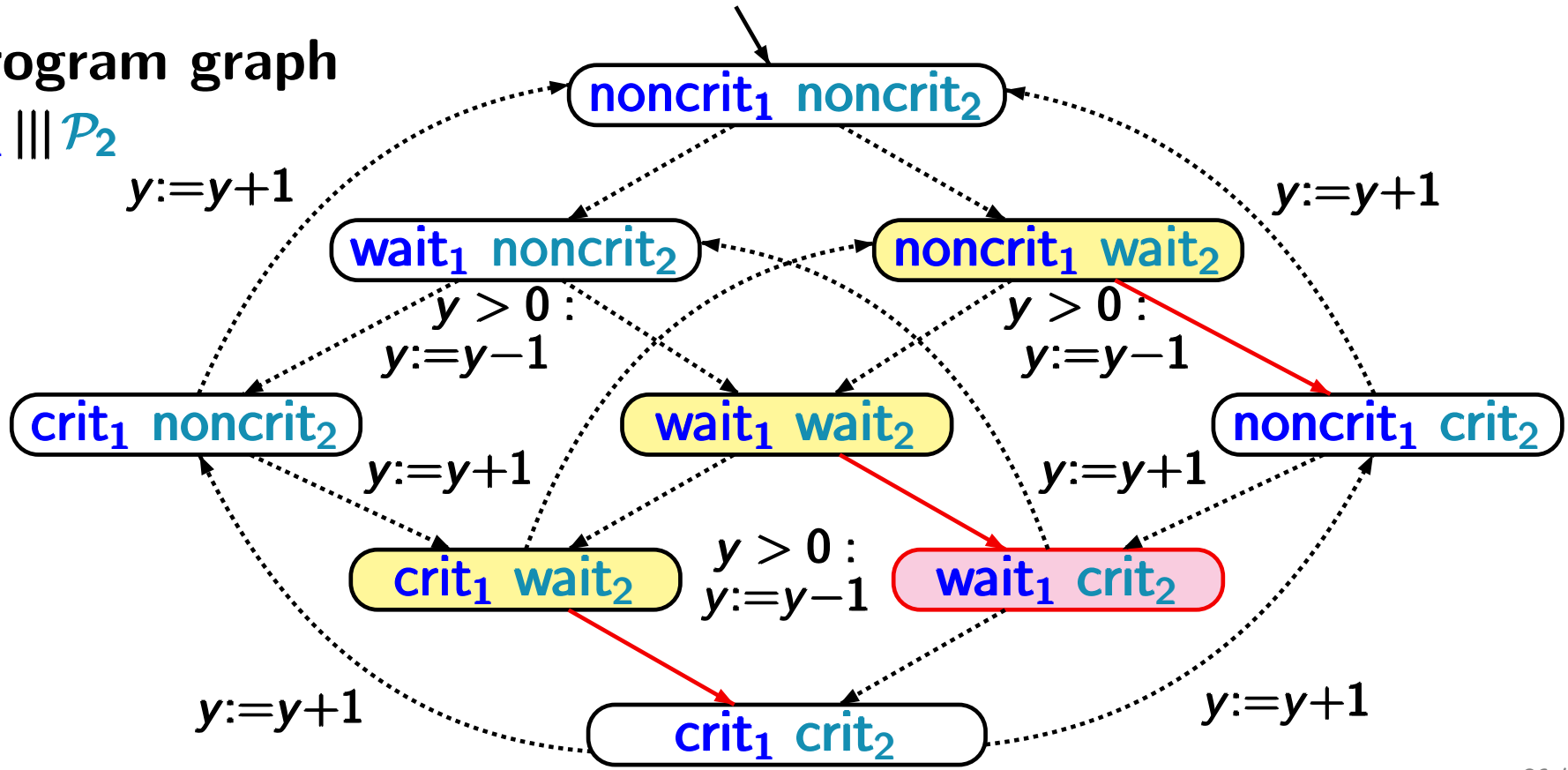


Mutual exclusion with semaphore

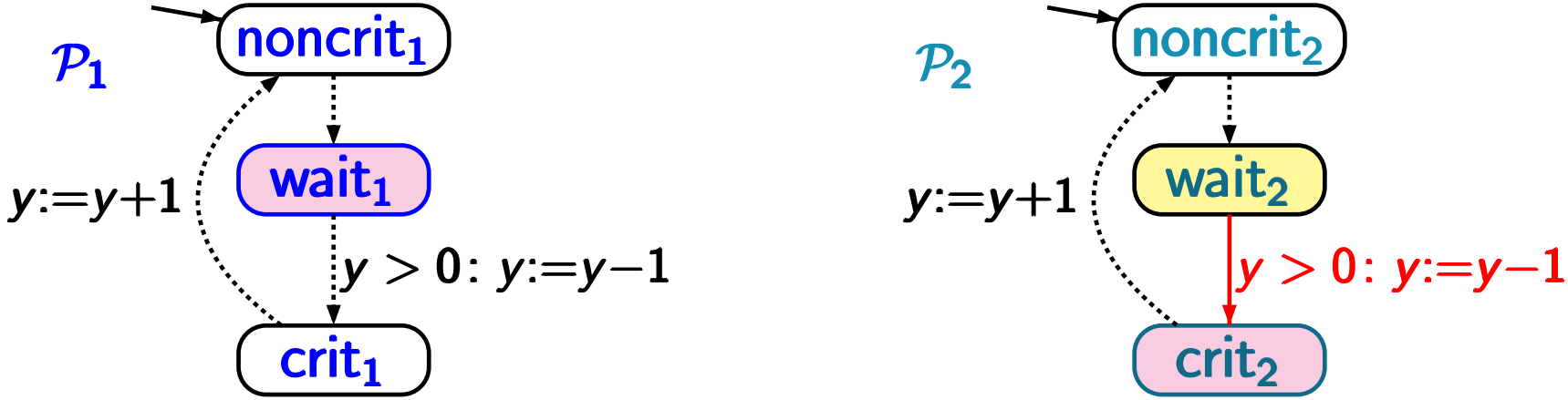


program graph

$\mathcal{P}_1 \,|||\, \mathcal{P}_2$

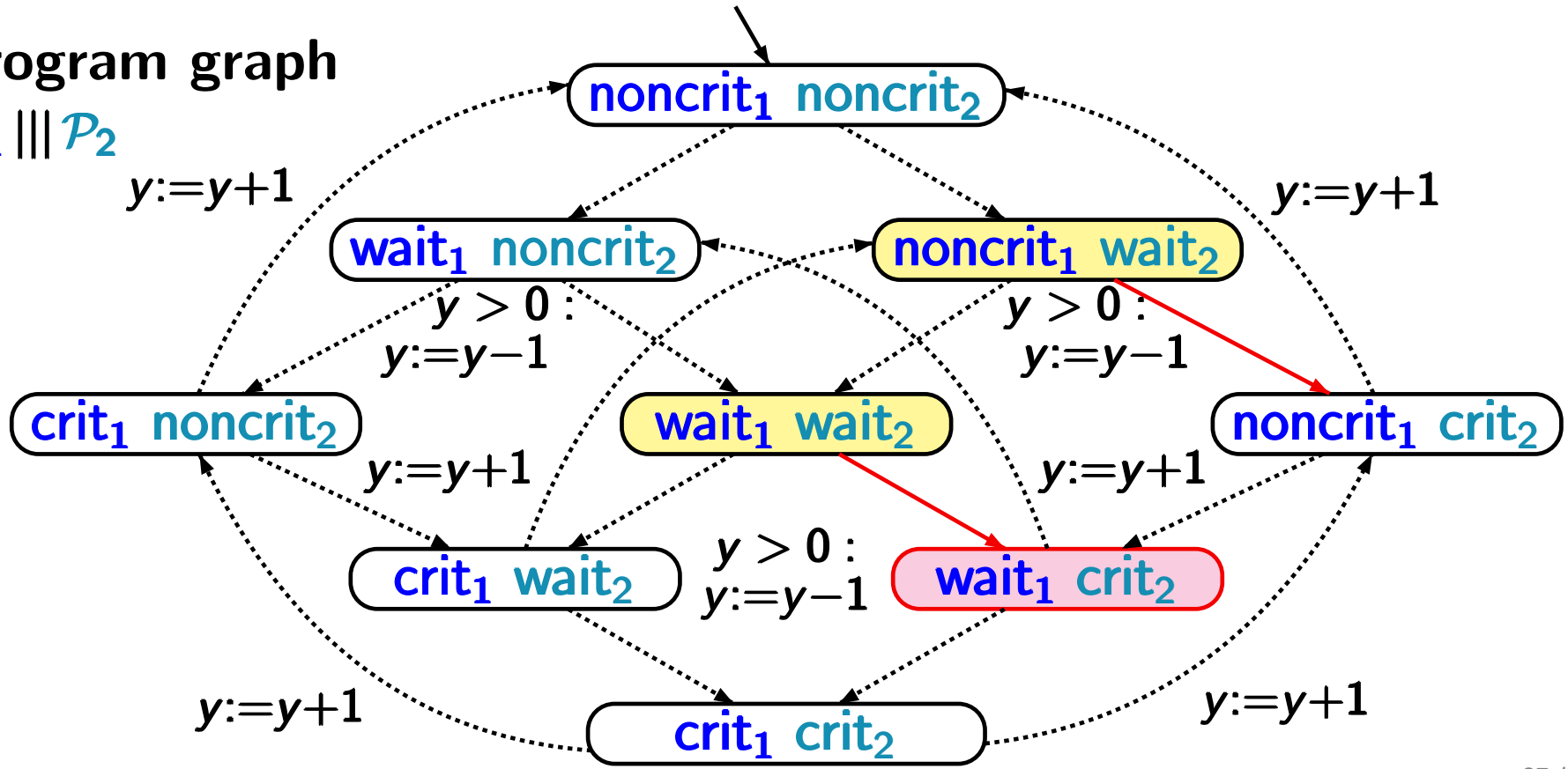


Mutual exclusion with semaphore

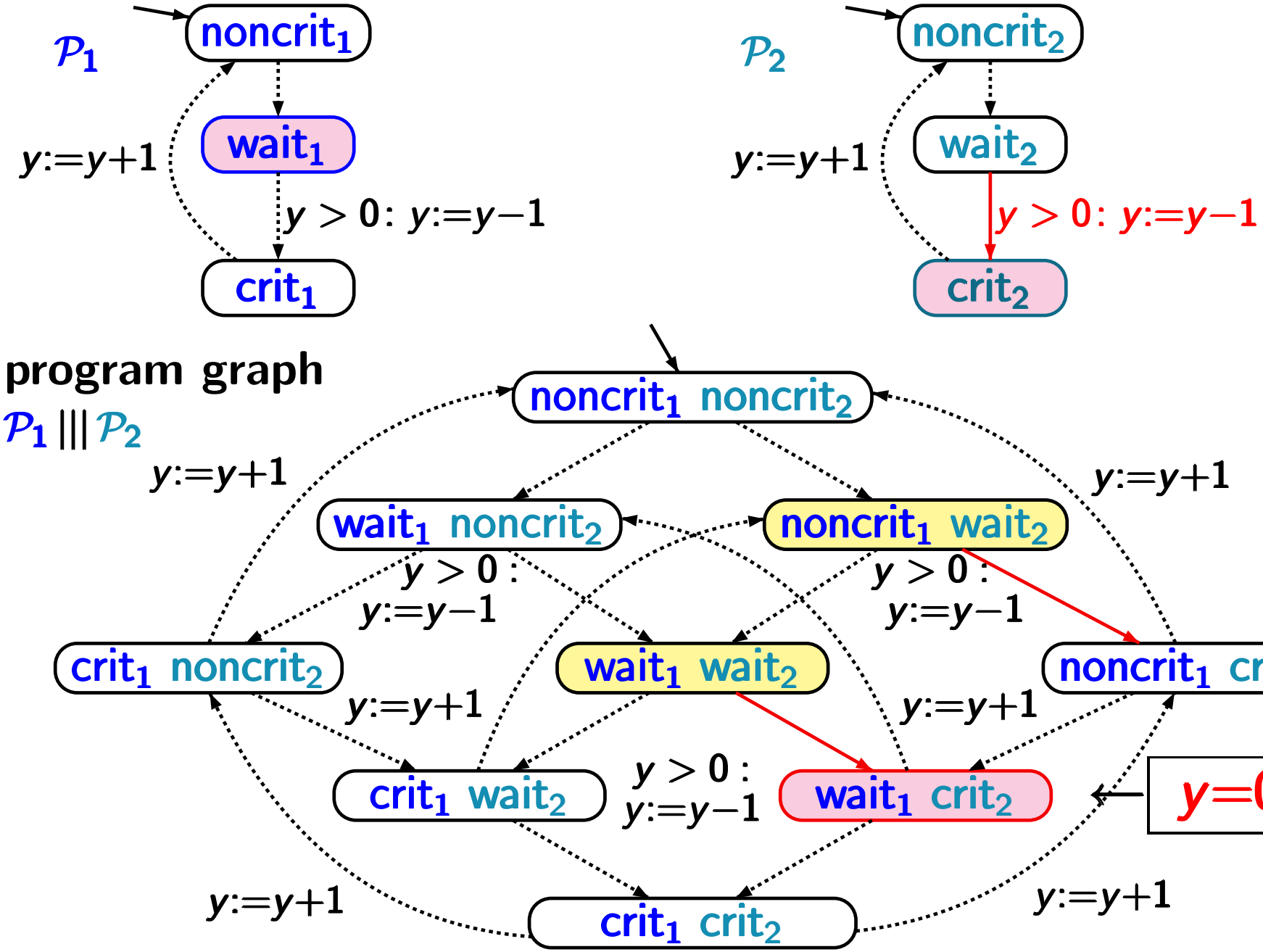


program graph

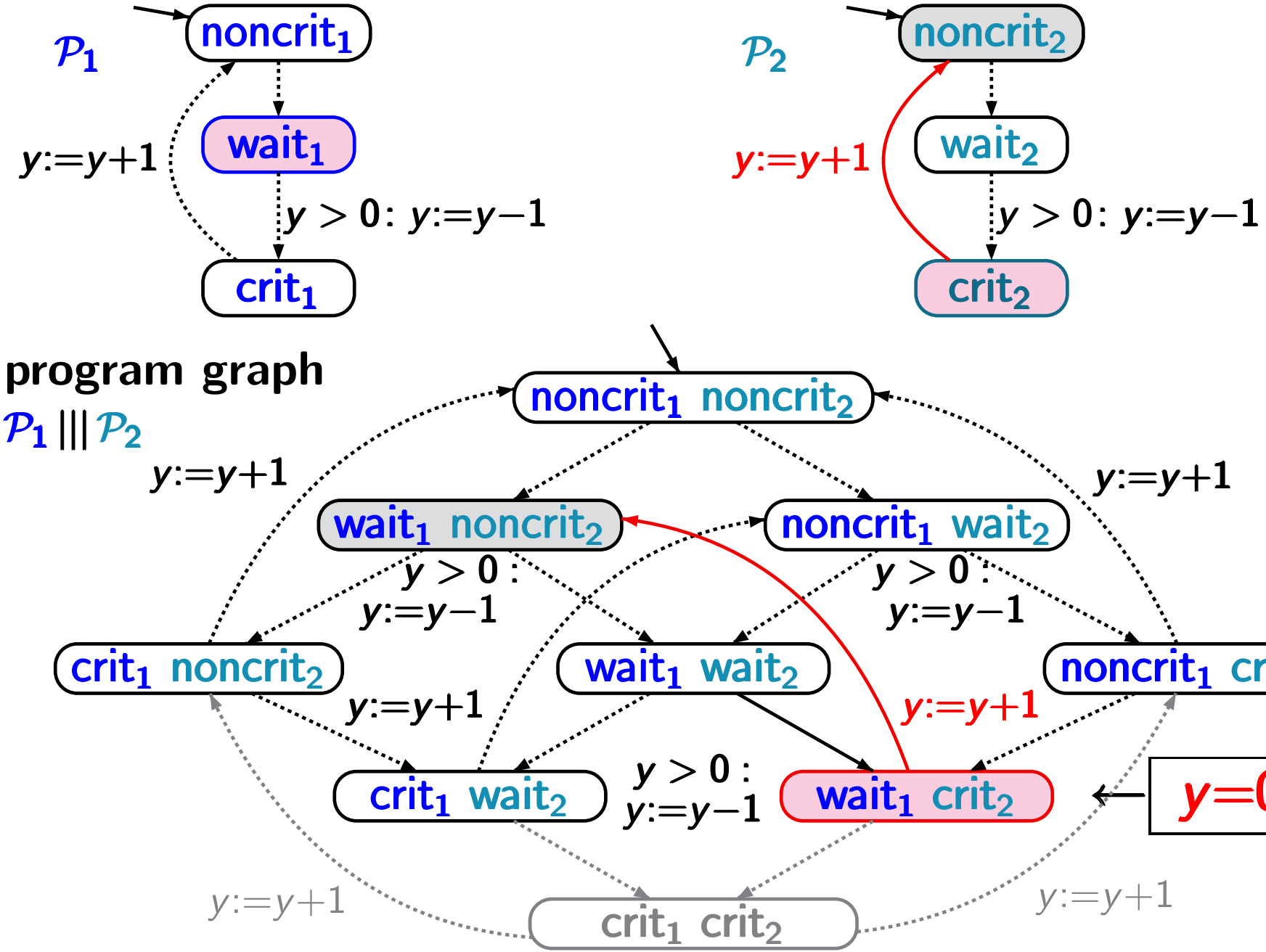
$\mathcal{P}_1 \,|||\, \mathcal{P}_2$



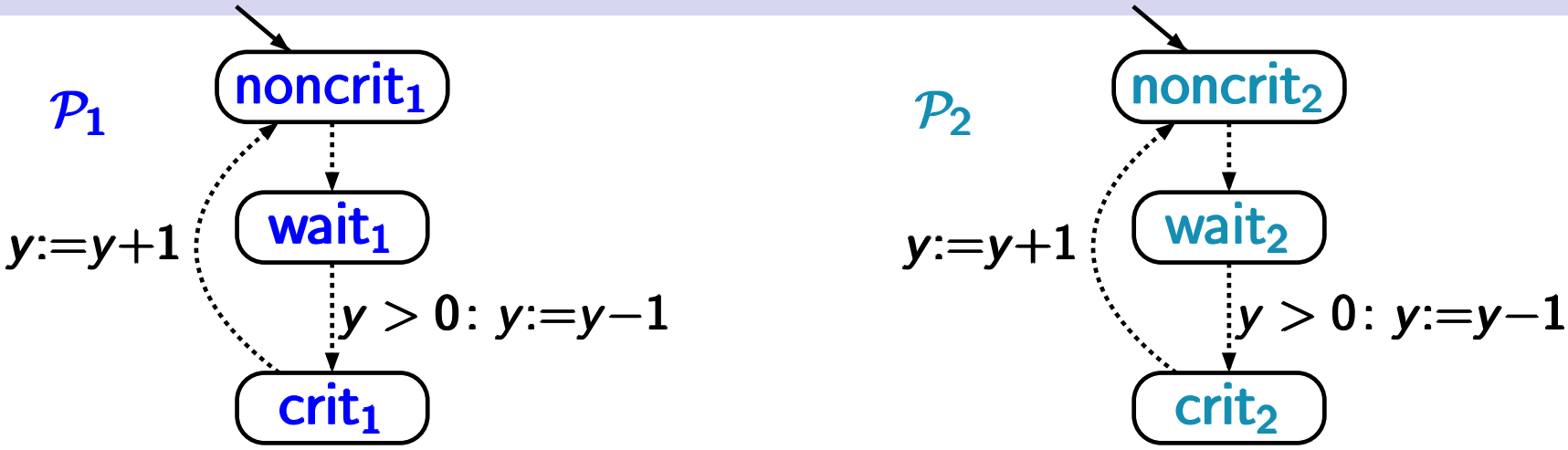
Mutual exclusion with semaphore



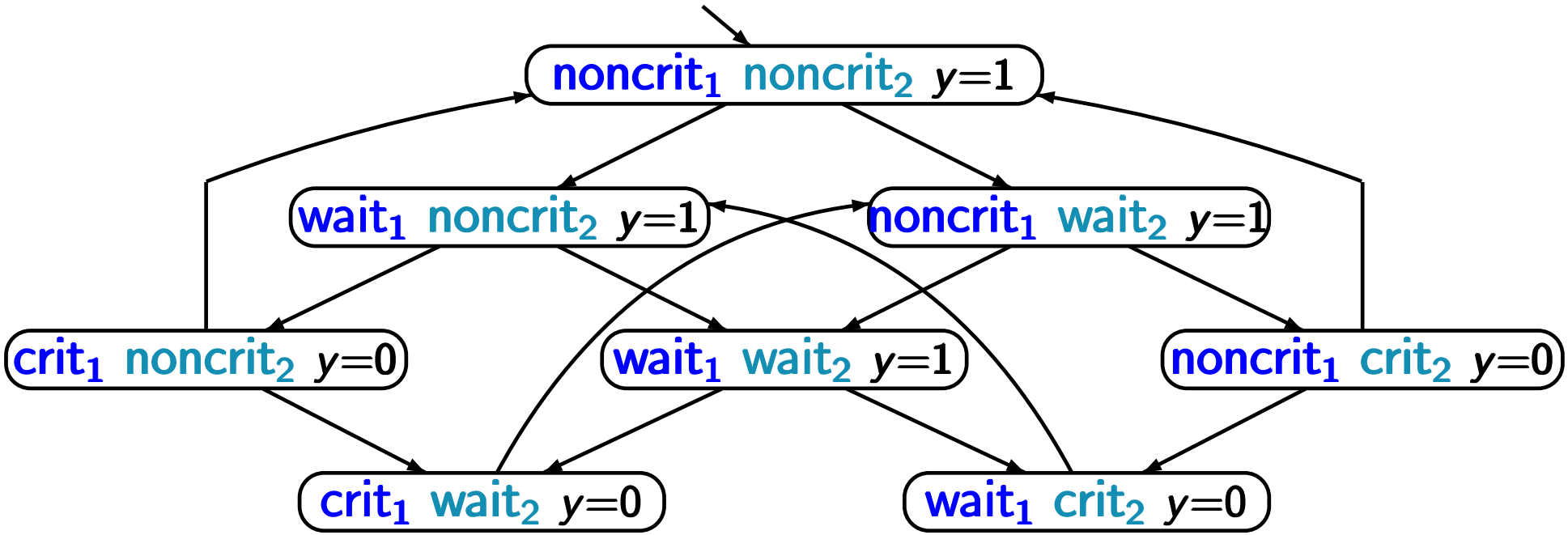
