Introduction

Modelling parallel systems

Linear Time Properties

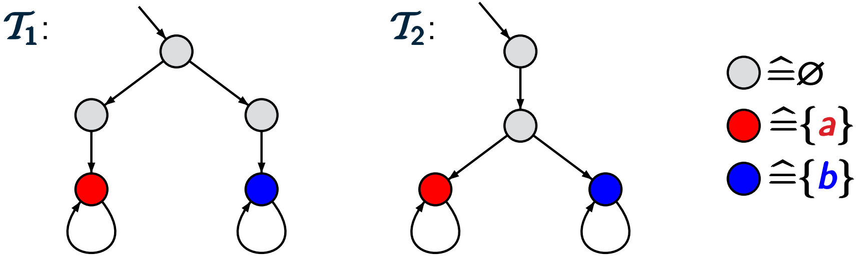
Regular Properties

Linear Temporal Logic (LTL)

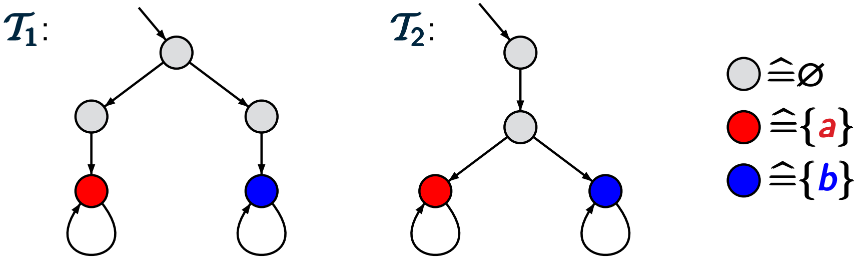
Computation-Tree Logic

Equivalences and Abstraction

Trace equivalence



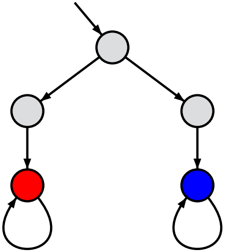
Trace equivalence



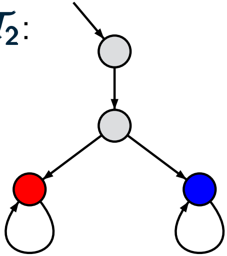
$$Traces(\mathcal{T}_1) = \{ \varnothing \varnothing a^\omega, \ \varnothing \varnothing b^\omega \} = Traces(\mathcal{T}_1)$$

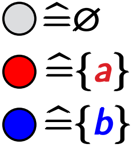
Trace equivalence

$\mathcal{T}_1$:



$\mathcal{T}_2$:

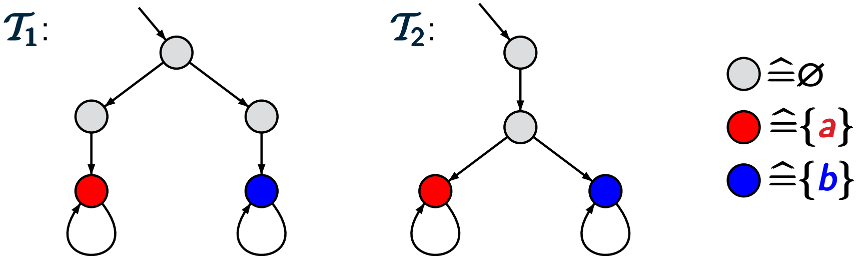




$$Traces(\mathcal{T}_1) = \{ \varnothing \varnothing a^\omega, \ \varnothing \varnothing b^\omega \} = Traces(\mathcal{T}_1)$$

CTL-formula $\Phi = \exists \bigcirc (\exists \bigcirc a \wedge \exists \bigcirc b)$

Trace equivalence

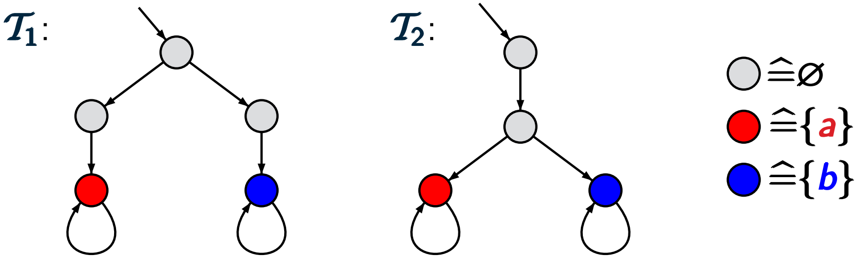


$$Traces(\mathcal{T}_1) = \{ \varnothing\,\varnothing\, a^\omega,\ \varnothing\,\varnothing\, b^\omega \} = Traces(\mathcal{T}_1)$$

