Model Checking I
alias
Reactive Systems Verification

Luca Tesei

MSc in Computer Science, University of Camerino

Topics

- State-based view of transition systems, Executions and Paths.
- Linear time view versus Branching time view.
- Traces of a transition system, examples.

Material

Reading:

Chapter 2 of the book, pages 20–26.
Chapter 3 of the book, pages 89–99.

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.

Overview

Overview

Introduction

Modelling parallel systems

Linear Time Properties

- state-based and linear time view ⟵
- definition of linear time properties
- invariants and safety
- liveness and fairness

Regular Properties

Linear Temporal Logic

Computation-Tree Logic

Equivalences and Abstraction

State-based view of TS

transition system $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$

State-based view of TS

transition system $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$

Act	for modeling interactions/communication
AP, L	for specifying properties

State-based view of TS

transition system $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$

Act	for modeling interactions/communication and specifying fairness assumptions
AP, L	for specifying properties

State-based view of TS

transition system $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$

$\downarrow$ abstraction from actions

state graph $G_{\mathcal{T}}$

- set of nodes = state space S
- edges = transitions without action label

Act for modeling interactions/communication and specifying fairness assumptions

AP, L for specifying properties

State-based view of TS

transition system $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$

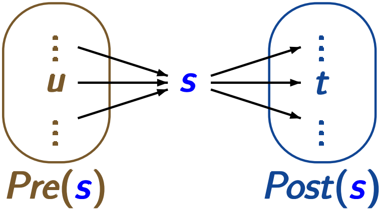
↓ abstraction from actions

state graph $G_{\mathcal{T}}$

- set of nodes = state space S
- edges = transitions without action label

use standard notations for graphs, e.g.,

$$Post(s) = \{t \in S : s \to t\}$$
$$Pre(s) = \{u \in S : u \to s\}$$



Execution fragments

execution fragment: sequence of consecutive transitions

$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$ infinite or

$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} s_n$ finite

Execution and path fragments

execution fragment: sequence of consecutive transitions

$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$ infinite or

$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} s_n$ finite

path fragment: sequence of states arising from the projection of an execution fragment to the states

Execution and path fragments

execution fragment: sequence of consecutive transitions

$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$ infinite or

$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} s_n$ finite

path fragment: sequence of states arising from the projection of an execution fragment to the states

$\pi = s_0\, s_1\, s_2 \ldots$ infinite or $\pi = s_0\, s_1 \ldots s_n$ finite

such that $s_{i+1} \in Post(s_i)$ for all $i < |\pi|$

Execution and path fragments

execution fragment: sequence of consecutive transitions

$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$ infinite or

$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} s_n$ finite

path fragment: sequence of states arising from the projection of an execution fragment to the states

$\pi = s_0\, s_1\, s_2 \ldots$ infinite or $\pi = s_0\, s_1 \ldots s_n$ finite

such that $s_{i+1} \in Post(s_i)$ for all $i < |\pi|$

initial: if $s_0 \in S_0$ = set of initial states

Execution and path fragments

execution fragment: sequence of consecutive transitions

$$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$$ infinite or

$$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} s_n$$ finite

path fragment: sequence of states arising from the projection of an execution fragment to the states

$\pi = s_0\, s_1\, s_2 \ldots$ infinite or $\pi = s_0\, s_1 \ldots s_n$ finite

such that $s_{i+1} \in Post(s_i)$ for all $i < |\pi|$

initial: if $s_0 \in S_0$ = set of initial states

maximal: if infinite or ending in a terminal state

Notations for paths

SBV2.3-2A

path fragment: sequence of states

$\pi = s_0\, s_1\, s_2 \ldots$ infinite or $\pi = s_0\, s_1 \ldots s_n$ finite

s.t. $s_{i+1} \in Post(s_i)$ for all $i < |\pi|$

initial: if $s_0 \in S_0$ = set of initial states

maximal: if infinite or ending in terminal state

path of TS $\mathcal{T}$ $\widehat{=}$ initial, maximal path fragment

Notations for paths

path fragment: sequence of states

$\pi = s_0\, s_1\, s_2 \ldots$ infinite or $\pi = s_0\, s_1 \ldots s_n$ finite

s.t. $s_{i+1} \in Post(s_i)$ for all $i < |\pi|$

initial: if $s_0 \in S_0$ = set of initial states

maximal: if infinite or ending in terminal state

path of TS $\mathcal{T}$ $\widehat{=}$ initial, maximal path fragment

path of state s $\widehat{=}$ maximal path fragment starting in state s

Notations for paths

path fragment: sequence of states

$\pi = s_0\, s_1\, s_2 \ldots$ infinite or $\pi = s_0\, s_1 \ldots s_n$ finite

s.t. $s_{i+1} \in Post(s_i)$ for all $i < |\pi|$

initial: if $s_0 \in S_0$ = set of initial states

maximal: if infinite or ending in terminal state

path of TS $\mathcal{T}$ $\widehat{=}$ initial, maximal path fragment

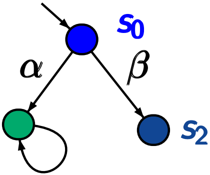
path of state s $\widehat{=}$ maximal path fragment starting in state s

$Paths(\mathcal{T})$ = set of all initial, maximal path fragments

$Paths(s)$ = set of all maximal path fragments starting in state s

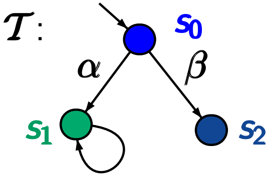
Paths of a TS

$\mathcal{T}$:



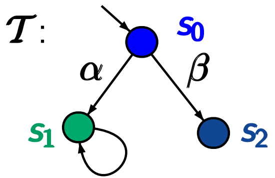
How many paths are there in $\mathcal{T}$?

Paths of a TS



How many paths are there in $\mathcal{T}$?

answer: 2, namely $s_0\, s_1\, s_1\, s_1 \ldots$ and $s_0\, s_2$



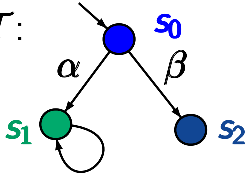
How many paths are there in $\mathcal{T}$?

answer: 2, namely $s_0\, s_1\, s_1\, s_1 \ldots$ and $s_0\, s_2$

$$
\begin{aligned}
\mathit{Paths}(s_1) &= \text{set of all maximal paths fragments starting in } s_1 \\
&= \{s_1^\omega\} \text{ where } s_1^\omega = s_1\, s_1\, s_1\, s_1\, \ldots
\end{aligned}
$$

Paths of a TS and its states

$\mathcal{T}$:



How many paths are there in $\mathcal{T}$?

answer: 2, namely $s_0\, s_1\, s_1\, s_1 \ldots$ and $s_0\, s_2$

$Paths(s_1)$	$=$	set of all maximal paths fragments starting in s_1
	$=$	$\{s_1^\omega\}$ where $s_1^\omega = s_1\, s_1\, s_1\, s_1\, \ldots$
$Paths_{fin}(s_1)$	$=$	set of all finite path fragments starting in s_1
	$=$	$\{s_1^n : n \in \mathbb{N}, n \geq 1\}$