Mutual exclusion with semaphore



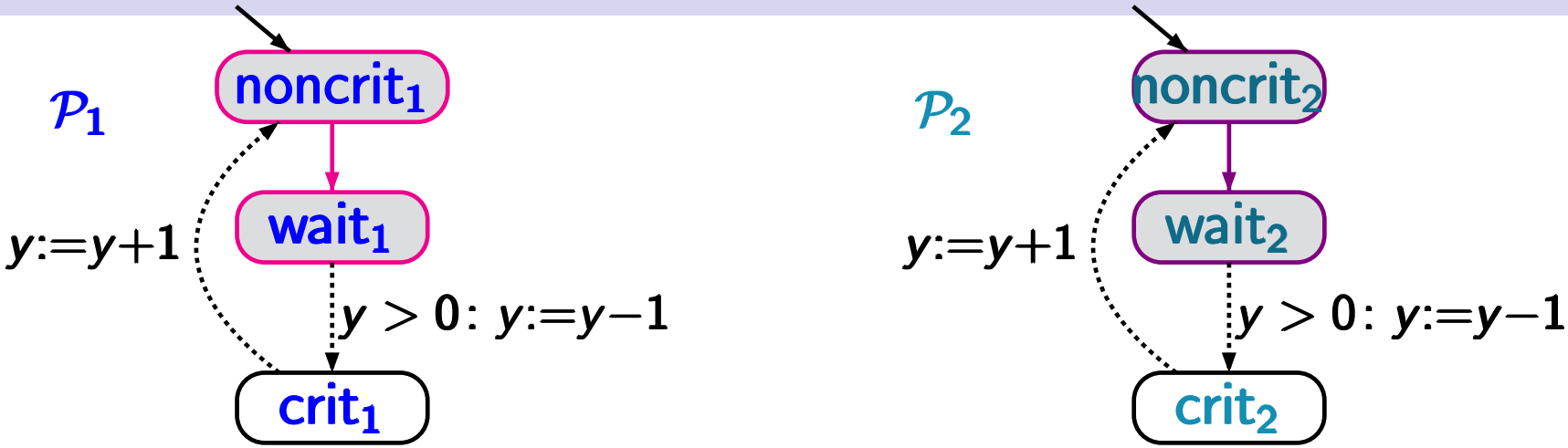
TS for mutual exclusion with semaphore



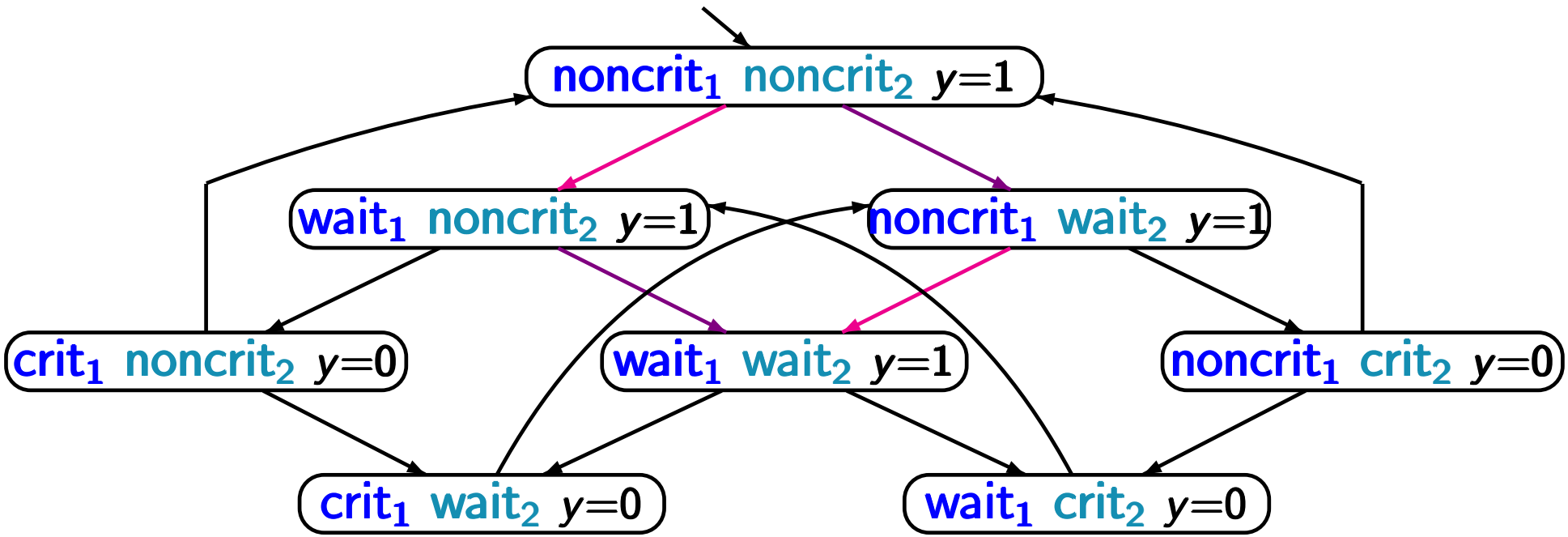
reachable fragment of the transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \mathcal{P}_2}$



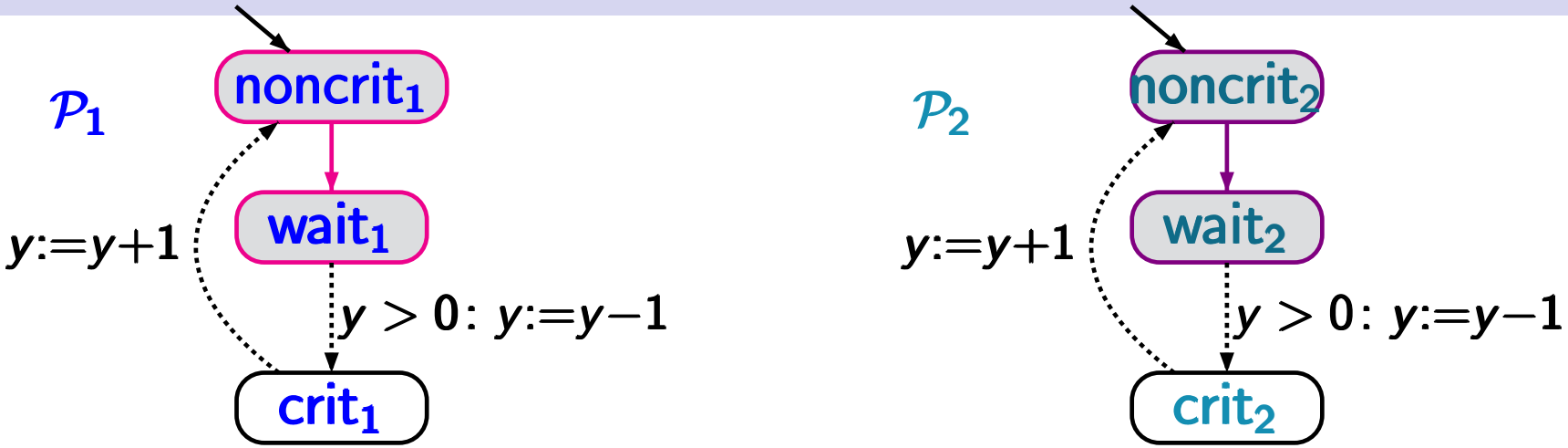
Concurrency of the request actions



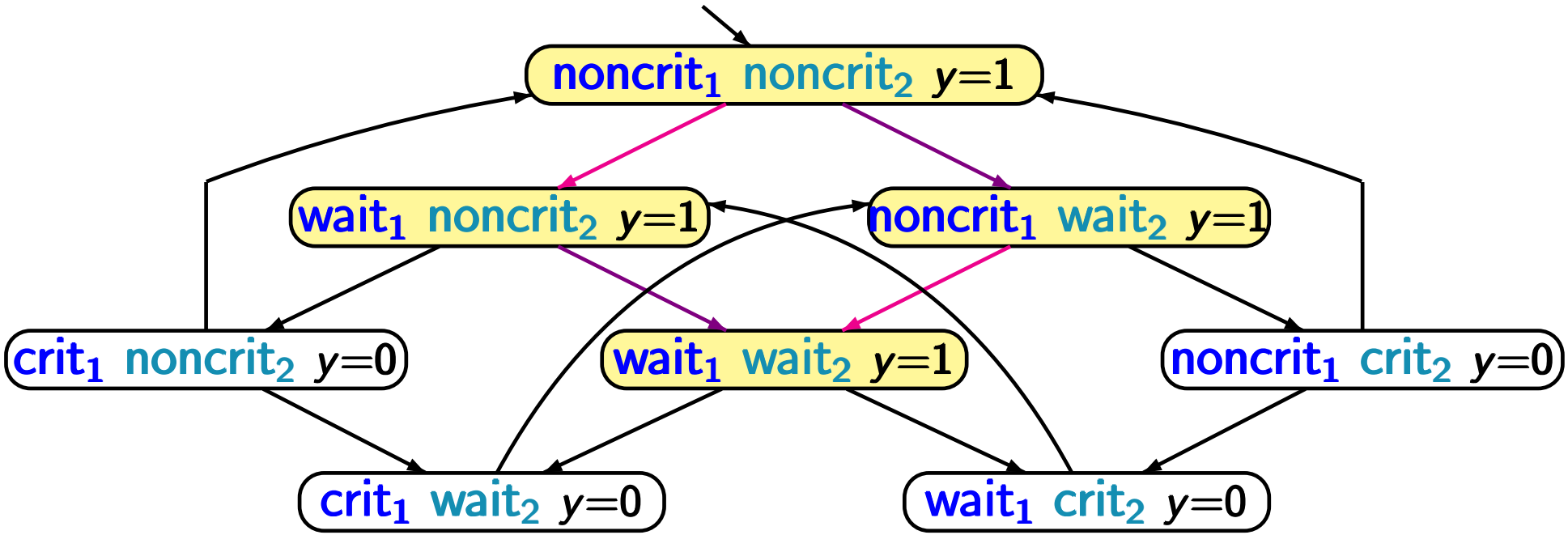
reachable fragment of the transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \mathcal{P}_2}$