CTL-formula $\Phi = \exists\bigcirc(\exists\bigcirc a \wedge \exists\bigcirc b)$

$\mathcal{T}_1 \not\models \Phi$ and $\mathcal{T}_2 \models \Phi$

Trace equivalence is not compatible with CTL



$$Traces(\mathcal{T}_1) = \{\ \varnothing\,\varnothing\,a^\omega,\ \varnothing\,\varnothing\,b^\omega\ \} = Traces(\mathcal{T}_1)$$

CTL-formula $\Phi = \exists\bigcirc(\exists\bigcirc a \wedge \exists\bigcirc b)$

$\mathcal{T}_1 \not\models \Phi$ and $\mathcal{T}_2 \models \Phi$

Implementation relations

- for the design of complex systems
  - $\leadsto$ comparison of **2** transition systems

Implementation relations

- for the design of complex systems
  $\rightsquigarrow$ comparison of **2** transition systems

- for the analysis of complex systems

Implementation relations

- for the design of complex systems
  - $\rightsquigarrow$ comparison of $\mathbf{2}$ transition systems
- for the analysis of complex systems
  - $\rightsquigarrow$ homogeneous model checking approach

Implementation relations

- for the design of complex systems
  - $\leadsto$ comparison of **2** transition systems
- for the analysis of complex systems
  - $\leadsto$ homogeneous model checking approach
  - $\leadsto$ graph minimization

Implementation relations

- for the design of complex systems
  - $\rightsquigarrow$ comparison of **2** transition systems
- for the analysis of complex systems
  - $\rightsquigarrow$ homogeneous model checking approach
  - $\rightsquigarrow$ graph minimization

use equivalence relation $\sim$ for the states of a single transition system $\mathcal{T}$ and analyze the quotient $\mathcal{T}/\sim$

Implementation relations

- for the design of complex systems
  - $\leadsto$ comparison of **2** transition systems
- for the analysis of complex systems
  - $\leadsto$ homogeneous model checking approach
  - $\leadsto$ graph minimization

use equivalence relation $\sim$ for the states of a single transition system $\mathcal{T}$ and analyze the quotient $\mathcal{T}/\!\sim$

goal: define the equivalence $\sim$ in such a way that

$$\mathcal{T} \models \Phi \quad \text{iff} \quad \mathcal{T}/\!\sim\, \models \Phi$$

for all "relevant" properties Φ

Linear-time implementation relations

Linear-time implementation relations

finite trace inclusion and equivalence:

e.g., $\mathit{Tracesfin}(\mathcal{T}_1) \subseteq \mathit{Tracesfin}(\mathcal{T}_2)$

trace inclusion and trace equivalence:

e.g., $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$

Linear-time implementation relations

finite trace inclusion and equivalence:

e.g., $\mathit{Tracesfin}(\mathcal{T}_1) \subseteq \mathit{Tracesfin}(\mathcal{T}_2)$

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

e.g., $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$

Linear-time implementation relations

finite trace inclusion and equivalence:

e.g., $\mathit{Tracesfin}(\mathcal{T}_1) \subseteq \mathit{Tracesfin}(\mathcal{T}_2)$

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

e.g., $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$

preserves all **LTL** properties

finite trace inclusion and equivalence:

e.g., $\mathit{Tracesfin}(\mathcal{T}_1) \subseteq \mathit{Tracesfin}(\mathcal{T}_2)$

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

e.g., $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$

preserves all **LTL** properties

* none of the LT relations is compatible with **CTL**

Linear-time implementation relations

finite trace inclusion and equivalence:

e.g., $\mathit{Tracesfin}(\mathcal{T}_1) \subseteq \mathit{Tracesfin}(\mathcal{T}_2)$

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

e.g., $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$

preserves all **LTL** properties

* none of the LT relations is compatible with **CTL**
* checking LT relations is computationally hard

finite trace inclusion and equivalence:

e.g., $\mathit{Tracesfin}(\mathcal{T}_1) \subseteq \mathit{Tracesfin}(\mathcal{T}_2)$