Overview

Introduction

Modelling parallel systems

Linear Time Properties

- state-based and linear time view ⟵
- definition of linear time properties
- invariants and safety
- liveness and fairness

Regular Properties

Linear Temporal Logic

Computation-Tree Logic

Equivalences and Abstraction

Introduction

Modelling parallel systems

Linear Time Properties

- state-based and linear time view ⟵
- definition of linear time properties
- invariants and safety
- liveness and fairness

Regular Properties

Linear Temporal Logic

Computation-Tree Logic

Equivalences and Abstraction

Linear-time vs branching-time

Linear-time vs branching-time

transition system

$$\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$$

Linear-time vs branching-time

transition system

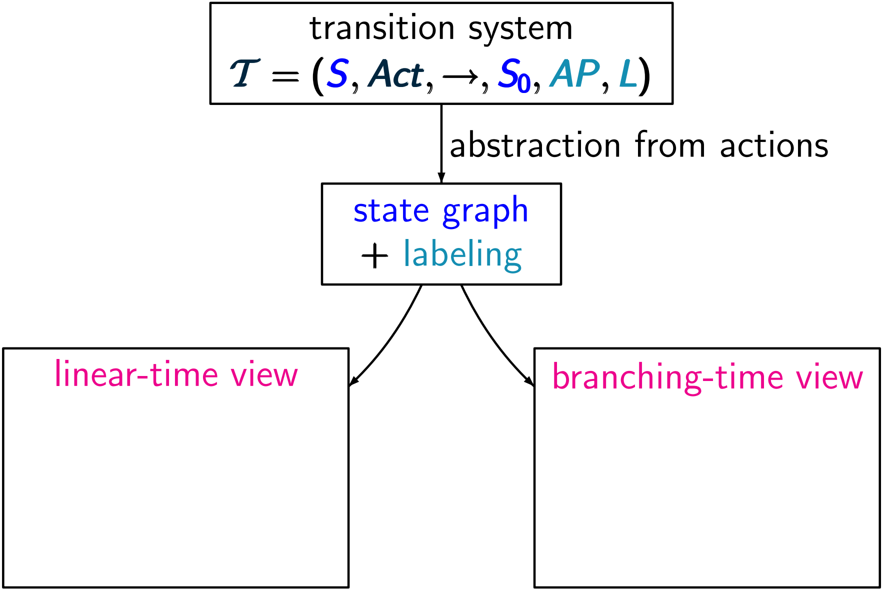
$\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$

↓ abstraction from actions

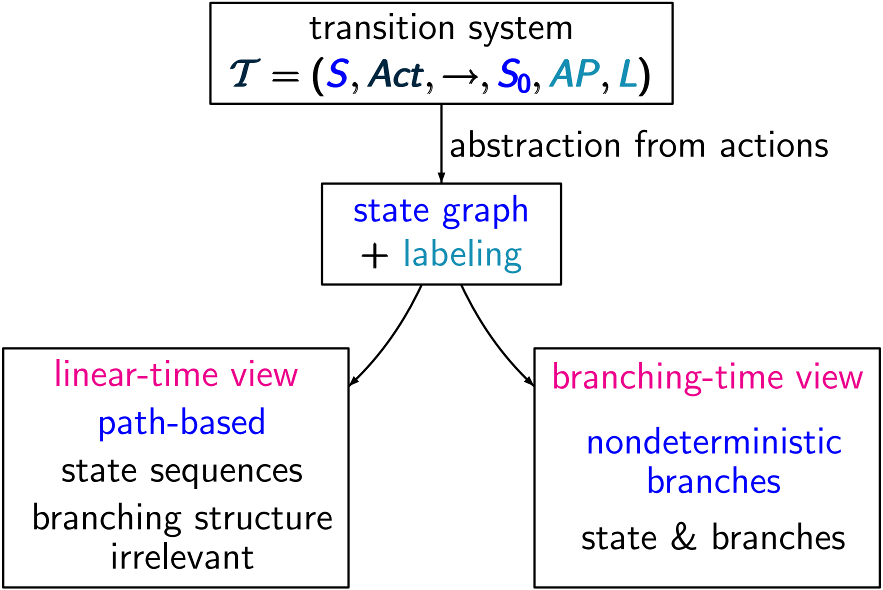
state graph

+ labeling

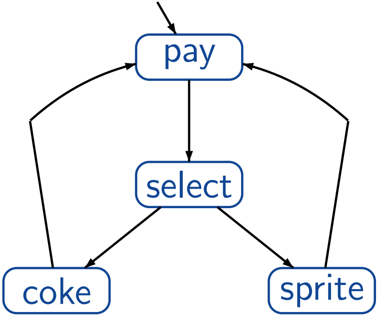
Linear-time vs branching-time



Linear-time vs branching-time

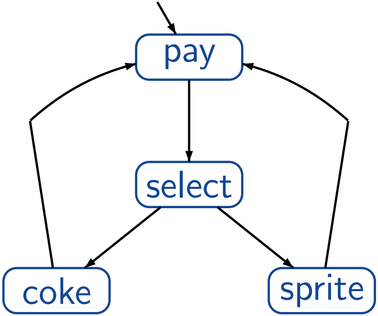


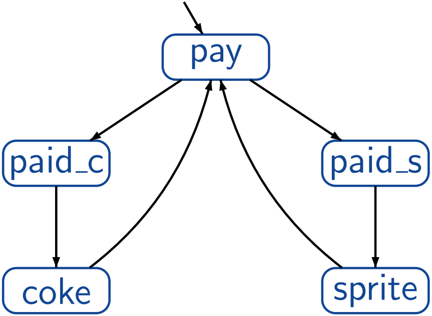
Example: vending machine



vending machine with
1 coin deposit
select drink after
having paid

Example: vending machine

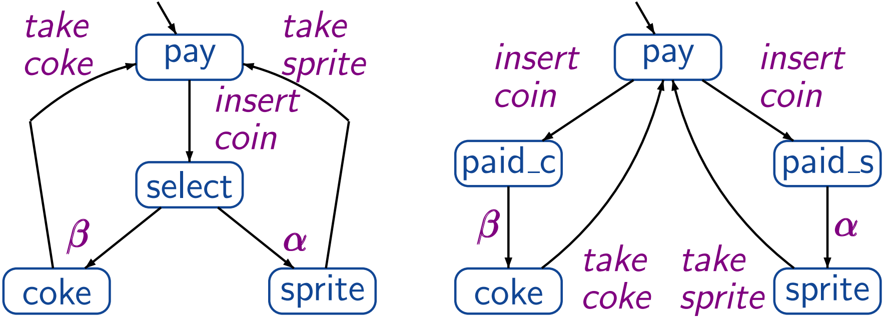




vending machine with
1 coin deposit
select drink after having paid

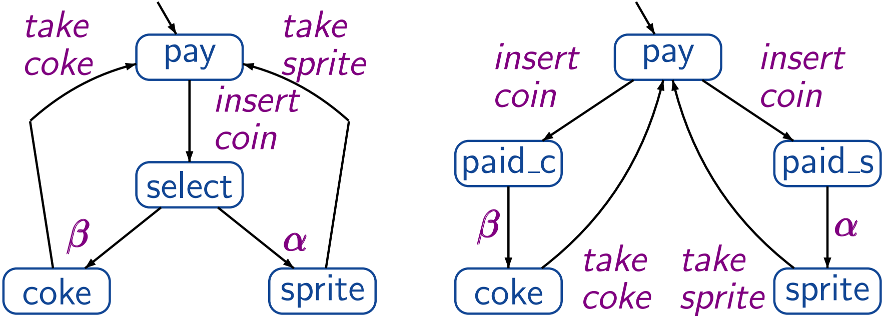
vending machine with
2 coin deposits
select drink by inserting the coin