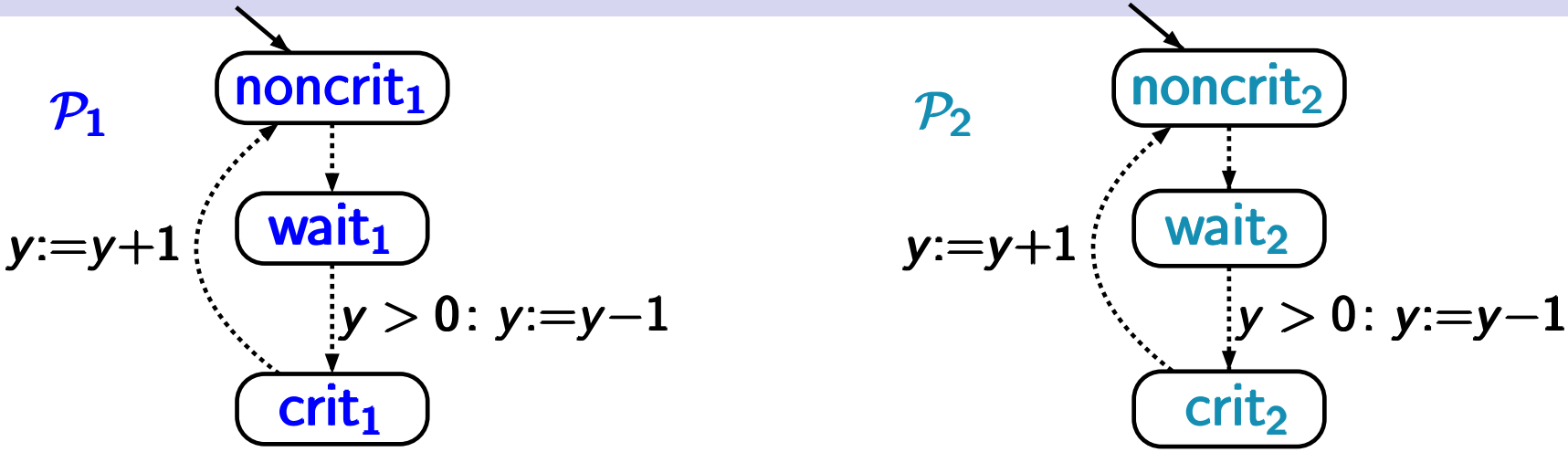
Concurrency of the request actions



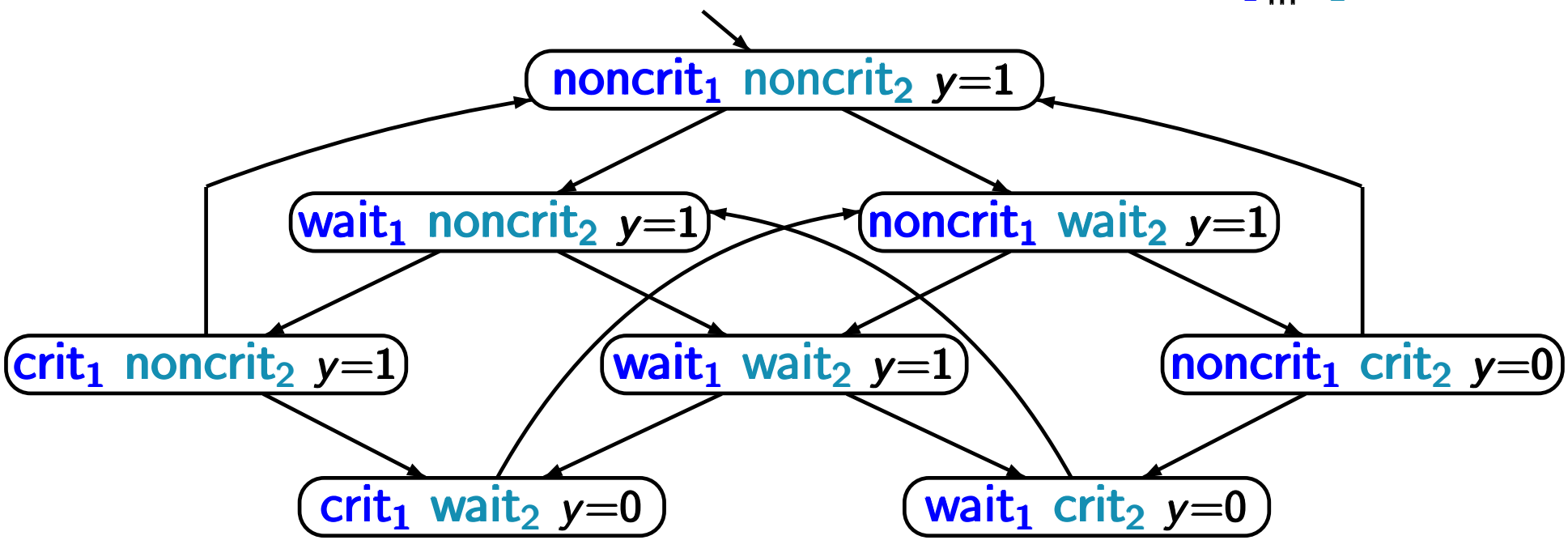
interleaving of the independent request actions



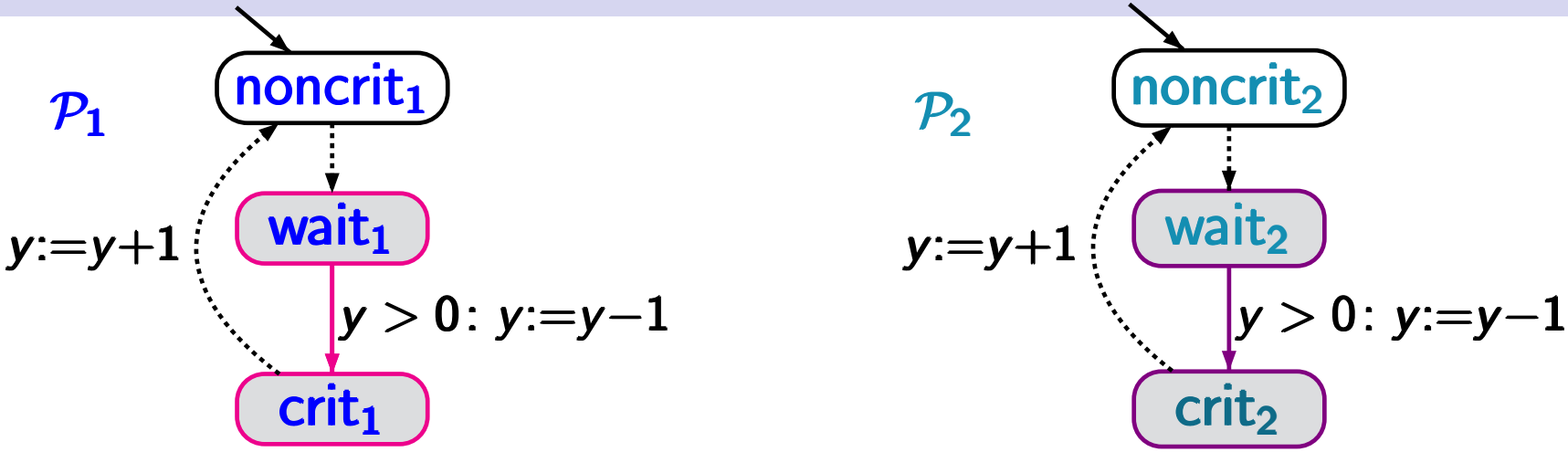
Competition



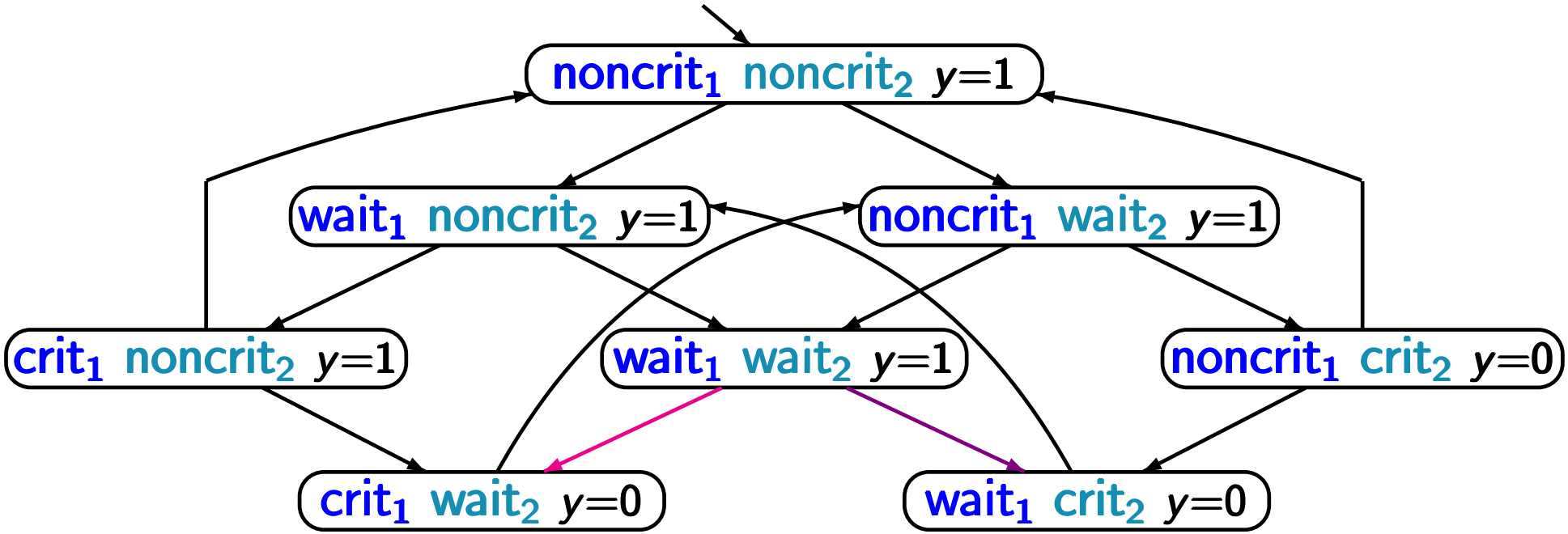
reachable fragment of the transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \mathcal{P}_2}$



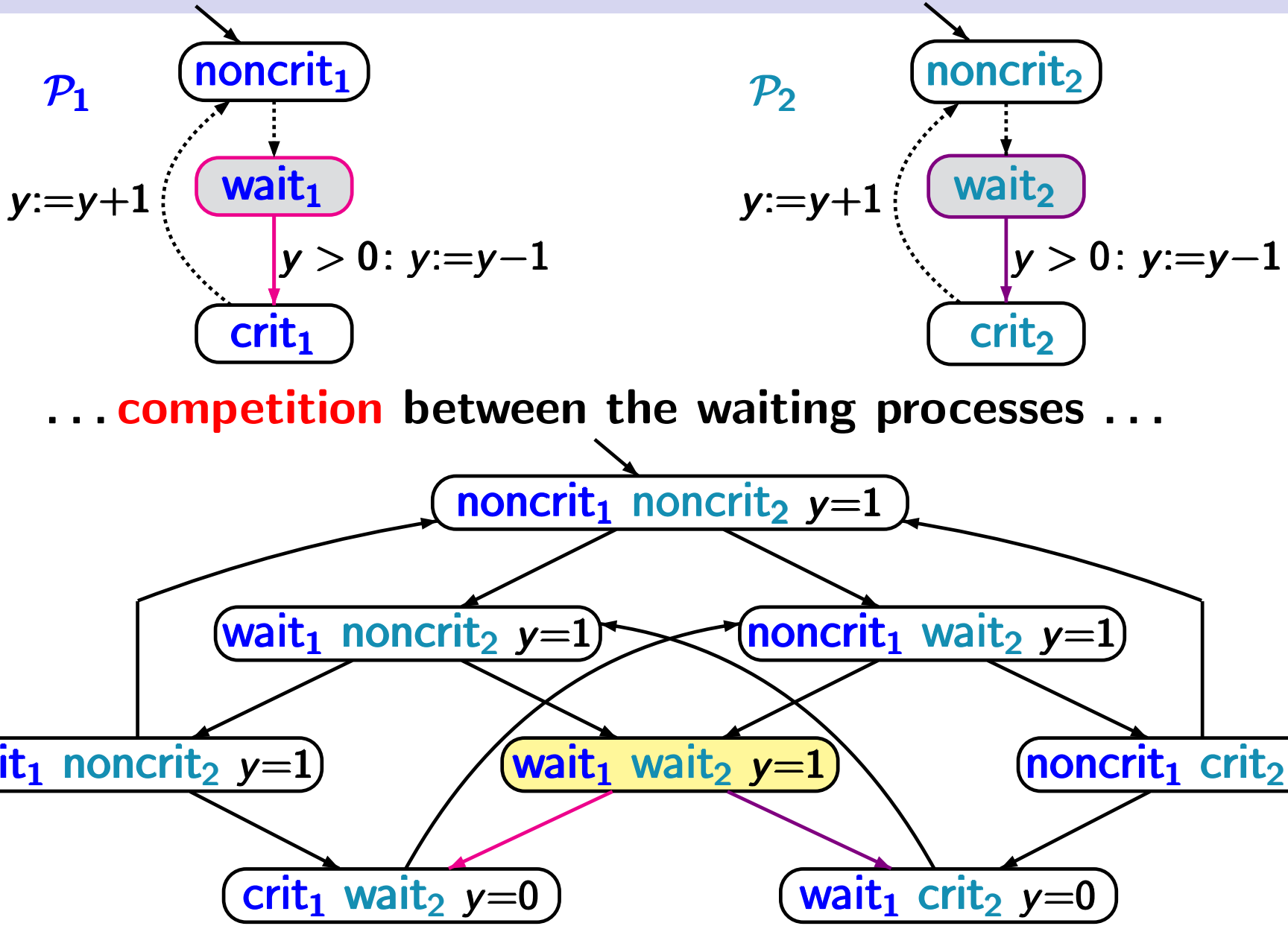
Competition



reachable fragment of the transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \mathcal{P}_2}$