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

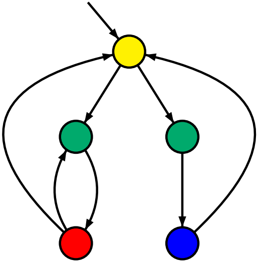
e.g., $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$

preserves all **LTL** properties

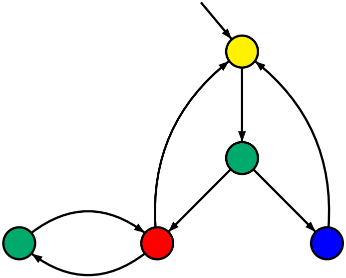
- none of the LT relations is compatible with **CTL**
- checking LT relations is computationally hard
- **minimization ???**

Minimization w.r.t. trace equivalence?

$\mathcal{T}_1$:

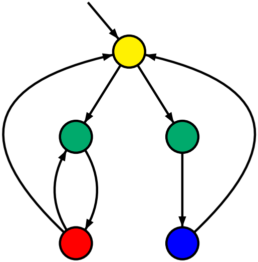


$\mathcal{T}_2$:

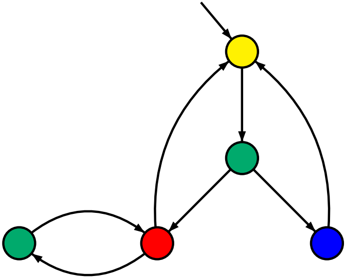


Minimization w.r.t. trace equivalence?

$\mathcal{T}_1$:



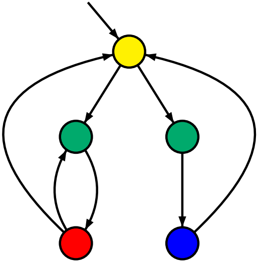
$\mathcal{T}_2$:



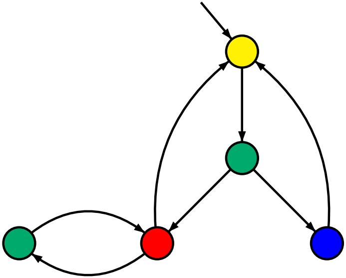
- $Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$

Minimization w.r.t. trace equivalence?

$\mathcal{T}_1$:



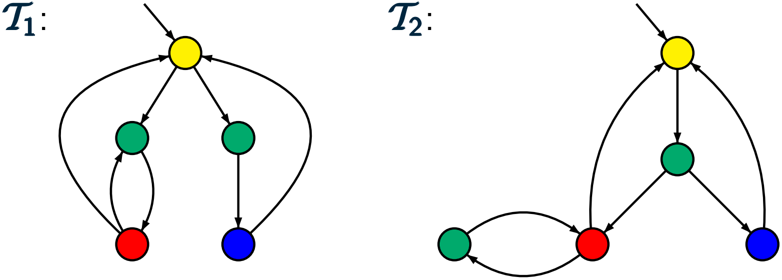
$\mathcal{T}_2$:



- $\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$

  but $\mathcal{T}_1$ and $\mathcal{T}_2$ are not isomorphic

Minimization w.r.t. trace equivalence?



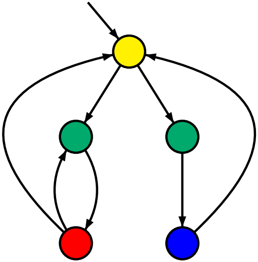
- $\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$

  but $\mathcal{T}_1$ and $\mathcal{T}_2$ are not isomorphic

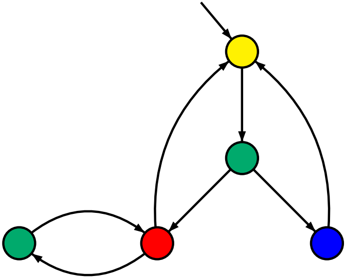
- $\mathcal{T}_1$, $\mathcal{T}_2$ have **5** states and **7** transitions each

Minimization w.r.t. trace equivalence?