Example: vending machine



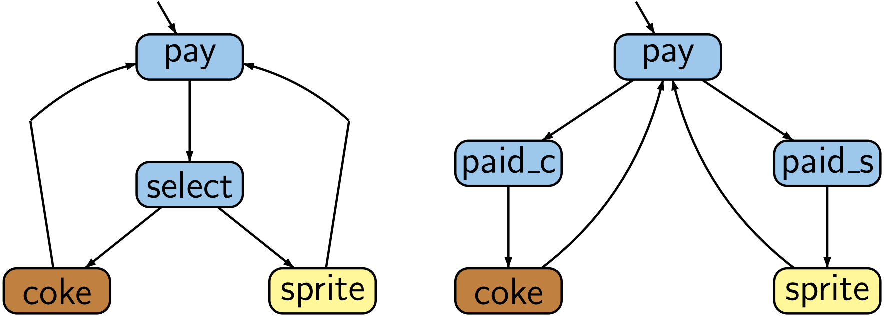
vending machine with
1 coin deposit
select drink after
having paid

vending machine with
2 coin deposits
select drink by inserting
the coin



state based view: abstracts from actions and projects onto atomic propositions, e.g. $AP = \{coke, sprite\}$

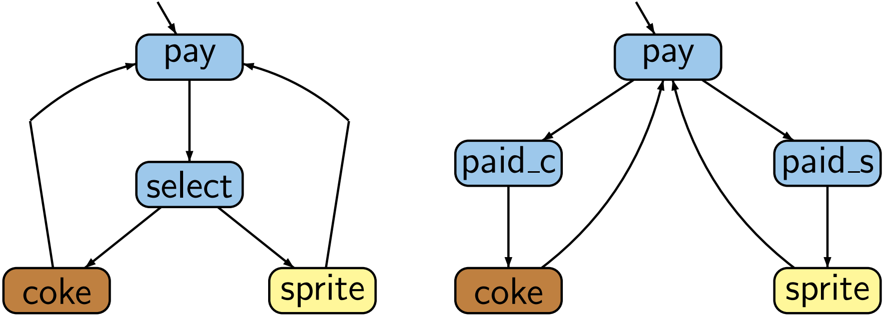
Example: vending machine



state based view: abstracts from actions and projects onto atomic propositions, e.g. $AP = \{coke, sprite\}$

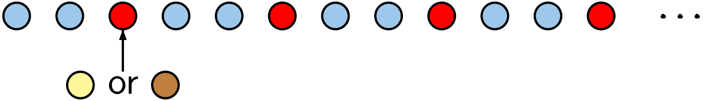
e.g., $L(\text{coke}) = \{coke\}$, $L(\text{pay}) = \varnothing$

Example: vending machine

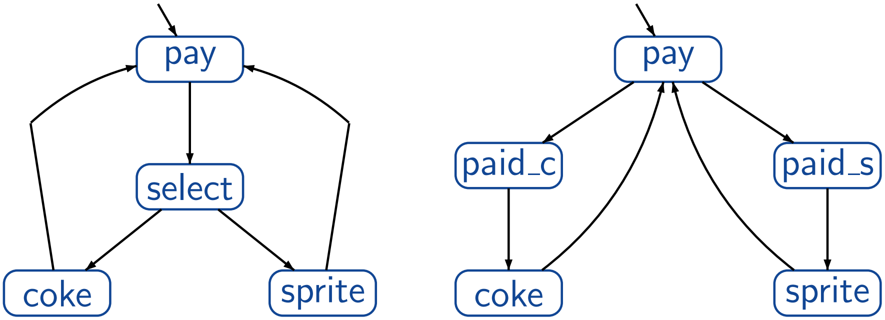


state based view: abstracts from actions and projects onto atomic propositions, e.g. $AP = \{coke, sprite\}$

linear time: all observable behaviors are of the form

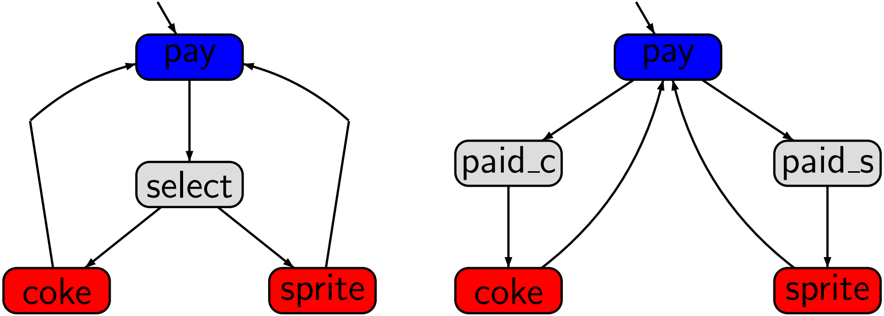


Example: vending machine



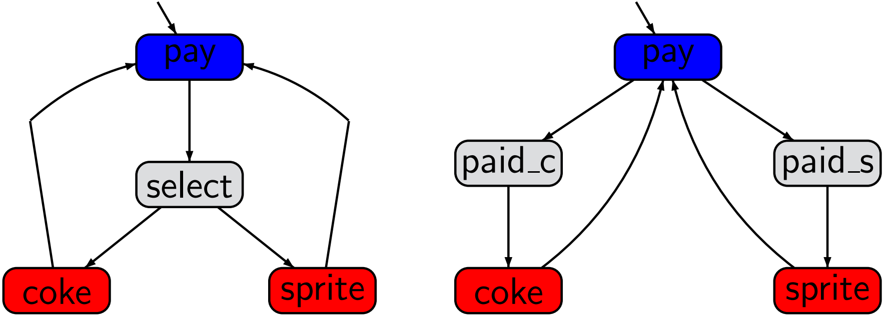
state based view: abstracts from actions and projects on atomc propositions, e.g., $AP = \{pay, drink\}$

Example: vending machine



state based view: abstracts from actions and projects on atomc propositions, e.g., $AP = \{pay, drink\}$