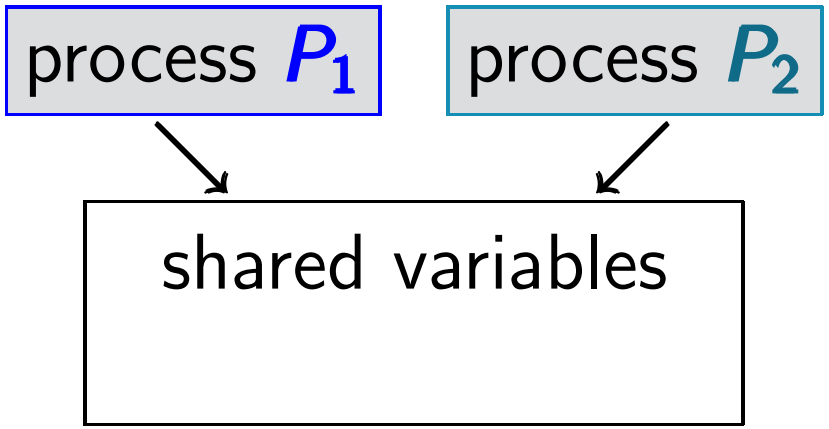


Competition

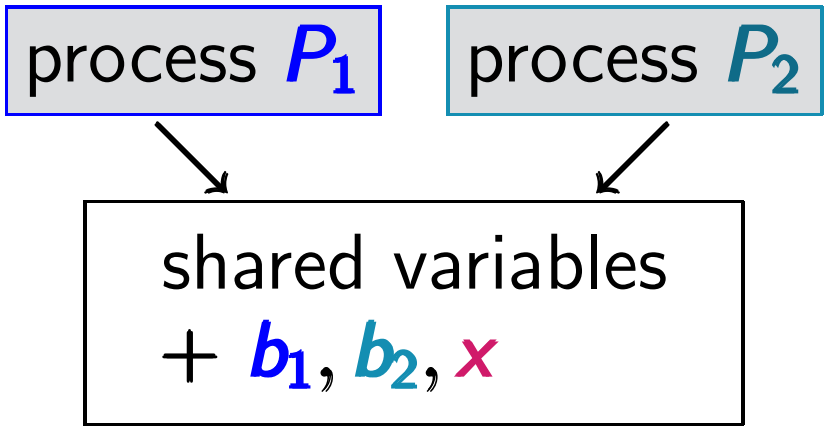


Peterson algorithm for mutual exclusion

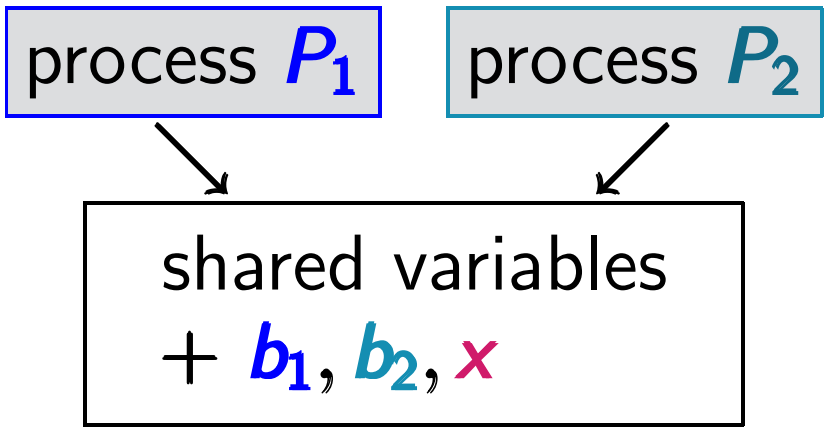
Peterson algorithm for mutual exclusion



Peterson algorithm for mutual exclusion

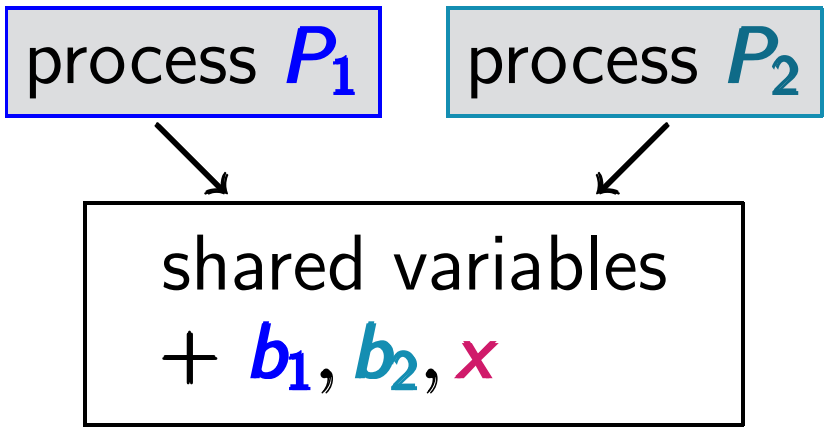


Peterson algorithm for mutual exclusion



b_1, b_2 Boolean variables, $x \in \{1, 2\}$

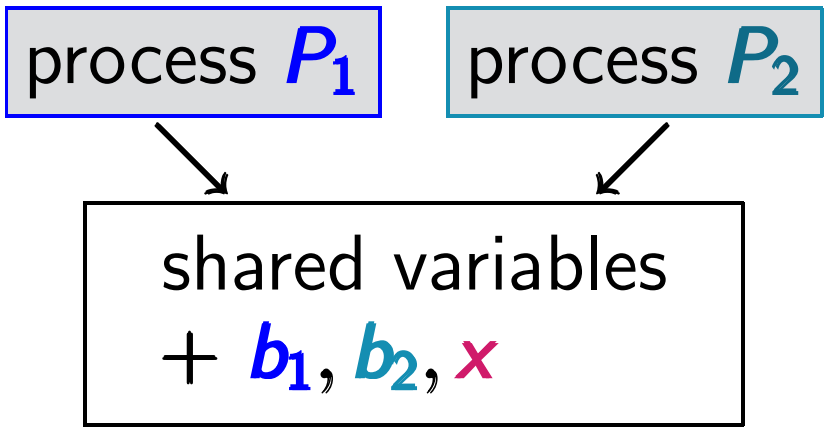
Peterson algorithm for mutual exclusion

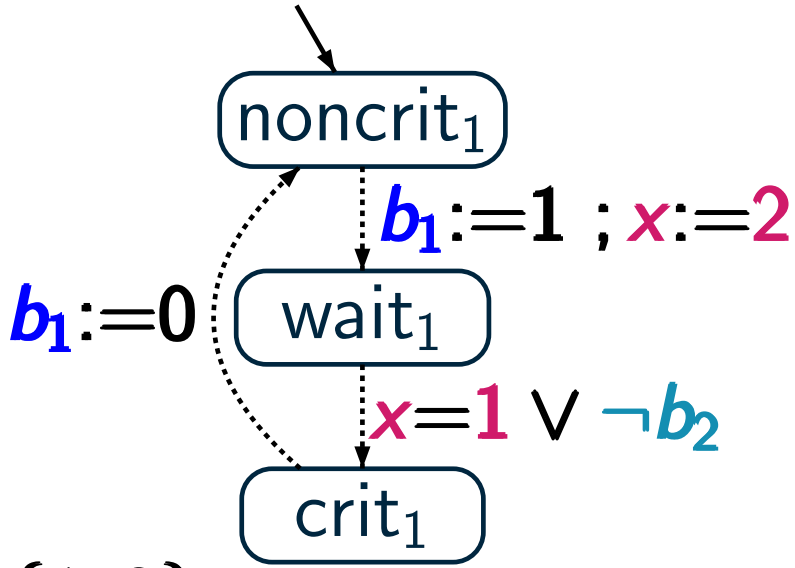


b_1, b_2 Boolean variables, $x \in \{1, 2\}$

```
LOOP FOREVER                                   (* protocol for P1 *)
    noncritical actions;
    b1:=1 ; x:=2;
    AWAIT x=1 ∨ ¬b2 DO critical section OD
    b1:=0
END LOOP
```

Peterson algorithm for mutual exclusion

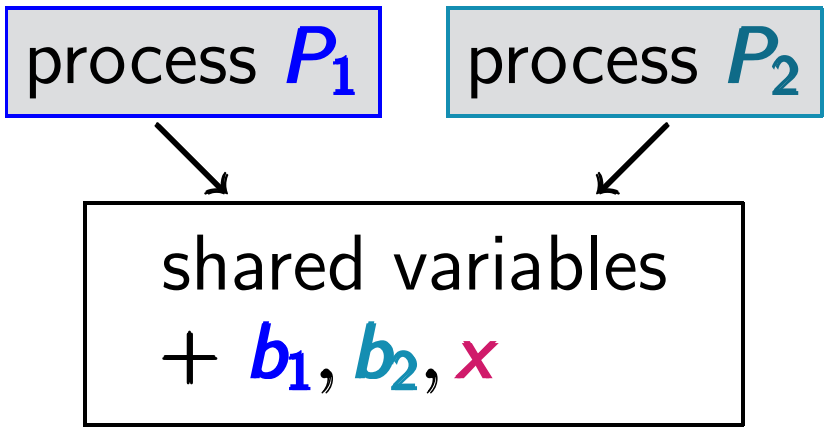


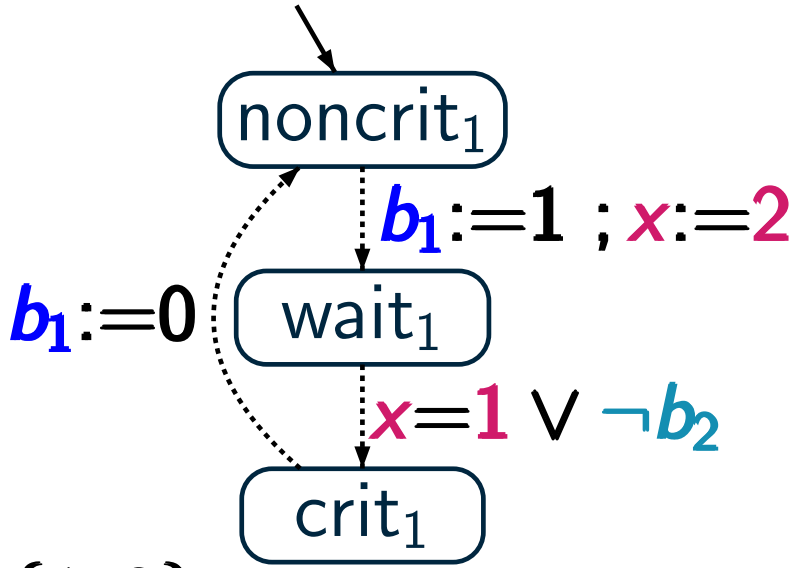


b_1, b_2 Boolean variables, $x \in \{1, 2\}$

```
LOOP FOREVER                          (* protocol for P1 *)
    noncritical actions;
    b1:=1 ; x:=2;
    AWAIT x=1 ∨ ¬b2 DO critical section OD
    b1:=0
END LOOP
```

Peterson algorithm for mutual exclusion

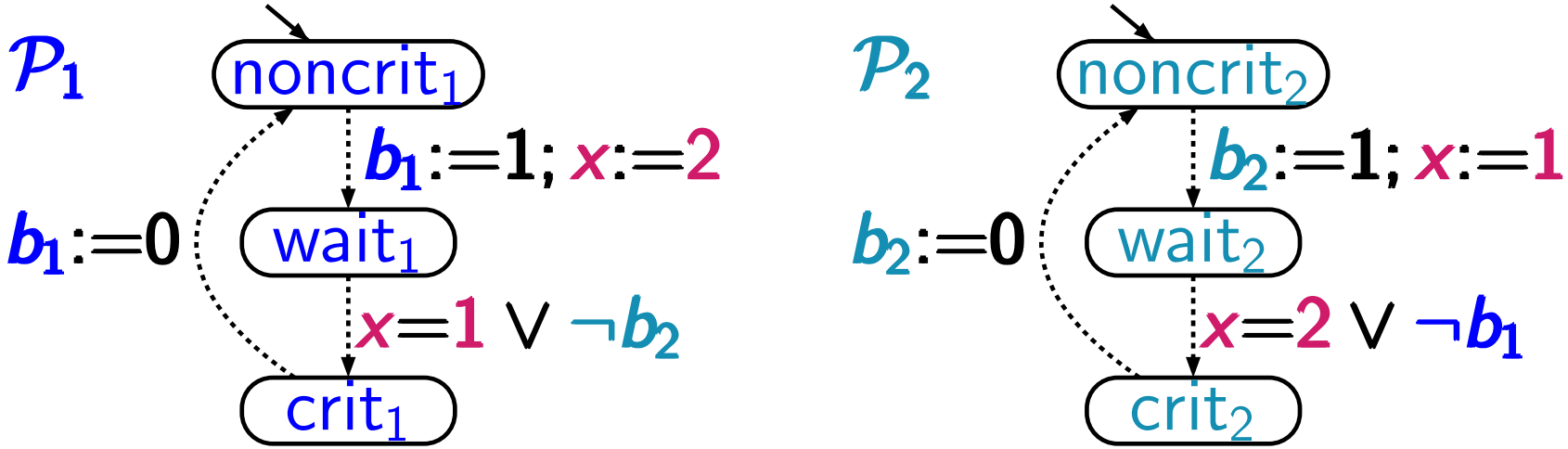




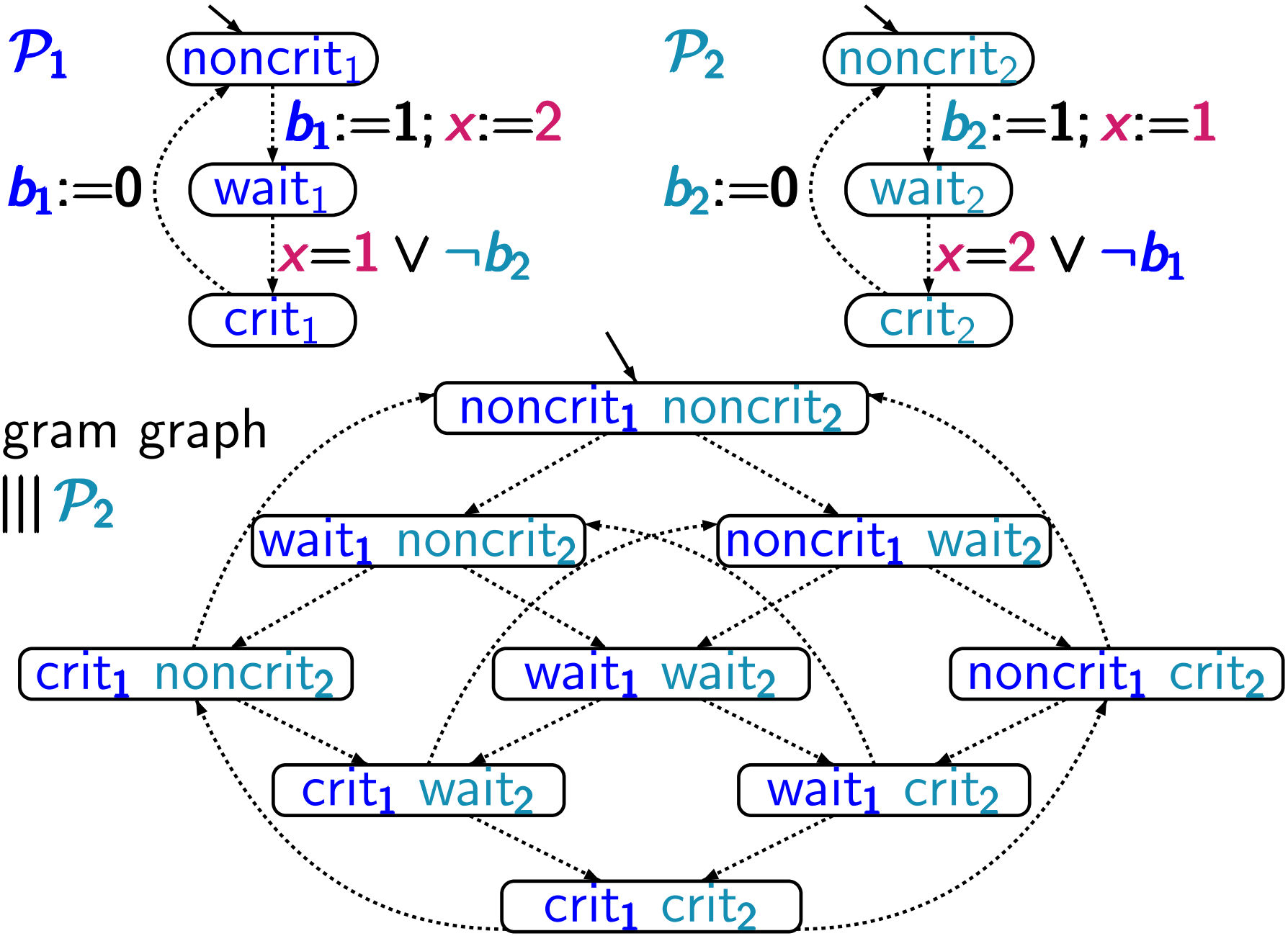
b_1, b_2 Boolean variables, $x \in \{1, 2\}$

```
LOOP FOREVER                          (* protocol for P1 *)
    noncritical actions;
    atomic{b1:=1 ; x:=2};
    AWAIT x=1 ∨ ¬b2 DO critical section OD
    b1:=0
END LOOP
```