$\mathcal{T}_1$:



$\mathcal{T}_2$:



- $Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$

  but $\mathcal{T}_1$ and $\mathcal{T}_2$ are not isomorphic
- $\mathcal{T}_1$, $\mathcal{T}_2$ have **5** states and **7** transitions each
- there is no smaller TS that is trace-equivalent to $\mathcal{T}_i$

Classification of implementation relations

Classification of implementation relations

- linear vs. branching time
  - linear time: trace relations
  - branching time: (bi)simulation relations

Classification of implementation relations

- linear vs. branching time
  - linear time: trace relations
  - branching time: (bi)simulation relations
- (nonsymmetric) preorders vs. equivalences:
  - preorders: trace inclusion, simulation
  - equivalences: trace equivalence, bisimulation

Classification of implementation relations

- linear vs. branching time
  - linear time: trace relations
  - branching time: (bi)simulation relations
- (nonsymmetric) preorders vs. equivalences:
  - preorders: trace inclusion, simulation
  - equivalences: trace equivalence, bisimulation
- strong vs. weak relations
  - strong: reasoning about all transitions
  - weak: abstraction from stutter steps

Overview

Introduction
Modelling parallel systems
Linear Time Properties
Regular Properties
Linear Temporal Logic (LTL)
Computation-Tree Logic
Equivalences and Abstraction
- bisimulation ⟵
- CTL, CTL*-equivalence
- computing the bisimulation quotient
- abstraction stutter steps
- simulation relations

Bisimulation for two transition systems

Bisimulation for two transition systems

let $\mathcal{T}_1 = (S_1, Act_1, \rightarrow_1, S_{0,1}, AP, L_1)$,

$\mathcal{T}_2 = (S_2, Act_2, \rightarrow_2, S_{0,2}, AP, L_2)$

be two transition systems

Bisimulation for two transition systems

let $\mathcal{T}_1 = (S_1, Act_1, \rightarrow_1, S_{0,1}, AP, L_1)$,

$\mathcal{T}_2 = (S_2, Act_2, \rightarrow_2, S_{0,2}, AP, L_2)$

be two transition systems

- with the same set AP

Bisimulation for two transition systems

let
$$\mathcal{T}_1 = (S_1, Act_1, \rightarrow_1, S_{0,1}, AP, L_1),$$
$$\mathcal{T}_2 = (S_2, Act_2, \rightarrow_2, S_{0,2}, AP, L_2)$$

be two transition systems

- with the same set AP
- possibly containing terminal states

let $\mathcal{T}_1 = (S_1, Act_1, \rightarrow_1, S_{0,1}, AP, L_1)$,
$\mathcal{T}_2 = (S_2, Act_2, \rightarrow_2, S_{0,2}, AP, L_2)$

be two transition systems

- with the same set AP
- possibly containing terminal states

Bisimulation equivalence of $\mathcal{T}_1$ and $\mathcal{T}_2$ requires that $\mathcal{T}_1$ and $\mathcal{T}_2$ can simulate each other in a stepwise manner.

let $\mathcal{T}_1 = (S_1, \cancel{Act_1}, \rightarrow_1, S_{0,1}, AP, L_1)$,

$\mathcal{T}_2 = (S_2, \cancel{Act_2}, \rightarrow_2, S_{0,2}, AP, L_2)$

be two transition systems

- with the same set AP $\longleftarrow$ $\boxed{\text{observables}}$
- possibly containing terminal states

Bisimulation equivalence of $\mathcal{T}_1$ and $\mathcal{T}_2$ requires that $\mathcal{T}_1$ and $\mathcal{T}_2$ can simulate each other in a stepwise manner.

Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$

Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$

binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$

binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L_1(s_1) = L_2(s_2)$

Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$

binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L_1(s_1) = L_2(s_2)$

(2) $\forall s_1' \in Post(s_1) \, \exists s_2' \in Post(s_2)$ s.t. $(s_1', s_2') \in \mathcal{R}$

Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$

binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L_1(s_1) = L_2(s_2)$

(2) $\forall s_1' \in Post(s_1)\, \exists s_2' \in Post(s_2)$ s.t. $(s_1', s_2') \in \mathcal{R}$

$$\boxed{\begin{tikzcd} s_1 \arrow[r, phantom, "-\mathcal{R}-"] \arrow[d] & s_2 \\ s_1' & \end{tikzcd}} \quad \text{can be completed to} \quad \boxed{\begin{tikzcd} s_1 \arrow[r, phantom, "-\mathcal{R}-"] \arrow[d] & s_2 \arrow[d] \\ s_1' \arrow[r, phantom, "-\mathcal{R}-"] & s_2' \end{tikzcd}}$$

Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$

binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L_1(s_1) = L_2(s_2)$

(2) $\forall s_1' \in Post(s_1) \, \exists s_2' \in Post(s_2)$ s.t. $(s_1', s_2') \in \mathcal{R}$

$$\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \\ s_1' & & \end{array} \qquad \text{can be completed to} \qquad \begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \downarrow \\ s_1' & -\mathcal{R}- & s_2' \end{array}$$

(3) $\forall s_2' \in Post(s_2) \, \exists s_1' \in Post(s_1)$ s.t. $(s_1', s_2') \in \mathcal{R}$

Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$

binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L_1(s_1) = L_2(s_2)$

(2) $\forall s_1' \in Post(s_1) \, \exists s_2' \in Post(s_2)$ s.t. $(s_1', s_2') \in \mathcal{R}$

$$\boxed{\begin{tikzcd} s_1 \arrow[r, phantom, "-\mathcal{R}-"] \arrow[d] & s_2 \\ s_1' & \end{tikzcd}} \quad \text{can be completed to} \quad \boxed{\begin{tikzcd} s_1 \arrow[r, phantom, "-\mathcal{R}-"] \arrow[d] & s_2 \arrow[d] \\ s_1' \arrow[r, phantom, "-\mathcal{R}-"] & s_2' \end{tikzcd}}$$

(3) $\forall s_2' \in Post(s_2) \, \exists s_1' \in Post(s_1)$ s.t. $(s_1', s_2') \in \mathcal{R}$

and such that the following initial condition holds:

(I) $\forall s_{0,1} \in S_{0,1} \, \exists s_{0,2} \in S_{0,2}$ s.t. $(s_{0,1}, s_{0,2}) \in \mathcal{R}$

Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$

binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L_1(s_1) = L_2(s_2)$

(2) $\forall s_1' \in Post(s_1)\, \exists s_2' \in Post(s_2)$ s.t. $(s_1', s_2') \in \mathcal{R}$

$$\boxed{\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \\ s_1' & & \end{array}} \quad \text{can be completed to} \quad \boxed{\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \downarrow \\ s_1' & -\mathcal{R}- & s_2' \end{array}}$$

(3) $\forall s_2' \in Post(s_2)\, \exists s_1' \in Post(s_1)$ s.t. $(s_1', s_2') \in \mathcal{R}$

and such that the following initial condition holds:

(I) $\forall s_{0,1} \in S_{0,1}\, \exists s_{0,2} \in S_{0,2}$ s.t. $(s_{0,1}, s_{0,2}) \in \mathcal{R}$

$\forall s_{0,2} \in S_{0,2}\, \exists s_{0,1} \in S_{0,1}$ s.t. $(s_{0,1}, s_{0,2}) \in \mathcal{R}$

Bisimulation equivalence $\sim$

bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$: relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

for all $(s_1, s_2) \in \mathcal{R}$: (1) labeling condition

$\left.\begin{array}{l}(2)\\(3)\end{array}\right\}$ mutual stepwise simulation

and initial condition (I)

bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$: relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

for all $(s_1, s_2) \in \mathcal{R}$: (1) labeling condition

(2) (3) $\left.\right\}$ mutual stepwise simulation

and initial condition (I)

bisimulation equivalence $\sim$ for TS:

Bisimulation equivalence $\sim$

bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$: relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

for all $(s_1, s_2) \in \mathcal{R}$: (1) labeling condition

(2), (3) $\Big\}$ mutual stepwise simulation

and initial condition (I)

> bisimulation equivalence $\sim$ for TS:
>
> $\mathcal{T}_1 \sim \mathcal{T}_2$ iff there is a bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$

bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$: relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

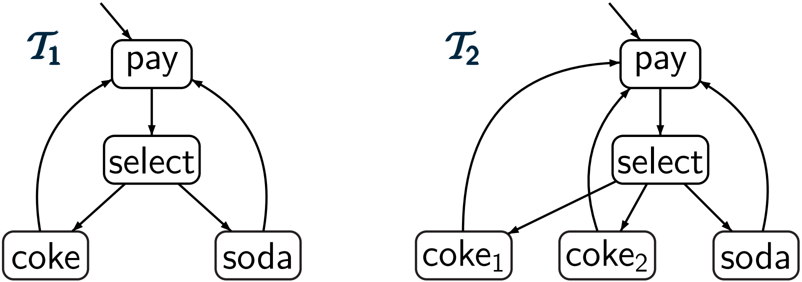
for all $(s_1, s_2) \in \mathcal{R}$: (1) labeling condition