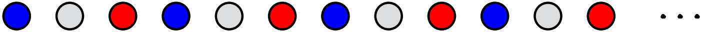
Example: vending machine



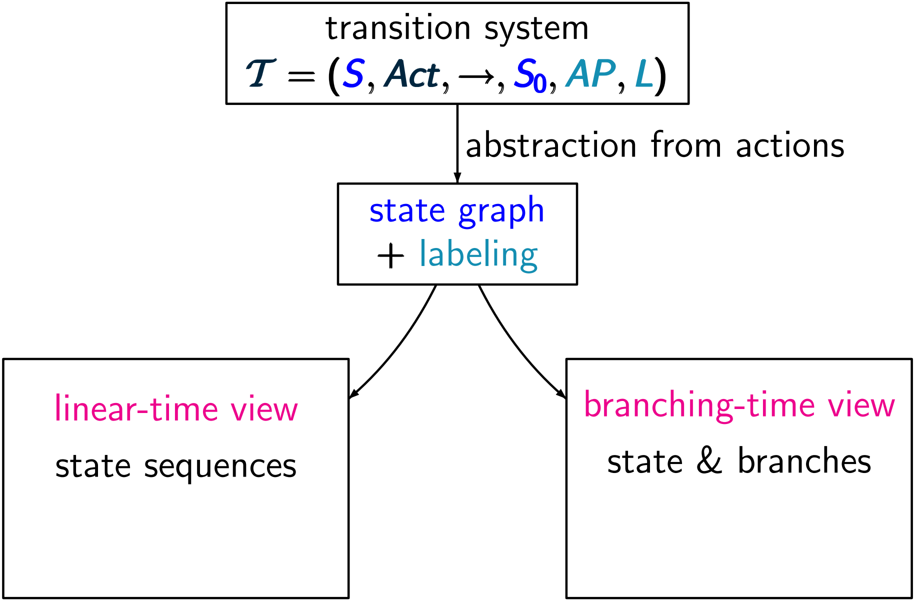
state based view: abstracts from actions and projects on atomc propositions, e.g., $AP = \{pay, drink\}$

linear & branching time:

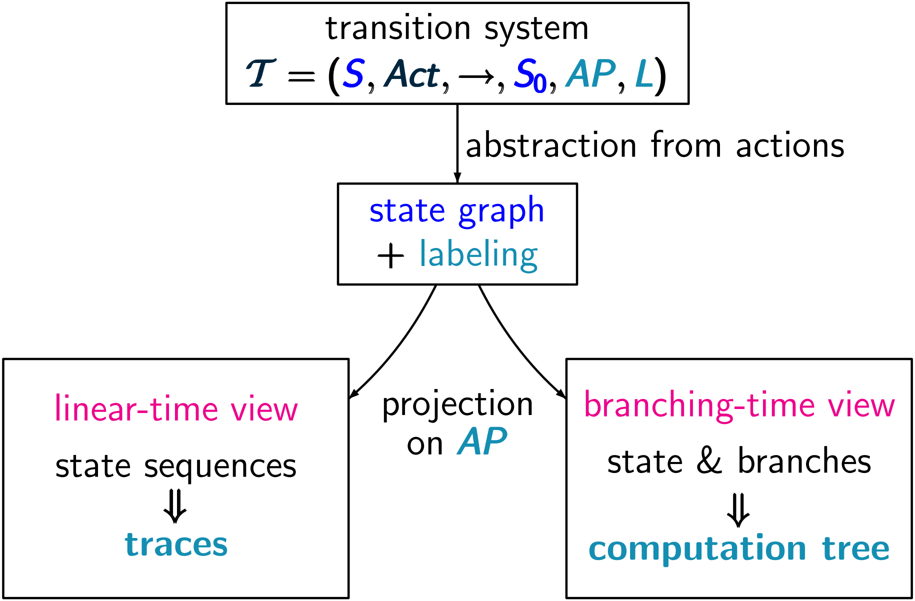
all observable behaviors have the form



Linear-time vs branching-time



Linear-time vs branching-time



Traces

LTB2.4-4

Traces

for TS with labeling function $L : S \rightarrow 2^{AP}$

execution: states + actions

$s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$ infinite or finite

↓

paths: sequences of states

$s_0\, s_1\, s_2 \ldots$ infinite or $s_0\, s_1 \ldots s_n$ finite

Traces

for TS with labeling function $L : S \to 2^{AP}$

execution: states + actions

$s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$ infinite or finite

↓

paths: sequences of states

$s_0\, s_1\, s_2 \ldots$ infinite or $s_0\, s_1 \ldots s_n$ finite

↓

traces: sequences of sets of atomic propositions

$L(s_0)\, L(s_1)\, L(s_2)\, \ldots$

Traces

for TS with labeling function $L : S \to 2^{AP}$

execution: states + actions

$s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$ infinite or finite

paths: sequences of states

$s_0\, s_1\, s_2 \ldots$ infinite or $s_0\, s_1 \ldots s_n$ finite

traces: sequences of sets of atomic propositions

$L(s_0)\, L(s_1)\, L(s_2)\, \ldots \quad \in (2^{AP})^\omega \cup (2^{AP})^+$

for TS with labeling function $L : S \rightarrow 2^{AP}$

execution: states + actions

$s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$ infinite or finite

paths: sequences of states

$s_0\, s_1\, s_2 \ldots$ infinite or $s_0\, s_1 \ldots s_n$ finite

traces: sequences of sets of atomic propositions

$L(s_0)\, L(s_1)\, L(s_2)\, \ldots \quad \in (2^{AP})^\omega \cup (2^{AP})^+$

for simplicity: we often assume that the given TS has **no terminal states**

Traces

for TS with labeling function $L : S \to 2^{AP}$

execution: states + actions

$s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$ infinite or ~~finite~~

↓

paths: sequences of states

$s_0\, s_1\, s_2 \ldots$ infinite or ~~$s_0\, s_1 \ldots s_n$ finite~~

↓

traces: sequences of sets of atomic propositions

$L(s_0)\, L(s_1)\, L(s_2)\, \ldots \in (2^{AP})^\omega \cup$ ~~$(2^{AP})^+$~~

for simplicity: we often assume that the given TS has **no terminal states**

Treatment of terminal states

perform standard graph algorithms to compute the reachable fragment of the given TS

$$Reach(\mathcal{T}) = \left\{ \begin{array}{c} \text{set of states that are reachable} \\ \text{from some initial state} \end{array} \right.$$

Treatment of terminal states

perform standard graph algorithms to compute the reachable fragment of the given TS

$$Reach(\mathcal{T}) = \left\{ \begin{array}{c} \text{set of states that are reachable} \\ \text{from some initial state} \end{array} \right.$$

for each reachable terminal state s:

- if s stands for an intended halting configuration then add a transition from s to a trap state:

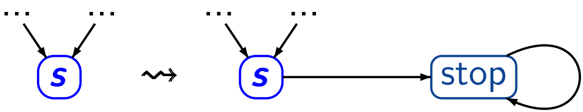
Treatment of terminal states

perform standard graph algorithms to compute the reachable fragment of the given TS

$$\mathit{Reach}(\mathcal{T}) = \left\{ \begin{array}{c} \text{set of states that are reachable} \\ \text{from some initial state} \end{array} \right.$$

for each reachable terminal state s:

- if s stands for an intended halting configuration then add a transition from s to a trap state:



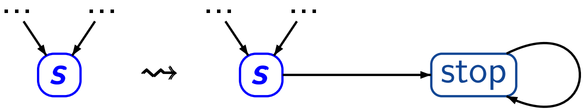
Treatment of terminal states

perform standard graph algorithms to compute the reachable fragment of the given TS

$$\textit{Reach}(\mathcal{T}) = \left\{ \begin{array}{c} \text{set of states that are reachable} \\ \text{from some initial state} \end{array} \right.$$

for each reachable terminal state s:

- if s stands for an intended halting configuration then add a transition from s to a trap state:



- if s stands for system fault, e.g., deadlock then correct the design before checking further properties