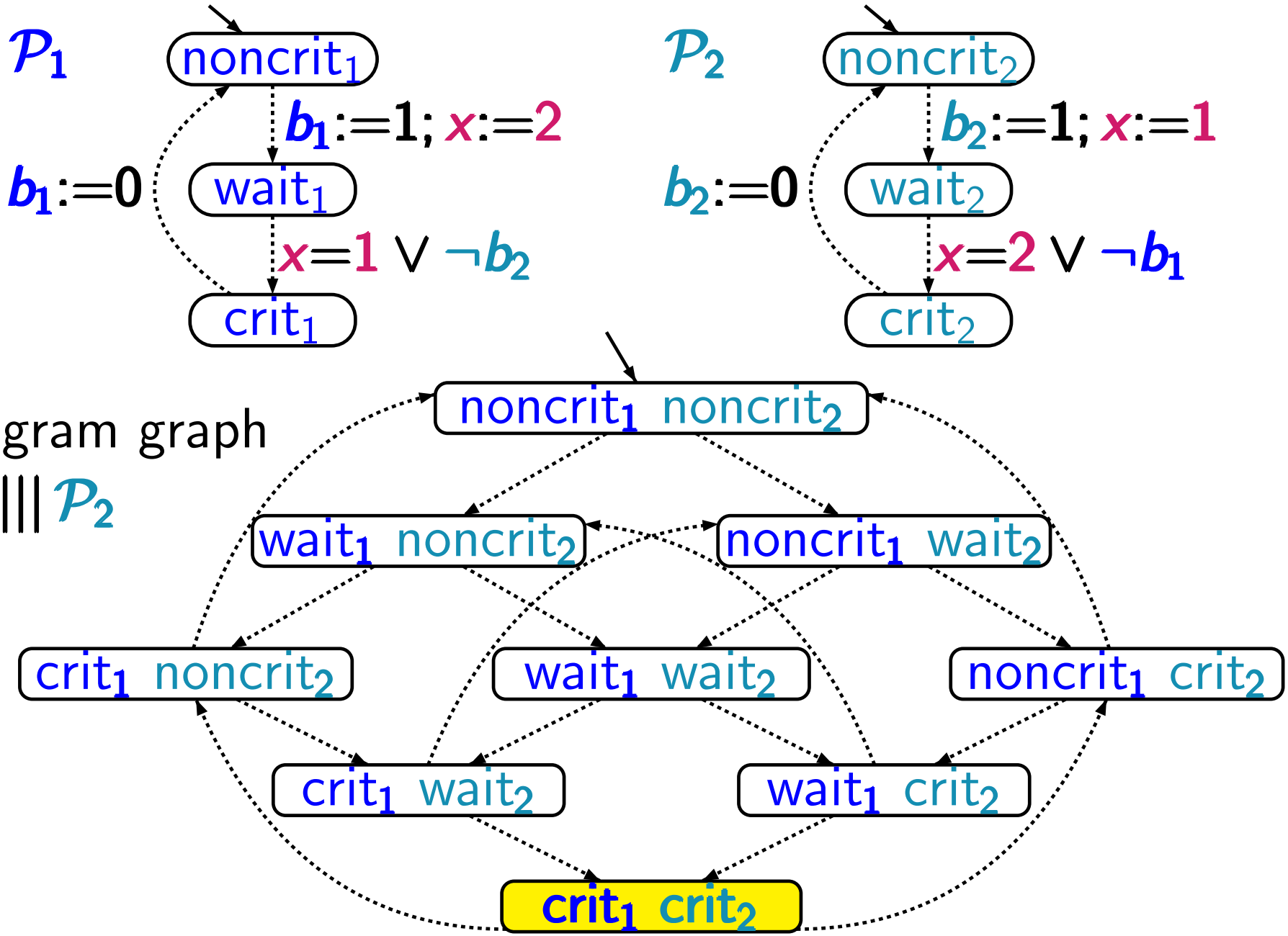
Program graphs for Peterson algorithm



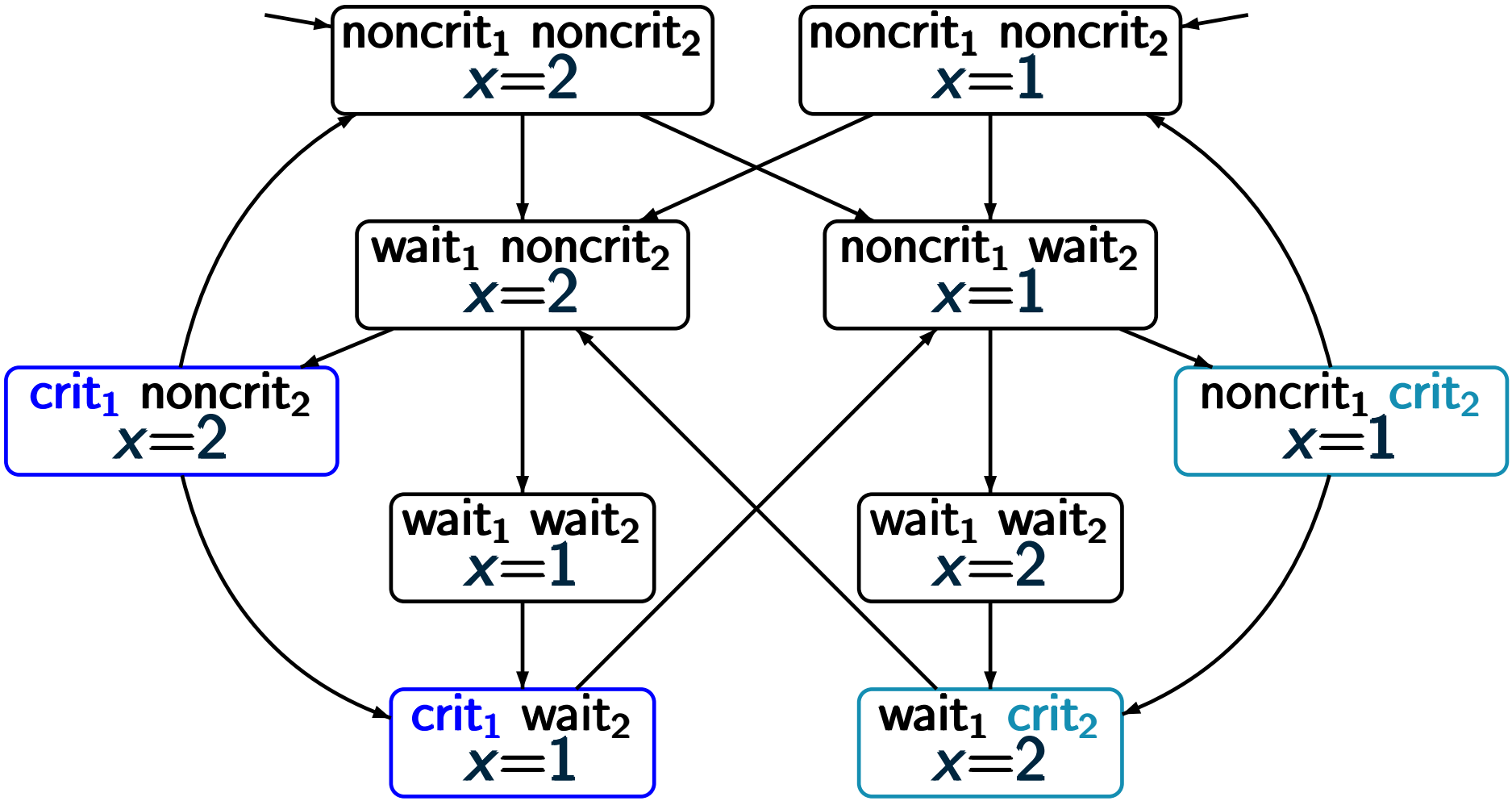
Program graphs for Peterson algorithm



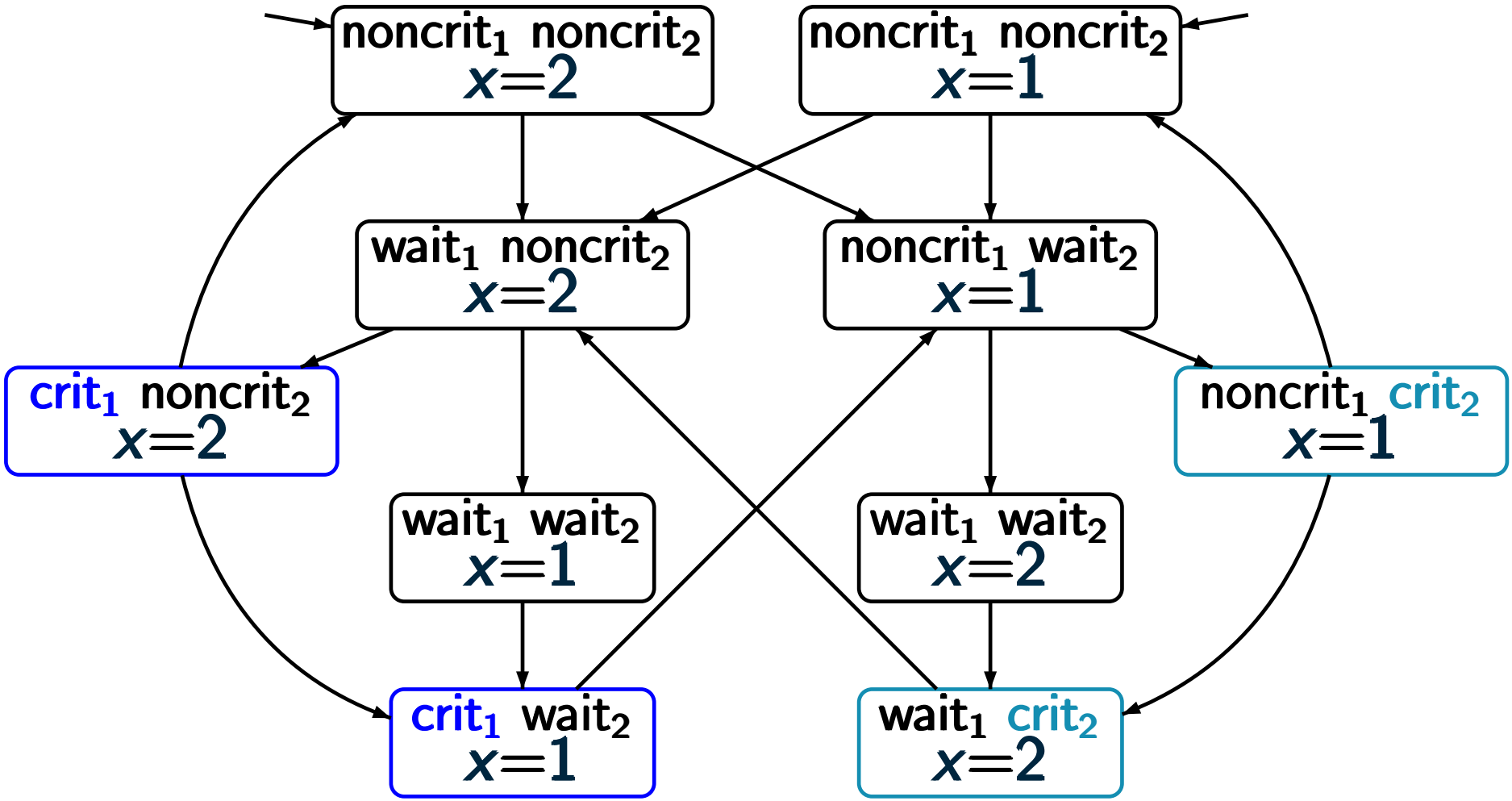
Program graphs for Peterson algorithm



TS for the Peterson algorithm



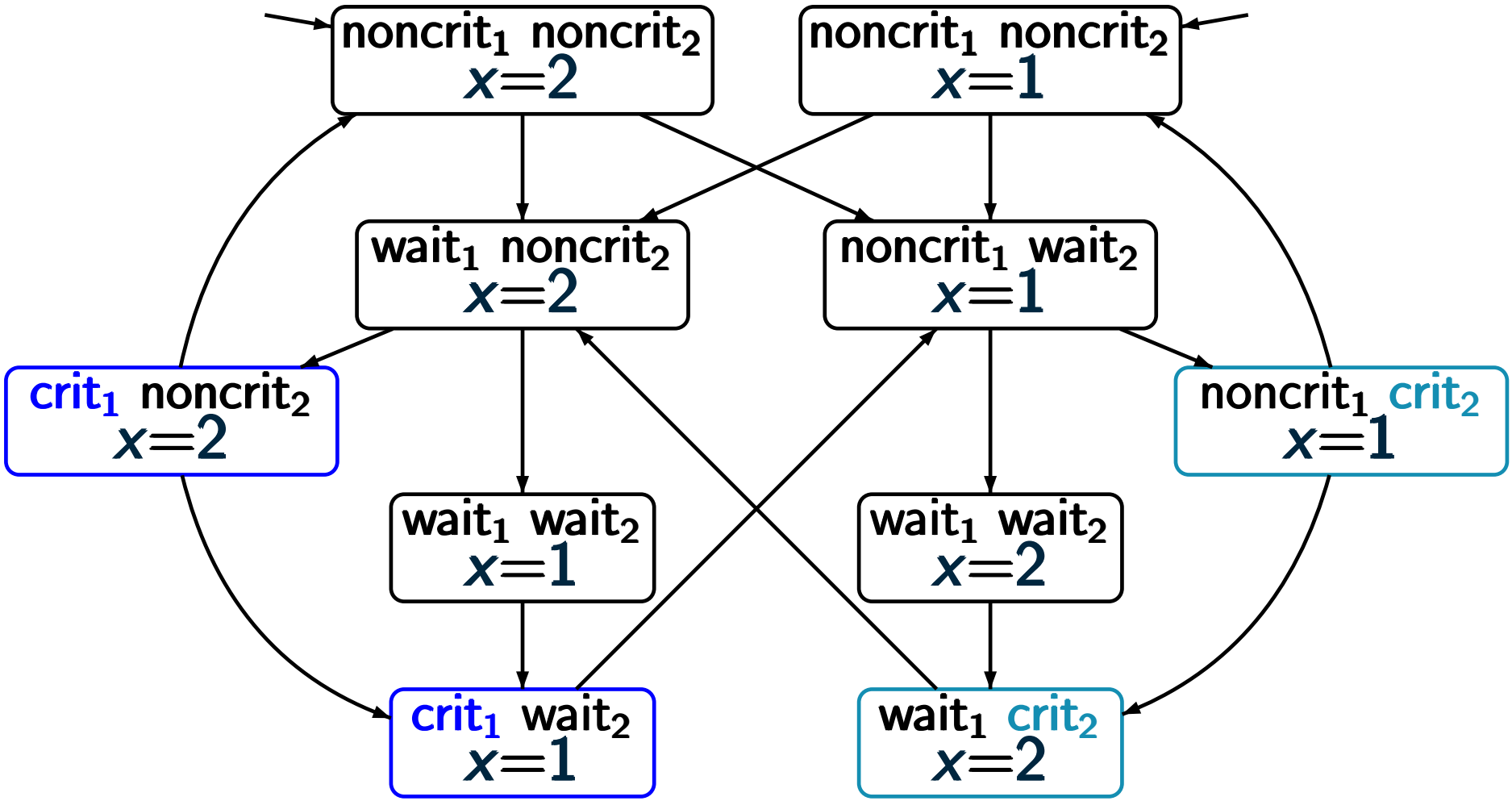
TS for the Peterson algorithm



value of b_1 is given by $\mathbf{wait_1} \vee \mathbf{crit_1}$

value of b_2 is given by $\mathbf{wait_2} \vee \mathbf{crit_2}$

TS for the Peterson algorithm

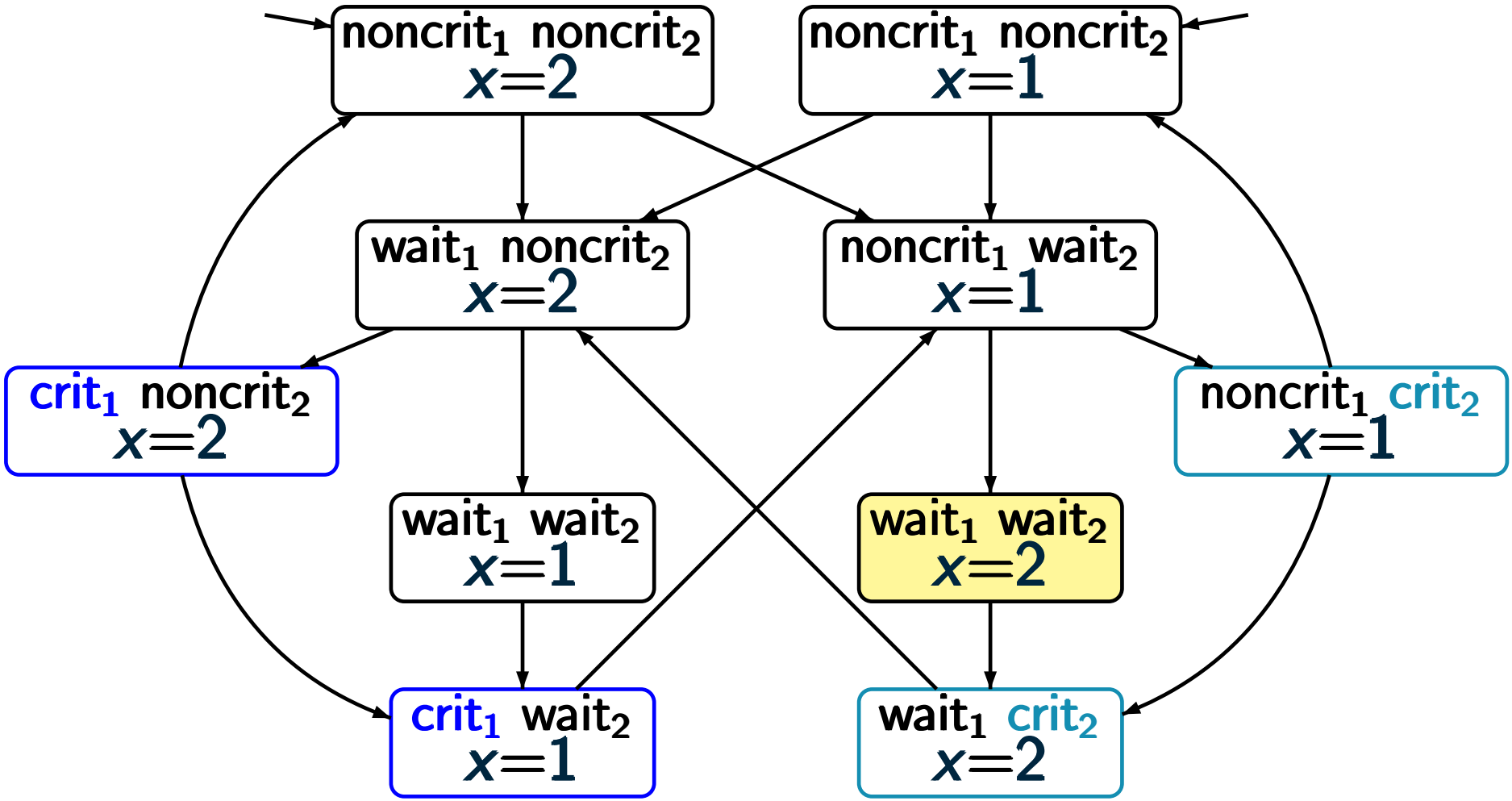


value of b_1 is given by $\mathbf{wait_1} \vee \mathbf{crit_1}$

value of b_2 is given by $\mathbf{wait_2} \vee \mathbf{crit_2}$

\+ unreachable states

TS for the Peterson algorithm

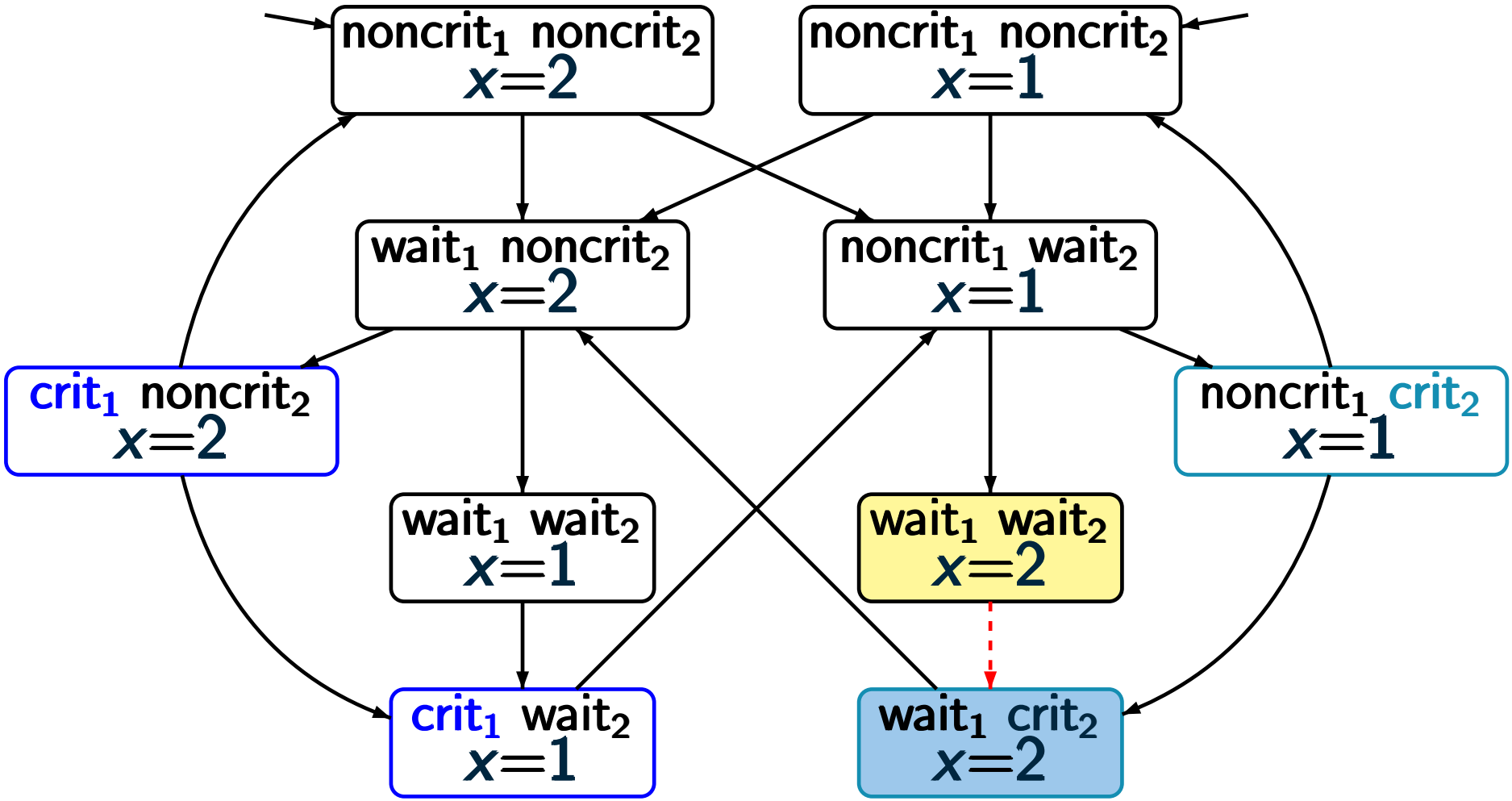


value of b_1 is given by $\mathbf{wait_1} \vee \mathbf{crit_1}$