(2), (3) $\Big\}$ mutual stepwise simulation

and initial condition (I)

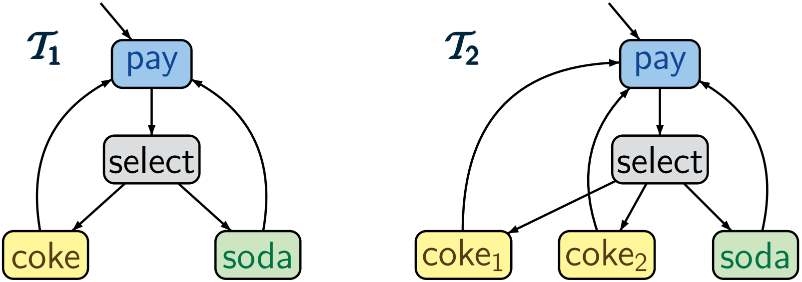
> bisimulation equivalence $\sim$ for TS:
>
> $\mathcal{T}_1 \sim \mathcal{T}_2$ iff there is a bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$

> for state s_1 of $\mathcal{T}_1$ and state s_2 of $\mathcal{T}_2$:
>
> $s_1 \sim s_2$ iff there exists a bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$ such that $(s_1, s_2) \in \mathcal{R}$



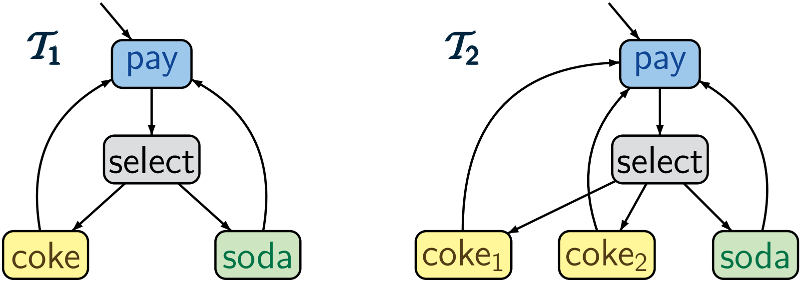
$$AP = \{pay, coke, soda\}$$

Two beverage machines



$AP = \{pay, coke, soda\}$

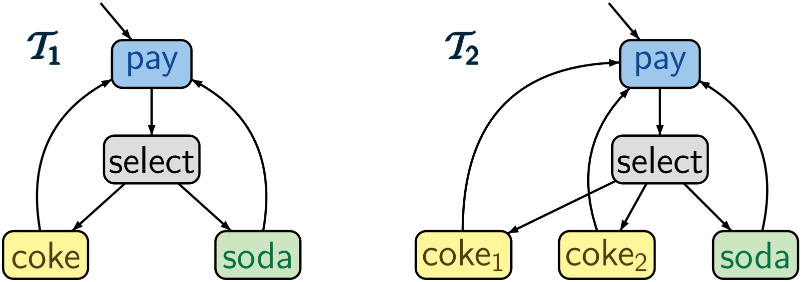
Two beverage machines



$AP = \{pay, coke, soda\}$

$\mathcal{T}_1 \sim \mathcal{T}_2$

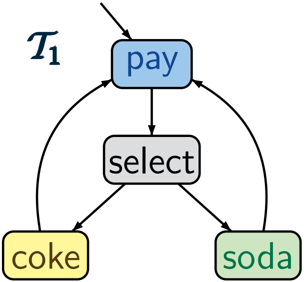
Two beverage machines

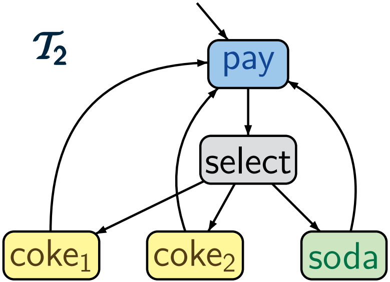


$AP = \{pay, coke, soda\}$

$\mathcal{T}_1 \sim \mathcal{T}_2$ as there is a bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$:

Two beverage machines



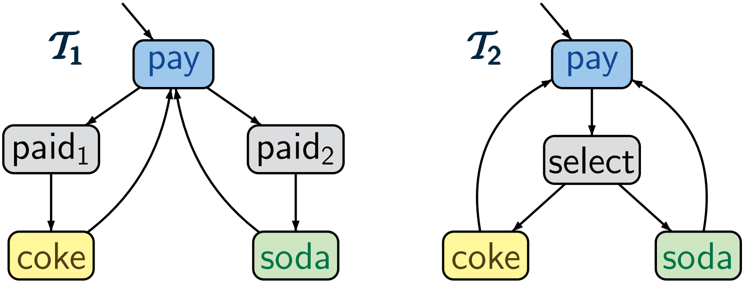


$AP = \{pay, coke, soda\}$

$\mathcal{T}_1 \sim \mathcal{T}_2$ as there is a bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$:

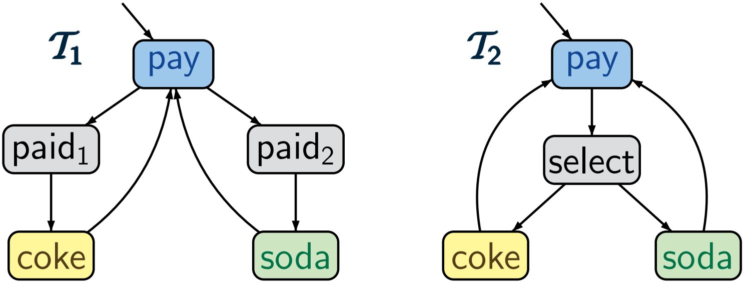
$$\{ \text{(pay,pay), (select,select), (soda,soda)}, (\text{coke},\text{coke}_1), (\text{coke},\text{coke}_2) \}$$

Two beverage machines



$$AP = \{pay, coke, soda\}$$

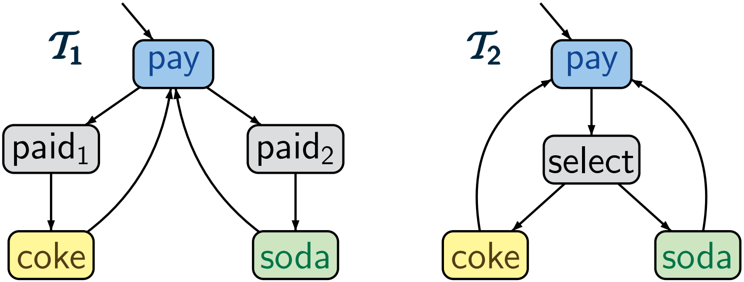
Two beverage machines



$AP = \{pay, coke, soda\}$

$\mathcal{T}_1 \not\sim \mathcal{T}_2$

Two beverage machines



$AP = \{pay, coke, soda\}$

$\mathcal{T}_1 \not\sim \mathcal{T}_2$

because there is no state in $\mathcal{T}_1$ that has both

- a successor labeled with coke and
- a successor labeled with soda

Simulation condition of bisimulations

$$\begin{tikzcd} s_1 \arrow[r, dash, "\mathcal{R}"] \arrow[d] & s_2 \\ s_1' & \end{tikzcd}$$

can be completed to

$$\begin{tikzcd} s_1 \arrow[r, dash, "\mathcal{R}"] \arrow[d] & s_2 \arrow[d] \\ s_1' \arrow[r, dash, "\mathcal{R}"] & s_2' \end{tikzcd}$$

Path lifting for bisimulation $\mathcal{R}$

$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \\
s_{1,1} & & \\
\downarrow & & \\
s_{1,2} & & \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
s_{1,4} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

Path lifting for bisimulation $\mathcal{R}$

$$
\begin{tikzcd}
s_1 \arrow[d] \arrow[r, phantom, "-\mathcal{R}-"] & s_2 \\
s_{1,1} \arrow[d] & \\
s_{1,2} \arrow[d] & \\
s_{1,3} \arrow[d] & \\
s_{1,4} \arrow[d] & \\
\vdots &
\end{tikzcd}
$$

can be
completed to

Path lifting for bisimulation $\mathcal{R}$

BSEQOR5.1-9-BIS

$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \\
s_{1,1} & & \\
\downarrow & & \\
s_{1,2} & & \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
s_{1,4} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

can be completed to

$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \downarrow \\
s_{1,1} & -\mathcal{R}- & s_{2,1} \\
\downarrow & & \\
s_{1,2} & & \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
s_{1,4} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