Traces of a transition system

Let $\mathcal{T}$ be a TS

$$\mathit{Traces}(\mathcal{T}) \stackrel{\text{def}}{=} \{\mathit{trace}(\pi) : \pi \in \mathit{Paths}(\mathcal{T})\}$$

$$\mathit{Traces}_{\mathit{fin}}(\mathcal{T}) \stackrel{\text{def}}{=} \{\mathit{trace}(\widehat{\pi}) : \widehat{\pi} \in \mathit{Paths}_{\mathit{fin}}(\mathcal{T})\}$$

Traces of a transition system

Let $\mathcal{T}$ be a TS

$$Traces(\mathcal{T}) \stackrel{\text{def}}{=} \{trace(\pi) : \pi \in Paths(\mathcal{T})\}$$

↑ initial, maximal path fragment

$$Traces_{fin}(\mathcal{T}) \stackrel{\text{def}}{=} \{trace(\widehat{\pi}) : \widehat{\pi} \in Paths_{fin}(\mathcal{T})\}$$

↑ initial, finite path fragment

Traces of a transition system

Let $\mathcal{T}$ be a TS ⟵ *without* terminal states

$$Traces(\mathcal{T}) \stackrel{\text{def}}{=} \{trace(\pi) : \pi \in Paths(\mathcal{T})\}$$

↑ initial, infinite path fragment

$$Traces_{fin}(\mathcal{T}) \stackrel{\text{def}}{=} \{trace(\widehat{\pi}) : \widehat{\pi} \in Paths_{fin}(\mathcal{T})\}$$

↑ initial, finite path fragment

Traces of a transition system

Let $\mathcal{T}$ be a TS ⟵ *without* terminal states

$$Traces(\mathcal{T}) \stackrel{\text{def}}{=} \{trace(\pi) : \pi \in Paths(\mathcal{T})\} \subseteq (2^{AP})^{\omega}$$

↑ initial, infinite path fragment

$$Traces_{fin}(\mathcal{T}) \stackrel{\text{def}}{=} \{trace(\widehat{\pi}) : \widehat{\pi} \in Paths_{fin}(\mathcal{T})\} \subseteq (2^{AP})^{*}$$

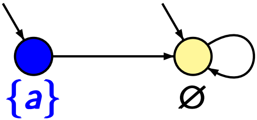
↑ initial, finite path fragment

Example: traces

Let $\mathcal{T}$ be a TS without terminal states.

$$Traces(\mathcal{T}) \stackrel{\text{def}}{=} \{trace(\pi) : \pi \in Paths(\mathcal{T})\} \subseteq (2^{AP})^{\omega}$$

$$Traces_{fin}(\mathcal{T}) \stackrel{\text{def}}{=} \{trace(\widehat{\pi}) : \widehat{\pi} \in Paths_{fin}(\mathcal{T})\} \subseteq (2^{AP})^{*}$$

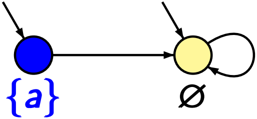


TS $\mathcal{T}$ with a single atomic proposition a

Let $\mathcal{T}$ be a TS without terminal states.

$$Traces(\mathcal{T}) \stackrel{\text{def}}{=} \{ trace(\pi) : \pi \in Paths(\mathcal{T}) \} \subseteq (2^{AP})^{\omega}$$

$$Traces_{fin}(\mathcal{T}) \stackrel{\text{def}}{=} \{ trace(\widehat{\pi}) : \widehat{\pi} \in Paths_{fin}(\mathcal{T}) \} \subseteq (2^{AP})^{*}$$

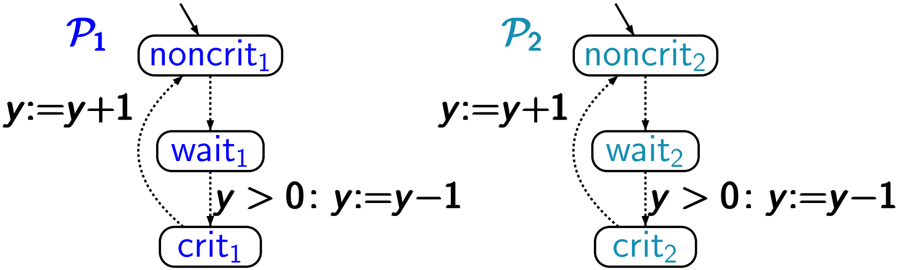


TS $\mathcal{T}$ with a single atomic proposition a

$$Traces(\mathcal{T}) = \{ \{a\}\varnothing^{\omega}, \varnothing^{\omega} \}$$

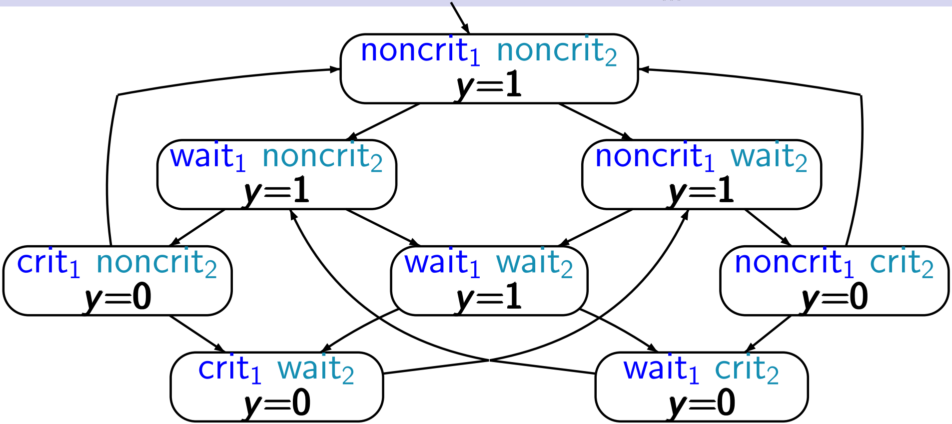
$$Traces_{fin}(\mathcal{T}) = \{\{a\}\varnothing^{n} : n \geq 0\} \cup \{\varnothing^{m} : m \geq 1\}$$

Mutual exclusion with semaphore



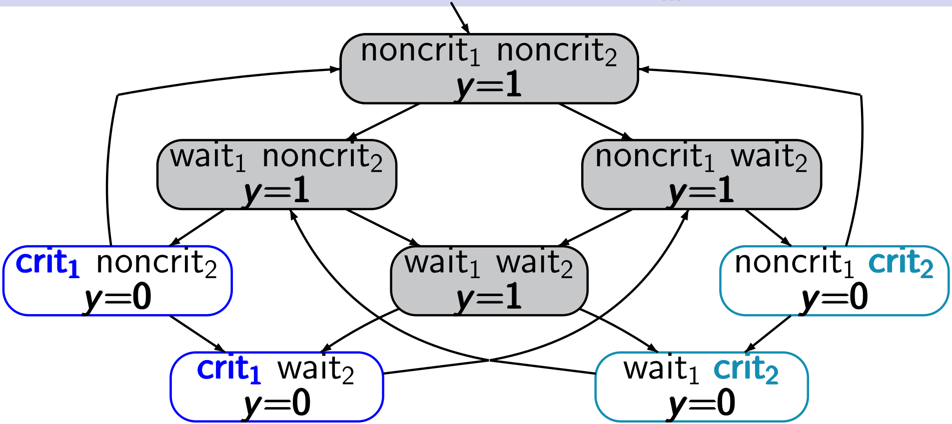
transition system $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$ arises by unfolding the composite program graph $\mathcal{P}_1 ||| \mathcal{P}_2$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$