value of b_2 is given by $\mathbf{wait_2} \vee \mathbf{crit_2}$

+ unreachable states

TS for the Peterson algorithm

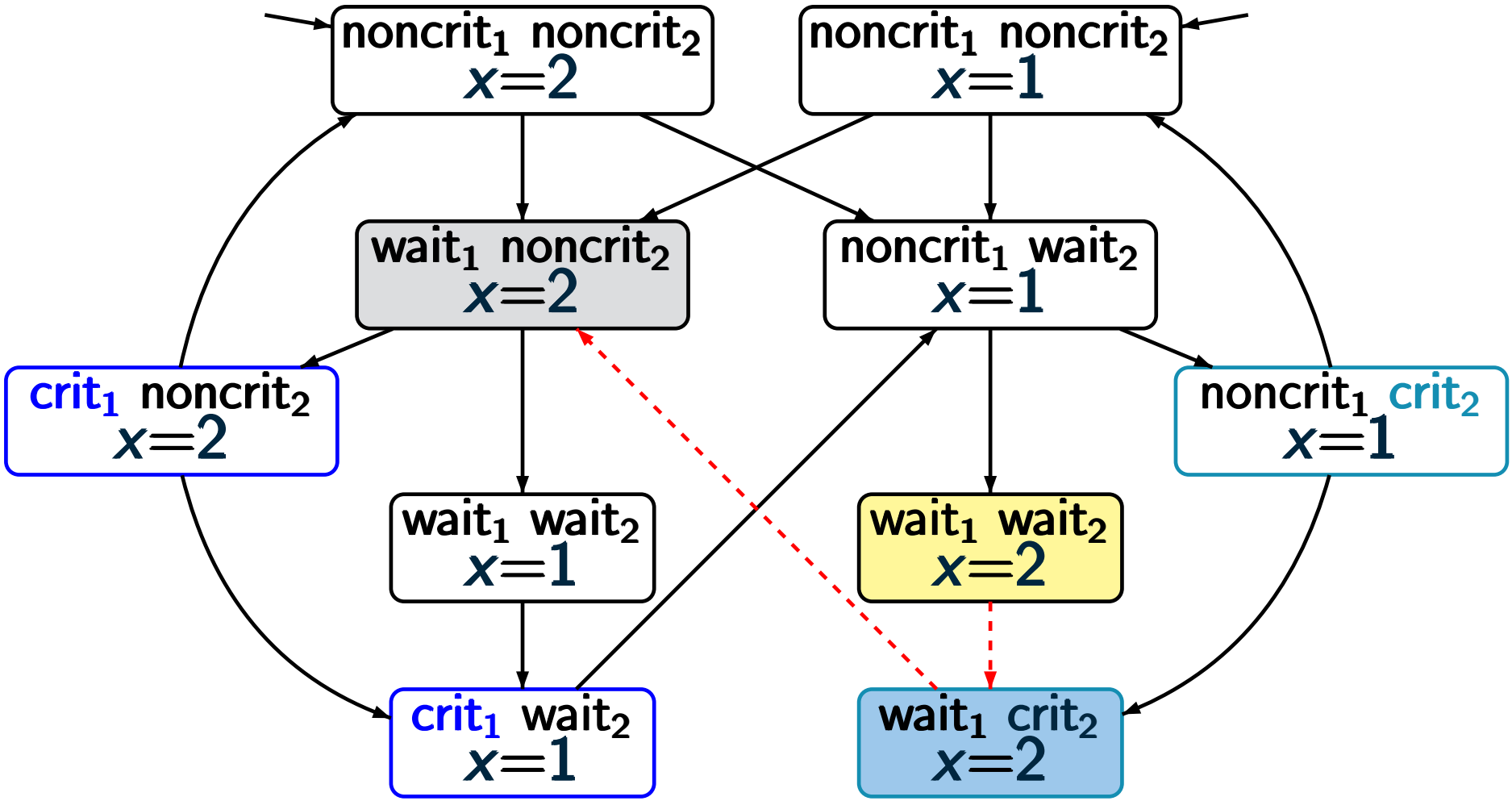


value of b_1 is given by $\mathbf{wait_1} \vee \mathbf{crit_1}$

value of b_2 is given by $\mathbf{wait_2} \vee \mathbf{crit_2}$

+ unreachable states

TS for the Peterson algorithm

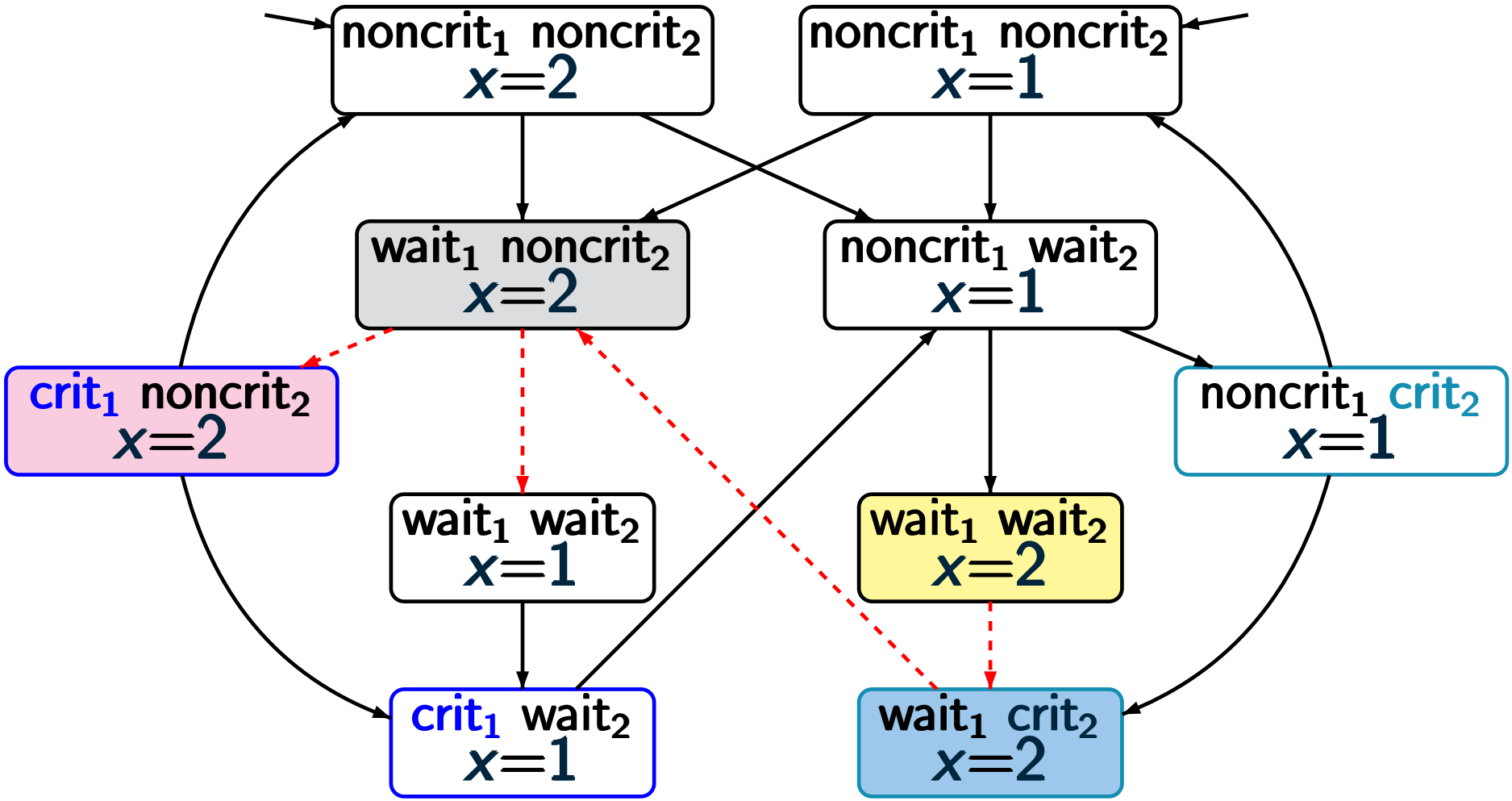


value of b_1 is given by $\mathbf{wait_1} \vee \mathbf{crit_1}$

value of b_2 is given by $\mathbf{wait_2} \vee \mathbf{crit_2}$

\+ unreachable states

TS for the Peterson algorithm

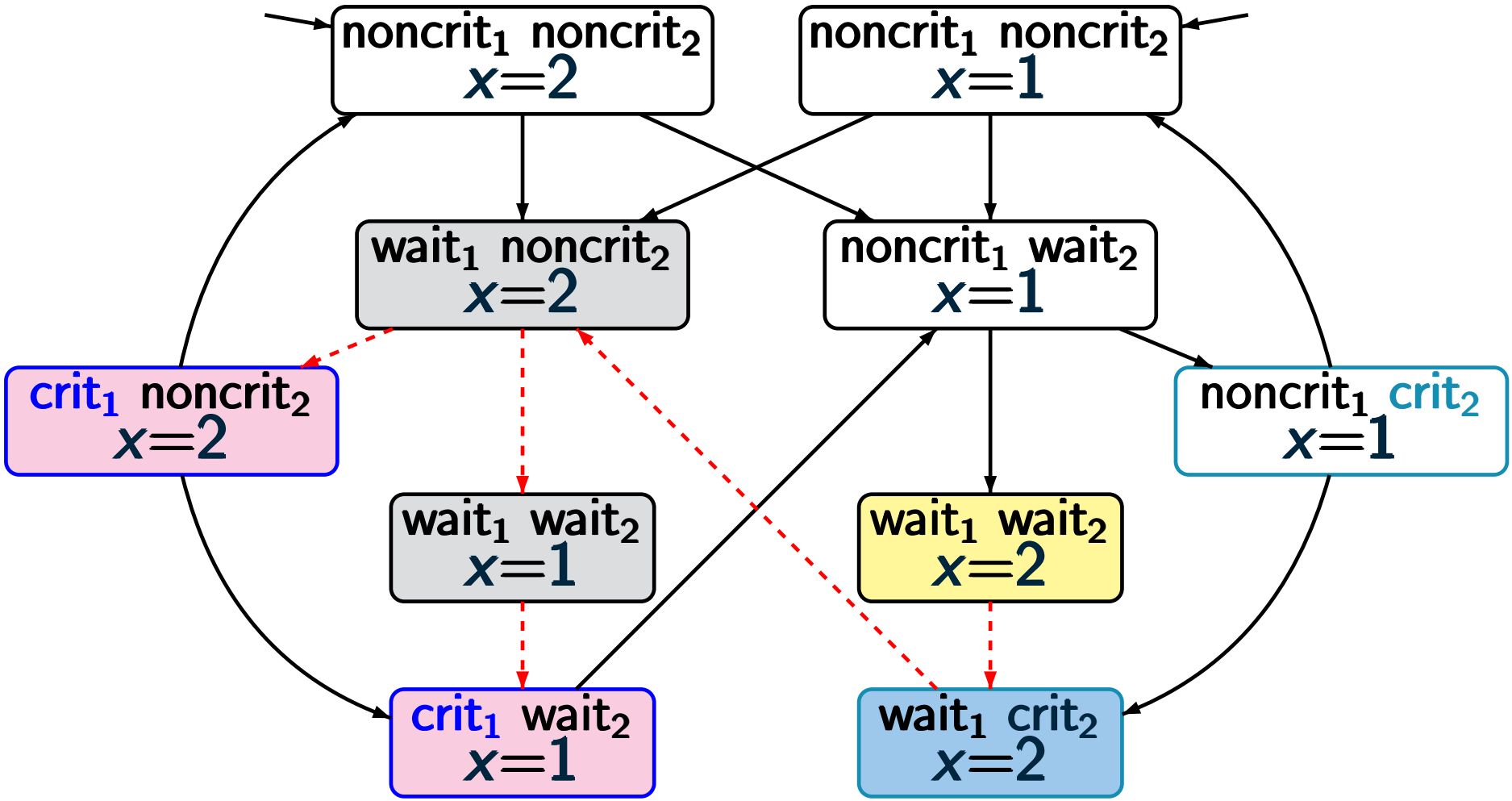


value of b_1 is given by $\mathbf{wait_1} \vee \mathbf{crit_1}$

value of b_2 is given by $\mathbf{wait_2} \vee \mathbf{crit_2}$

+ unreachable states

TS for the Peterson algorithm

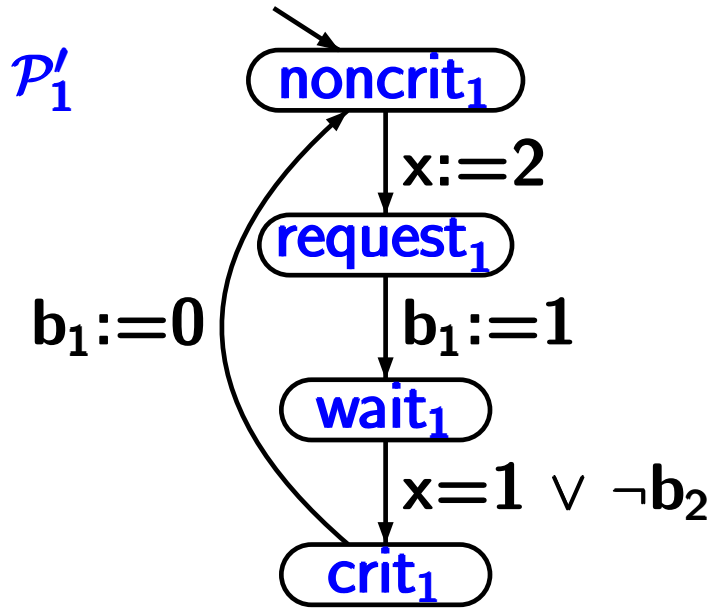


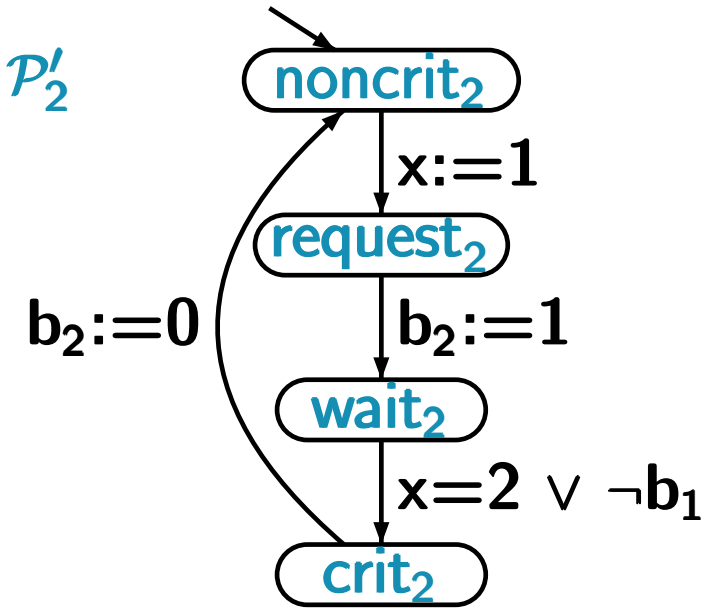
value of b_1 is given by $\mathbf{wait_1} \vee \mathbf{crit_1}$

value of b_2 is given by $\mathbf{wait_2} \vee \mathbf{crit_2}$

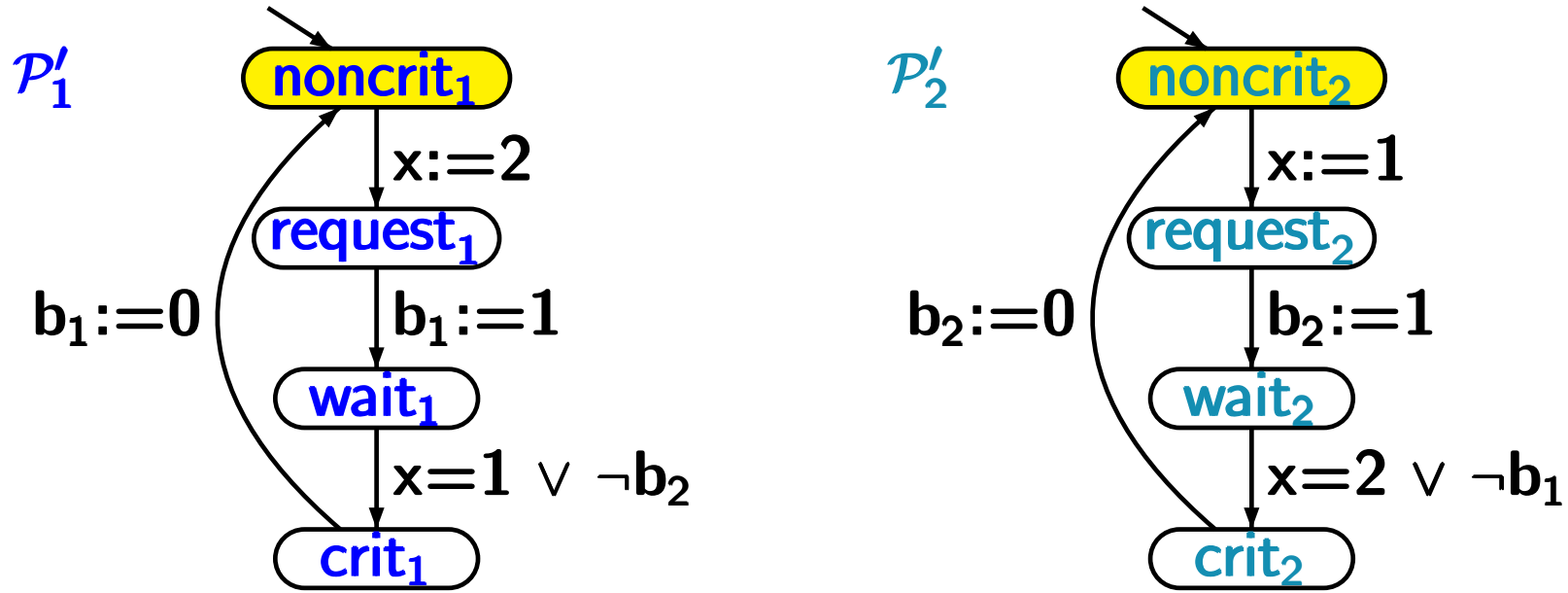
+ unreachable states

Variant of Peterson algorithm





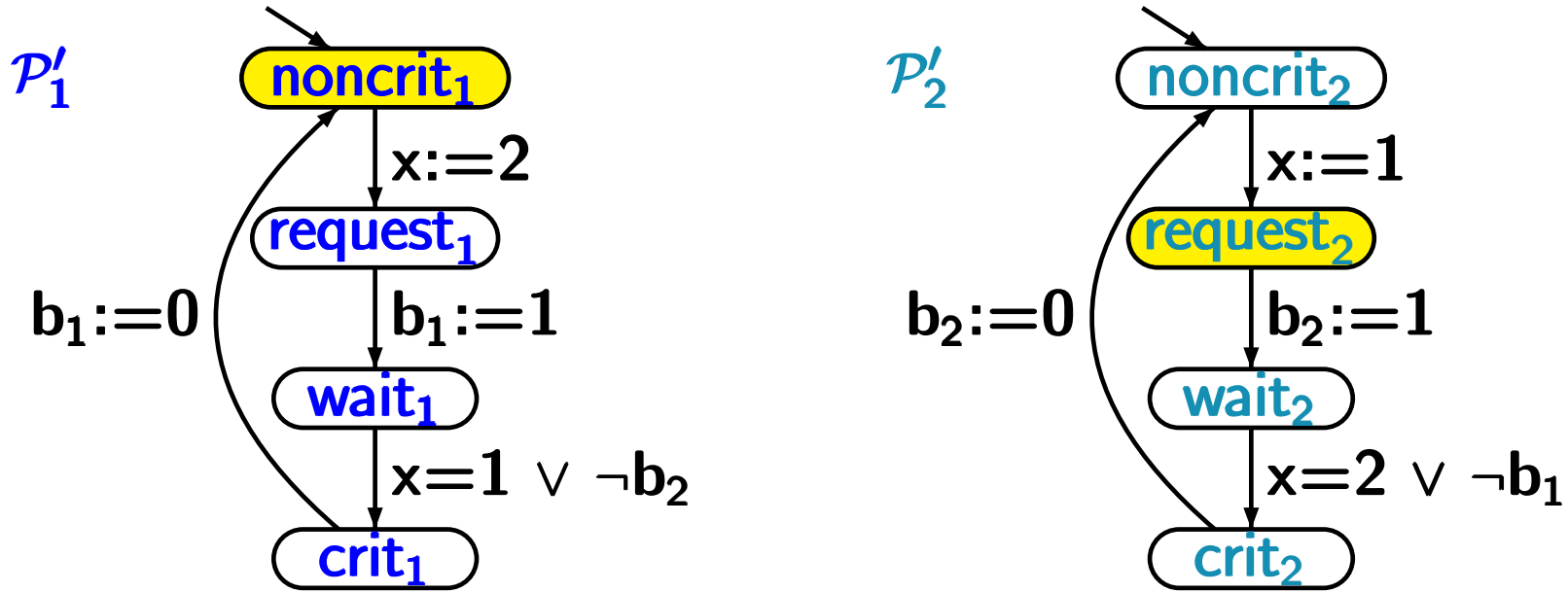
Variant of Peterson algorithm



possible executions

noncrit$_1$ noncrit$_2$ x=1 ¬b$_1$ ¬b$_2$

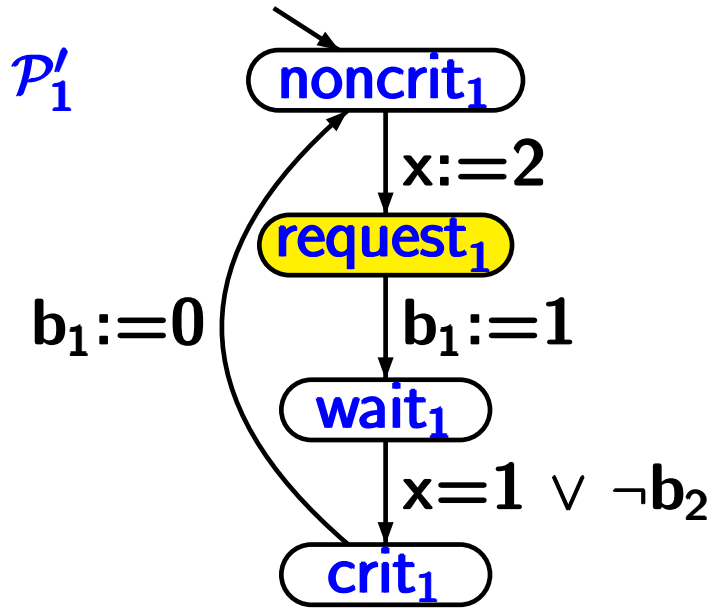
Variant of Peterson algorithm

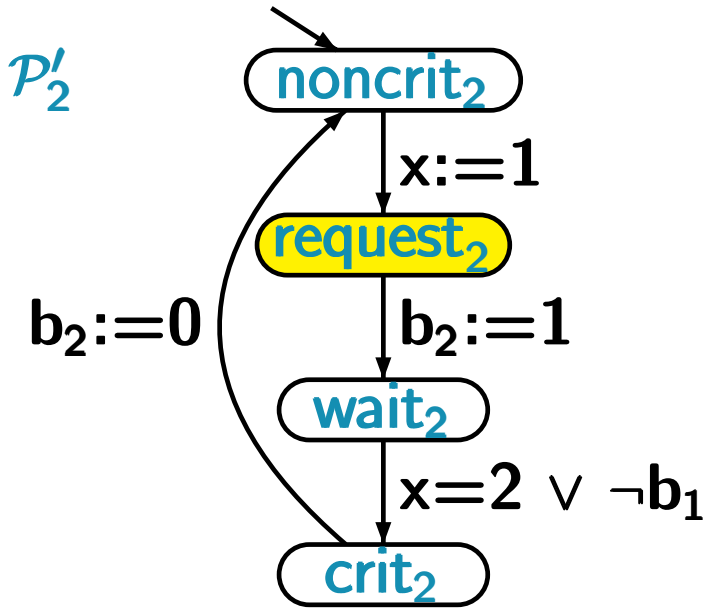


possible executions

noncrit$_1$	noncrit$_2$	x=1	¬b$_1$	¬b$_2$
noncrit$_1$	request$_2$	x=1	¬ b$_1$	¬b$_2$

Variant of Peterson algorithm

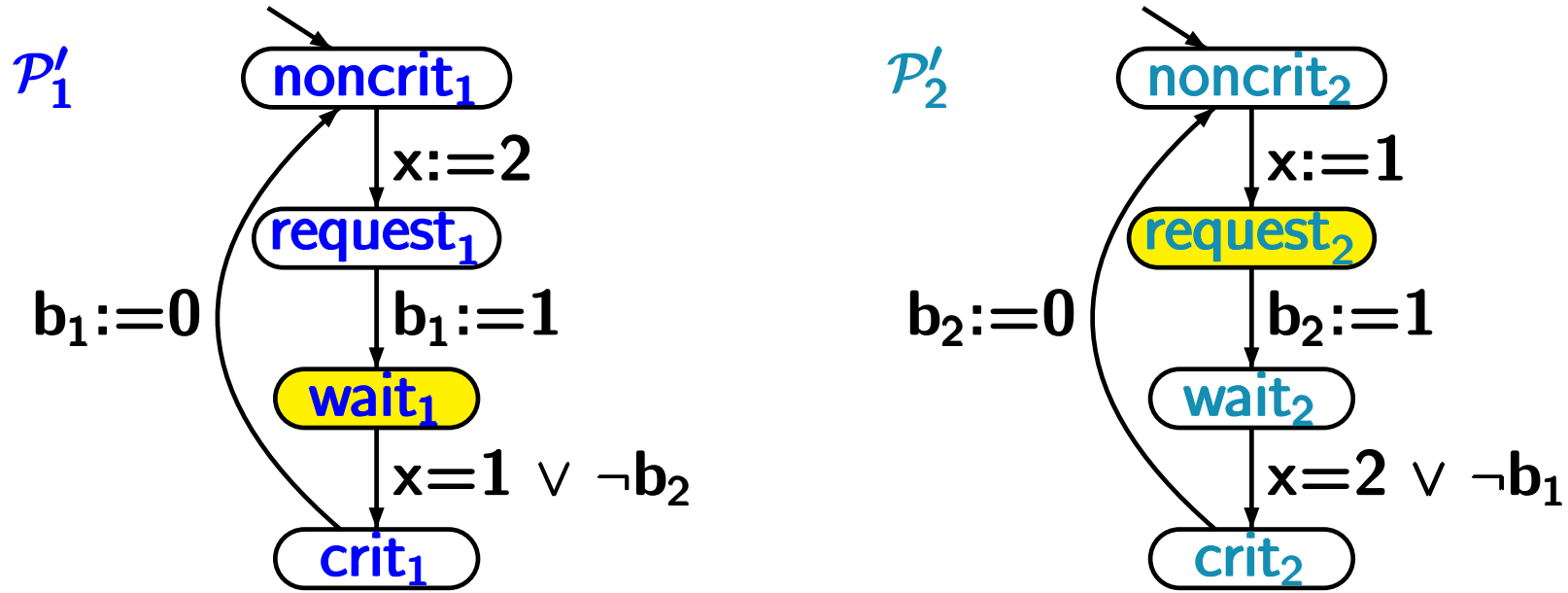




possible executions

noncrit$_1$	noncrit$_2$	x=1	¬b$_1$	¬b$_2$
noncrit$_1$	request$_2$	x=1	¬ b$_1$	¬b$_2$
request$_1$	request$_2$	x=2	¬b$_1$	¬b$_2$