set of atomic propositions $AP = \{\mathbf{crit_1}, \mathbf{crit_2}\}$

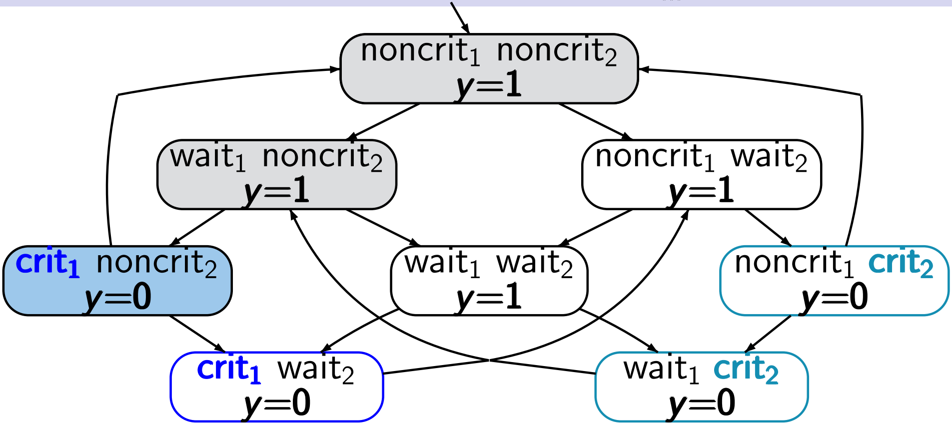
Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$



set of atomic propositions $AP = \{\mathsf{crit}_1, \mathsf{crit}_2\}$

e.g., $L(\langle \mathsf{noncrit}_1, \mathsf{noncrit}_2, y{=}1 \rangle) =$
$L(\langle \mathsf{wait}_1, \mathsf{noncrit}_2, y{=}1 \rangle) = \varnothing$

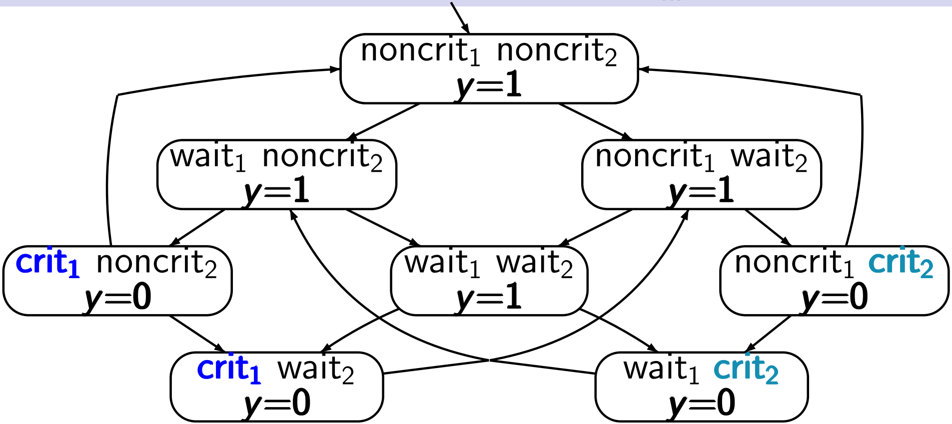
Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$



set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

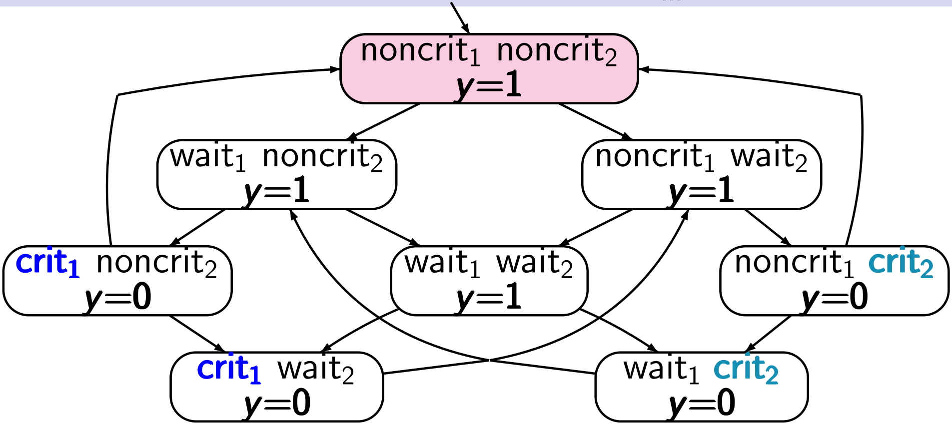


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

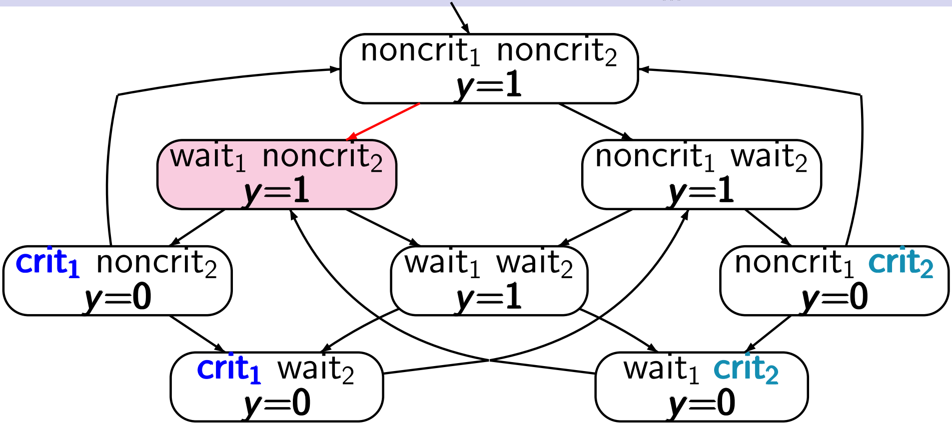


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

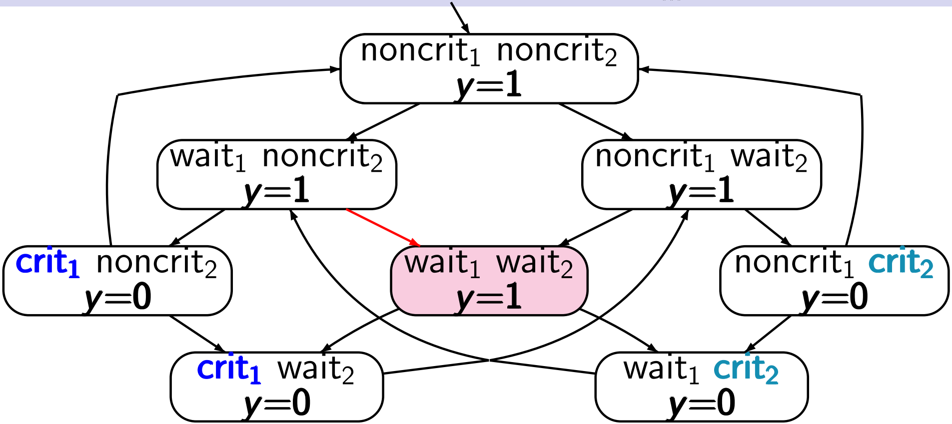


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

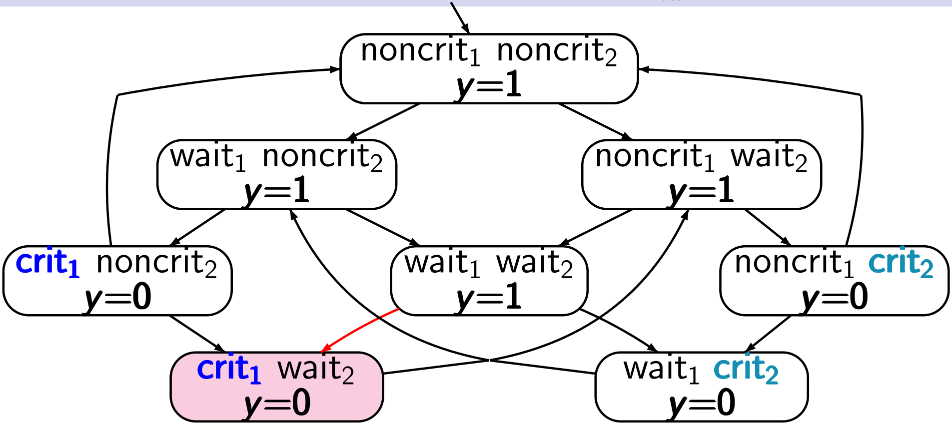


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1|||\mathcal{P}_2}$

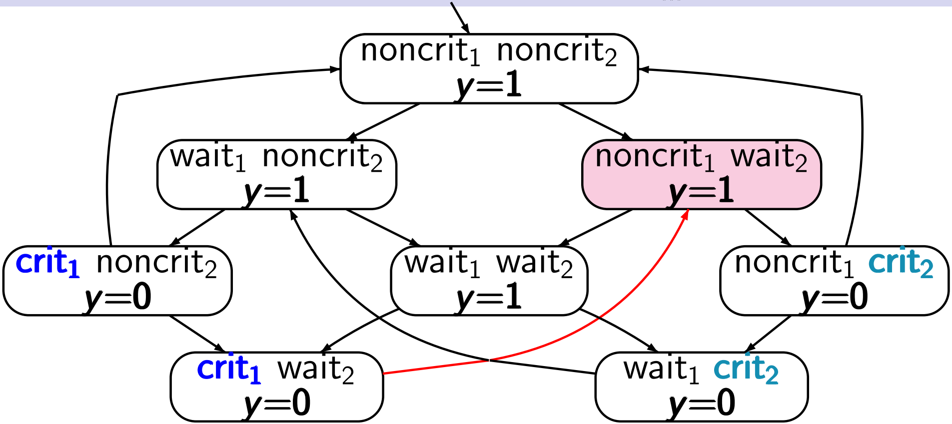


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

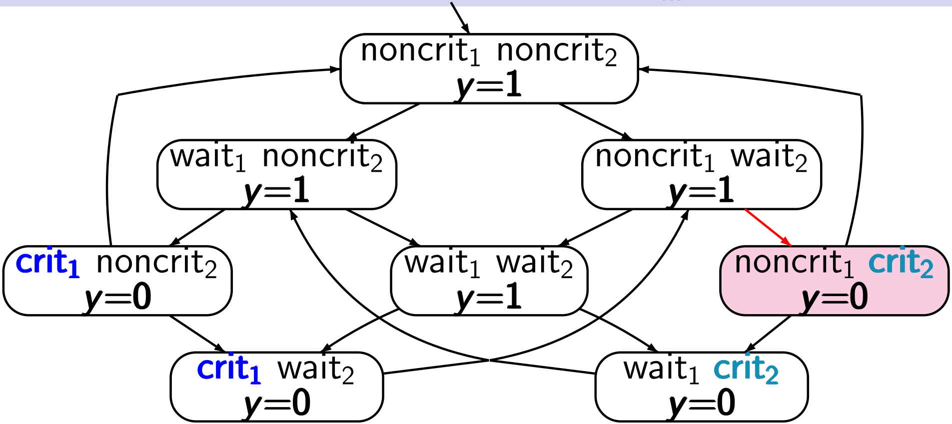


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

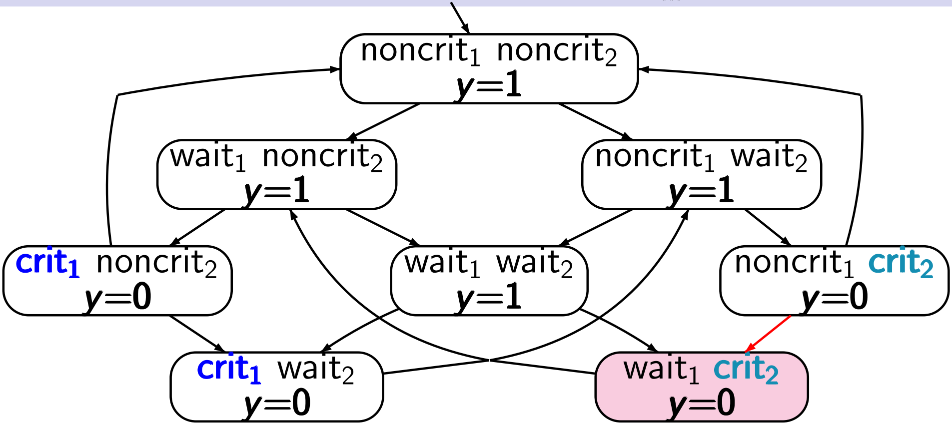


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

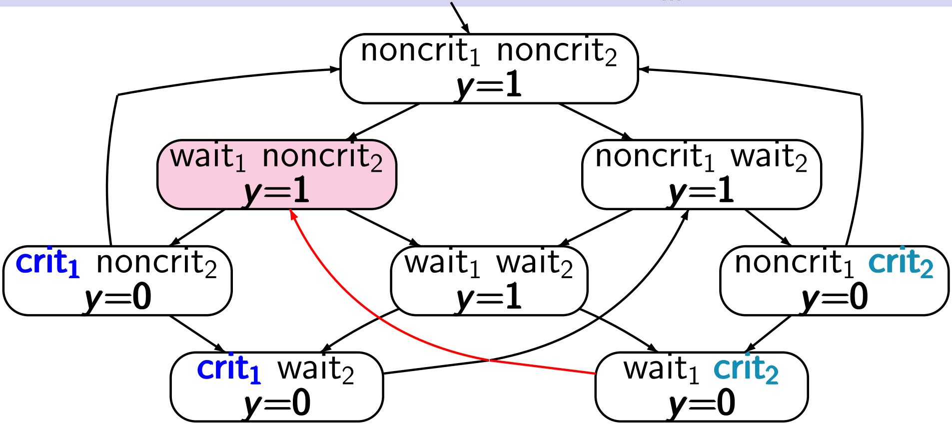


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphore $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