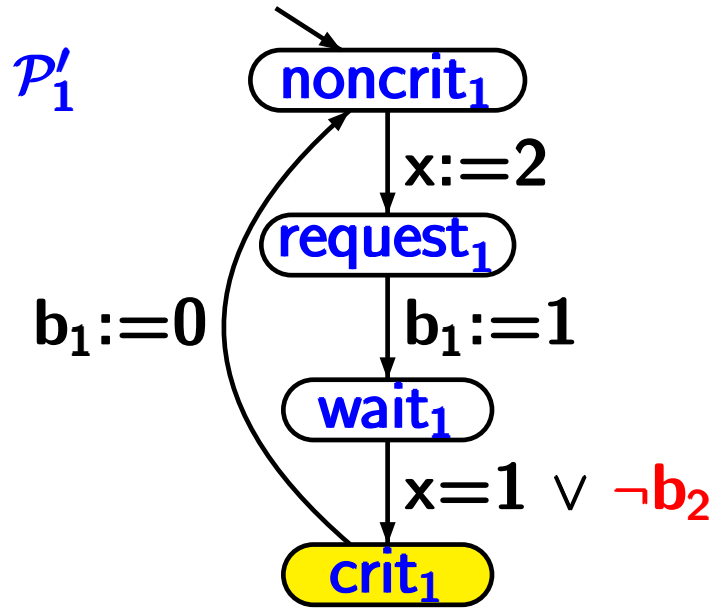
Variant of Peterson algorithm

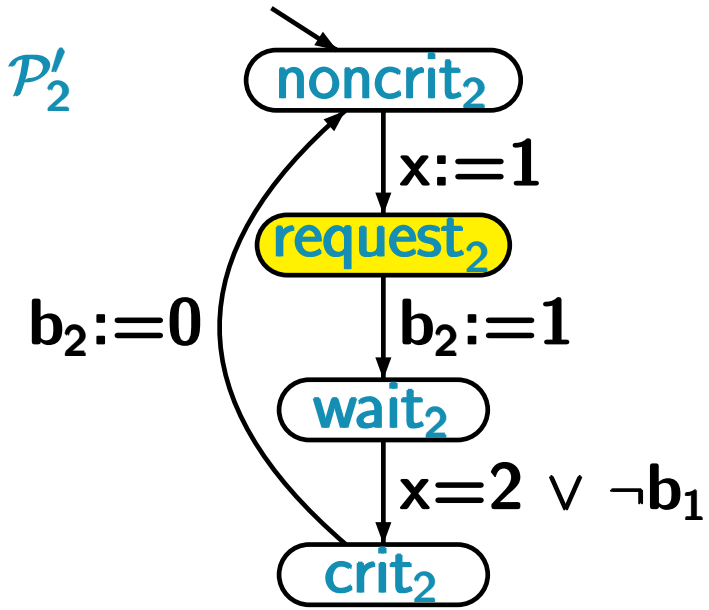


possible executions

noncrit$_1$	noncrit$_2$	x=1	¬b$_1$	¬b$_2$
noncrit$_1$	request$_2$	x=1	¬ b$_1$	¬b$_2$
request$_1$	request$_2$	x=2	¬b$_1$	¬b$_2$
wait$_1$	request$_2$	x=2	b$_1$	¬b$_2$

Variant of Peterson algorithm

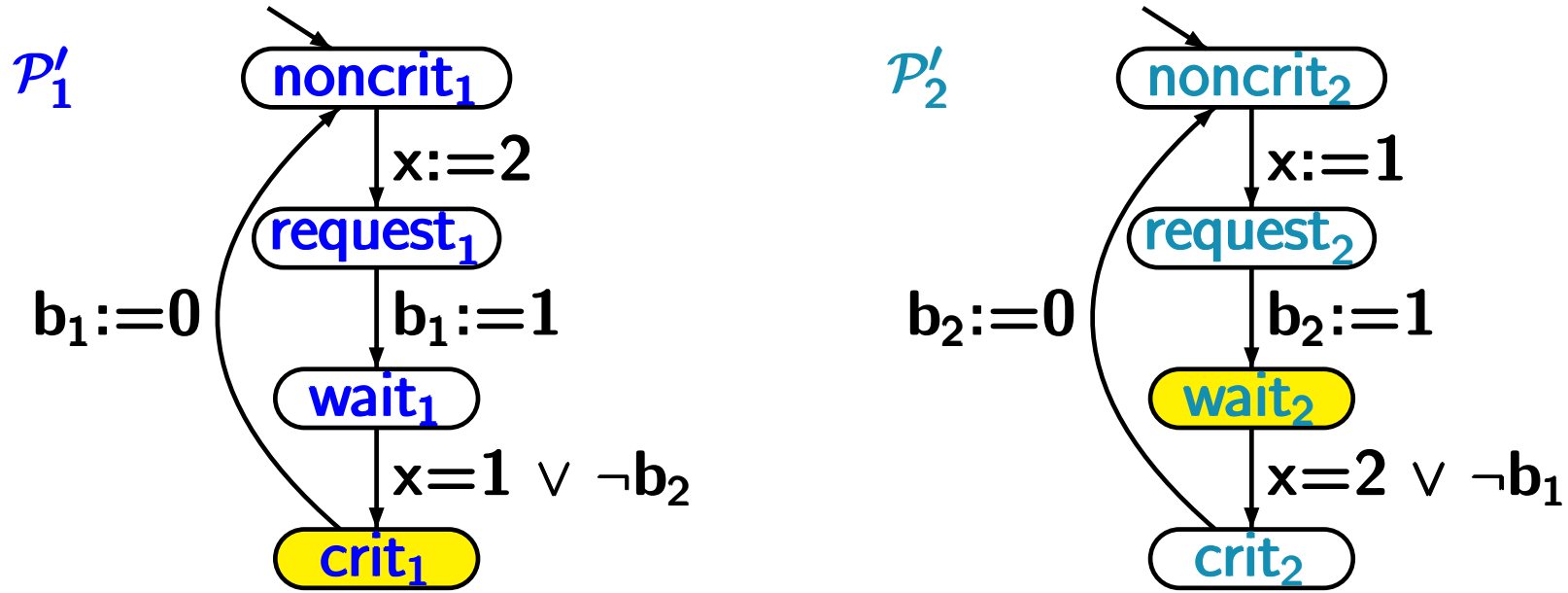




possible executions

noncrit$_1$	noncrit$_2$	x=1	¬b$_1$	¬b$_2$
noncrit$_1$	request$_2$	x=1	¬ b$_1$	¬b$_2$
request$_1$	request$_2$	x=2	¬b$_1$	¬b$_2$
wait$_1$	request$_2$	x=2	b$_1$	¬b$_2$
crit$_1$	request$_2$	x=2	b$_1$	¬b$_2$

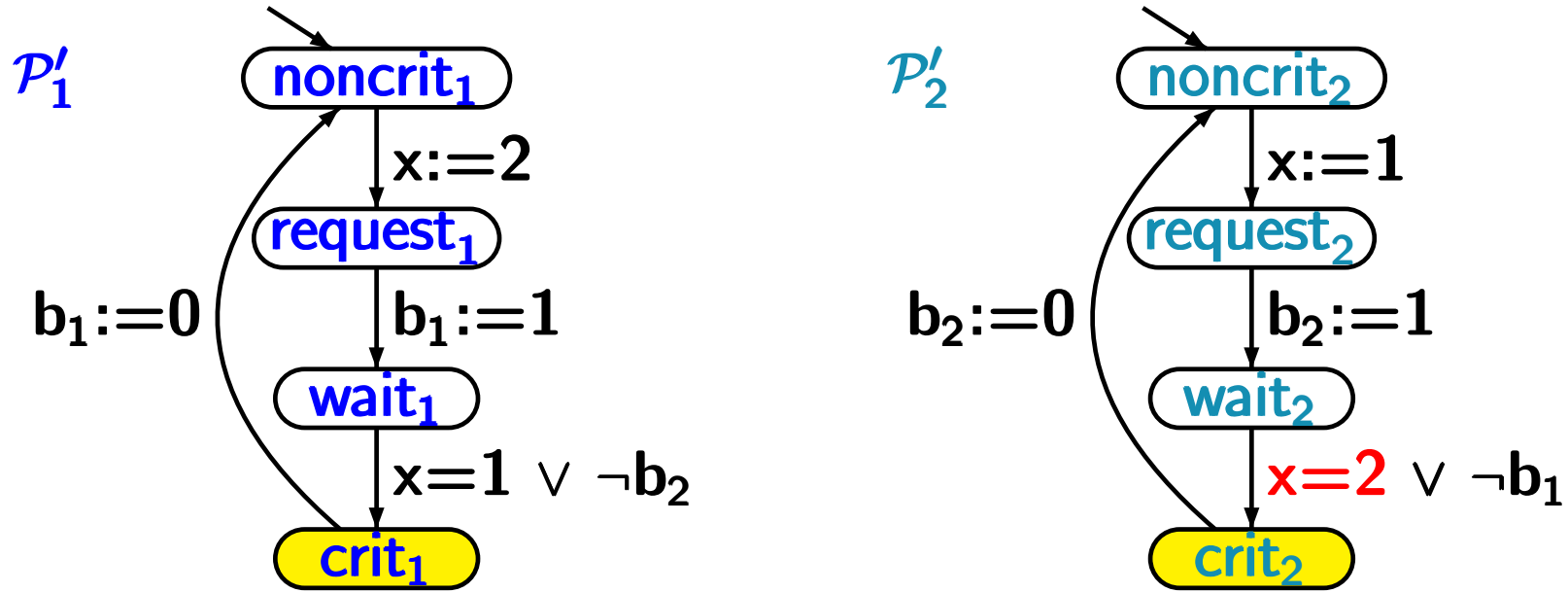
Variant of Peterson algorithm



possible executions

noncrit$_1$	noncrit$_2$	x=1	¬b$_1$	¬b$_2$
noncrit$_1$	request$_2$	x=1	¬ b$_1$	¬b$_2$
request$_1$	request$_2$	x=2	¬b$_1$	¬b$_2$
wait$_1$	request$_2$	x=2	b$_1$	¬b$_2$
crit$_1$	request$_2$	x=2	b$_1$	¬b$_2$
crit$_1$	wait$_2$	x=2	b$_1$	b$_2$

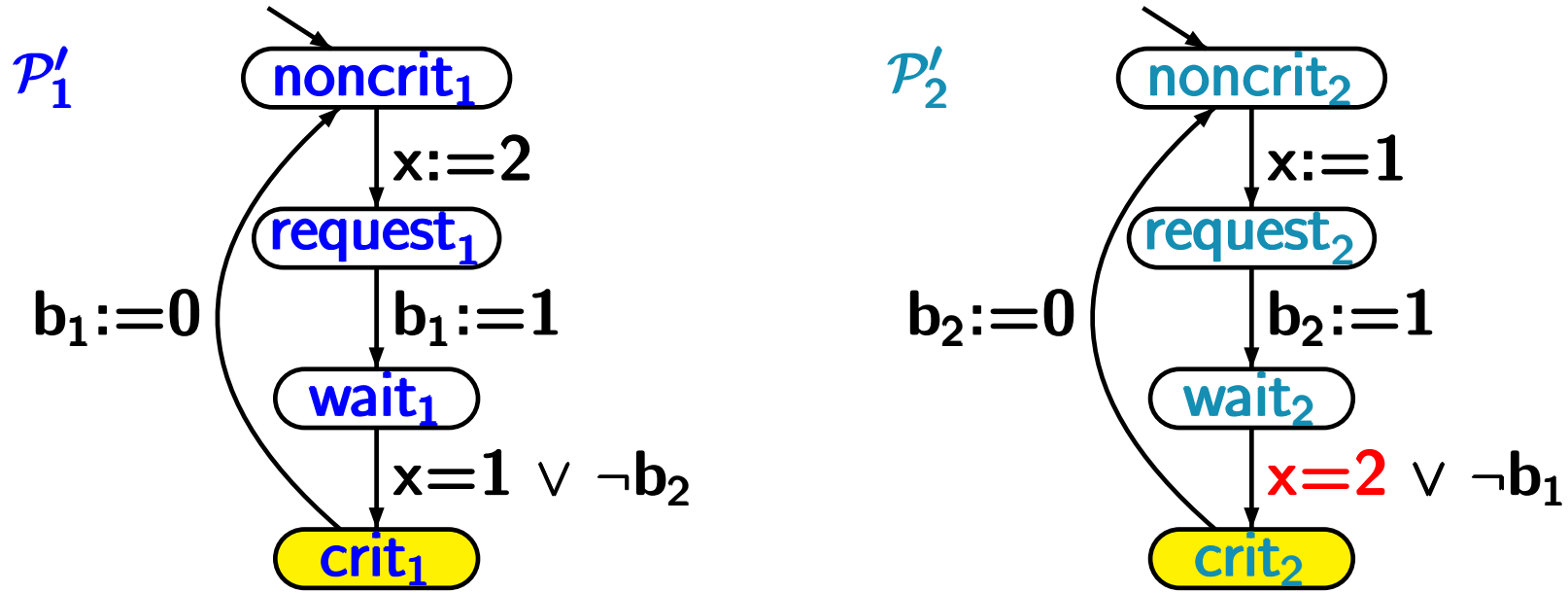
Variant of Peterson algorithm



possible executions

noncrit$_1$	noncrit$_2$	x=1	¬b$_1$	¬b$_2$
noncrit$_1$	request$_2$	x=1	¬ b$_1$	¬b$_2$
request$_1$	request$_2$	x=2	¬b$_1$	¬b$_2$
wait$_1$	request$_2$	x=2	b$_1$	¬b$_2$
crit$_1$	request$_2$	x=2	b$_1$	¬b$_2$
crit$_1$	wait$_2$	x=2	b$_1$	b$_2$
crit$_1$	crit$_2$	x=2	b$_1$	b$_2$

Variant of Peterson algorithm incorrect!



possible executions

noncrit$_1$	noncrit$_2$	x=1	¬b$_1$	¬b$_2$
noncrit$_1$	request$_2$	x=1	¬ b$_1$	¬b$_2$
request$_1$	request$_2$	x=2	¬b$_1$	¬b$_2$
wait$_1$	request$_2$	x=2	b$_1$	¬b$_2$
crit$_1$	request$_2$	x=2	b$_1$	¬b$_2$
crit$_1$	wait$_2$	x=2	b$_1$	b$_2$
crit$_1$	crit$_2$	x=2	b$_1$	b$_2$

How many states ...?

Given n processes by program graphs $\mathcal{P}_1, \ldots, \mathcal{P}_n$

How many states ...?

Given n processes by program graphs $\mathcal{P}_1, \ldots, \mathcal{P}_n$

- with 2 locations each
- over the set of variables $Var = \{x_1, \ldots, x_m\}$ with $Dom(x_i) = \{0, 1\}$

How many states has the transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \ldots \,|||\, \mathcal{P}_n}$?

How many states ...?

Given n processes by program graphs $\mathcal{P}_1, \ldots, \mathcal{P}_n$

- with 2 locations each
- over the set of variables $Var = \{x_1, \ldots, x_m\}$ with $Dom(x_i) = \{0, 1\}$

How many states has the transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \ldots \,|||\, \mathcal{P}_n}$?

answer: $2^n \cdot 2^m$

State explosion problem

Given n processes by program graphs $\mathcal{P}_1, \ldots, \mathcal{P}_n$

- with 2 locations each
- over the set of variables $Var = \{x_1, \ldots, x_m\}$ with $Dom(x_i) = \{0, 1\}$

How many states has the transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \ldots \,|||\, \mathcal{P}_n}$?

answer: $2^n \cdot 2^m$

state explosion: size of transition systems grows

- exponentially in the number of parallel processes
- exponentially in the number of variables