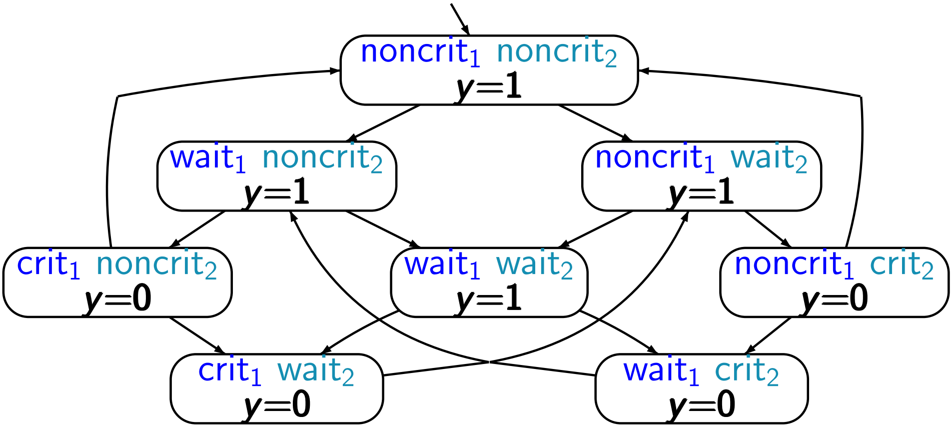


set of atomic propositions $AP = \{crit_1, crit_2\}$

traces, e.g., $\varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \varnothing\ \{crit_1\} \ldots$

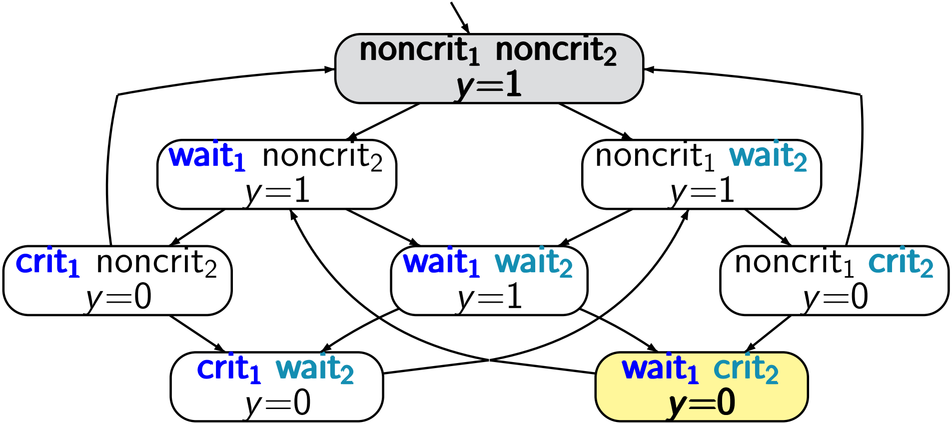
$\varnothing\ \varnothing\ \varnothing\ \{crit_1\}\ \varnothing\ \{crit_2\}\ \{crit_2\}\ \varnothing \ldots$

Mutual exclusion with semaphor $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$



set of propositions $AP = \{\mathbf{wait_1}, \mathbf{crit_1}, \mathbf{wait_2}, \mathbf{crit_2}\}$

Mutual exclusion with semaphor $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

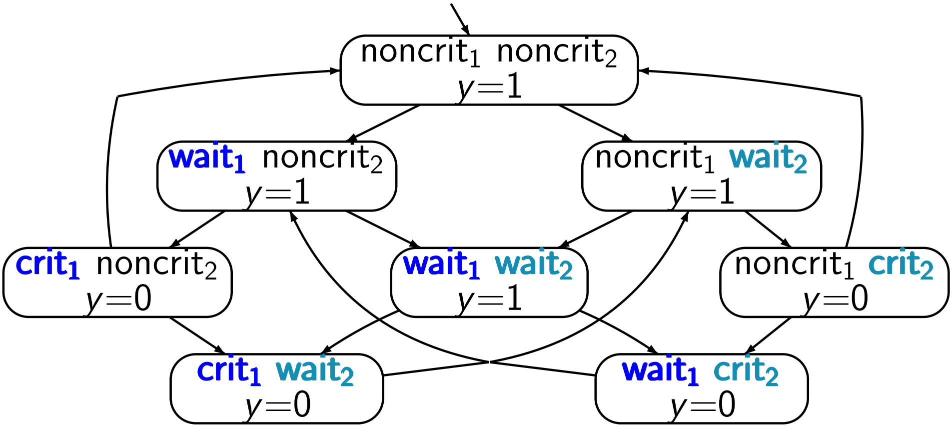


set of propositions $AP = \{wait_1, crit_1, wait_2, crit_2\}$

e.g., $L(\langle noncrit_1, noncrit_2, y{=}1\rangle) = \varnothing$

$L(\langle wait_1, crit_2, y{=}1\rangle) = \{wait_1, crit_2\}$

Mutual exclusion with semaphor $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$

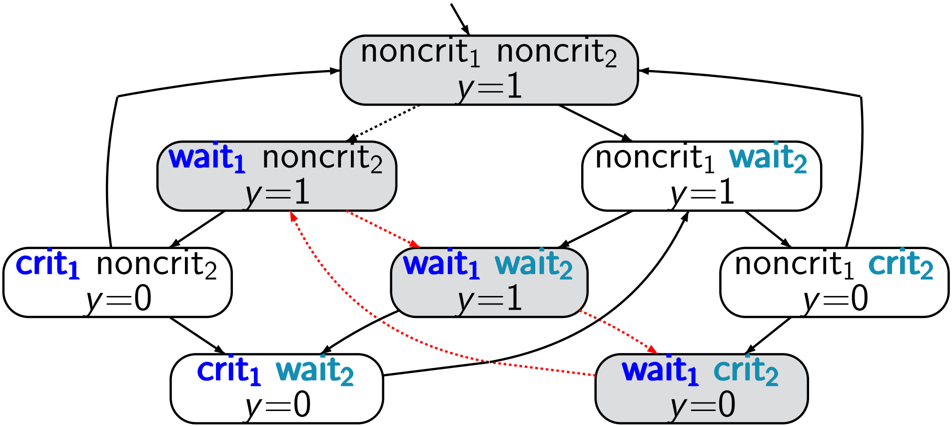


set of propositions $AP = \{\text{wait}_1, \text{crit}_1, \text{wait}_2, \text{crit}_2\}$

traces, e.g.,

$$\varnothing \left(\{\text{wait}_1\}\{\text{wait}_1, \text{wait}_2\}\{\text{wait}_1, \text{crit}_2\}\right)^\omega$$

Mutual exclusion with semaphor $\mathcal{T}_{\mathcal{P}_1 ||| \mathcal{P}_2}$



set of propositions $AP = \{\text{wait}_1, \text{crit}_1, \text{wait}_2, \text{crit}_2\}$

traces, e.g.,

$$\varnothing \left(\{\text{wait}_1\}\,\{\text{wait}_1, \text{wait}_2\}\,\{\text{wait}_1, \text{crit}_2\}\right)^{\omega}$$