Path lifting for bisimulation $\mathcal{R}$

$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \\
s_{1,1} & & \\
\downarrow & & \\
s_{1,2} & & \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
s_{1,4} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

can be completed to

$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \downarrow \\
s_{1,1} & -\mathcal{R}- & s_{2,1} \\
\downarrow & & \downarrow \\
s_{1,2} & -\mathcal{R}- & s_{2,2} \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
s_{1,4} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

Path lifting for bisimulation $\mathcal{R}$

$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \\
s_{1,1} & & \\
\downarrow & & \\
s_{1,2} & & \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
s_{1,4} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

can be completed to

$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \downarrow \\
s_{1,1} & -\mathcal{R}- & s_{2,1} \\
\downarrow & & \downarrow \\
s_{1,2} & -\mathcal{R}- & s_{2,2} \\
\downarrow & & \downarrow \\
s_{1,3} & -\mathcal{R}- & s_{2,3} \\
\downarrow & & \\
s_{1,4} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

Path lifting for bisimulation $\mathcal{R}$

BSEQOR5.1-9-BIS

$$
\boxed{\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \\
s_{1,1} & & \\
\downarrow & & \\
s_{1,2} & & \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
s_{1,4} & & \\
\downarrow & & \\
\vdots & &
\end{array}}
\quad \text{can be completed to} \quad
\boxed{\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \downarrow \\
s_{1,1} & -\mathcal{R}- & s_{2,1} \\
\downarrow & & \downarrow \\
s_{1,2} & -\mathcal{R}- & s_{2,2} \\
\downarrow & & \downarrow \\
s_{1,3} & -\mathcal{R}- & s_{2,3} \\
\downarrow & & \downarrow \\
s_{1,4} & -\mathcal{R}- & s_{2,4} \\
\downarrow & & \downarrow \\
\vdots & & \vdots
\end{array}}
$$

Properties of bisimulation equivalence

Properties of bisimulation equivalence

$\sim$ is an equivalence

Properties of bisimulation equivalence

$\sim$ is an equivalence, i.e.,

- reflexivity: $\mathcal{T} \sim \mathcal{T}$ for all transition systems $\mathcal{T}$

Properties of bisimulation equivalence

$\sim$ is an equivalence, i.e.,

- reflexivity: $\mathcal{T} \sim \mathcal{T}$ for all transition systems $\mathcal{T}$

$\uparrow$

If S is the state space of $\mathcal{T}$ then

$$\mathcal{R} = \{(s,s) : s \in S\}$$

is a bisimulation for $(\mathcal{T}, \mathcal{T})$

Properties of bisimulation equivalence

$\sim$ is an equivalence, i.e.,

- reflexivity: $\mathcal{T} \sim \mathcal{T}$ for all transition systems $\mathcal{T}$
- symmetry: $\mathcal{T}_1 \sim \mathcal{T}_2$ implies $\mathcal{T}_2 \sim \mathcal{T}_1$

Properties of bisimulation equivalence

$\sim$ is an equivalence, i.e.,

- reflexivity: $\mathcal{T} \sim \mathcal{T}$ for all transition systems $\mathcal{T}$
- symmetry: $\mathcal{T}_1 \sim \mathcal{T}_2$ implies $\mathcal{T}_2 \sim \mathcal{T}_1$

If $\mathcal{R}$ is a bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$ then

$$\mathcal{R}^{-1} = \{(s_2, s_1) : (s_1, s_2) \in \mathcal{R}\}$$

is a bisimulation for $(\mathcal{T}_2, \mathcal{T}_1)$

Properties of bisimulation equivalence

$\sim$ is an equivalence, i.e.,

- reflexivity: $\mathcal{T} \sim \mathcal{T}$ for all transition systems $\mathcal{T}$
- symmetry: $\mathcal{T}_1 \sim \mathcal{T}_2$ implies $\mathcal{T}_2 \sim \mathcal{T}_1$
- transitivity: if $\mathcal{T}_1 \sim \mathcal{T}_2$ and $\mathcal{T}_2 \sim \mathcal{T}_3$ then $\mathcal{T}_1 \sim \mathcal{T}_3$

$\sim$ is an equivalence, i.e.,

- reflexivity: $\mathcal{T} \sim \mathcal{T}$ for all transition systems $\mathcal{T}$
- symmetry: $\mathcal{T}_1 \sim \mathcal{T}_2$ implies $\mathcal{T}_2 \sim \mathcal{T}_1$
- transitivity: if $\mathcal{T}_1 \sim \mathcal{T}_2$ and $\mathcal{T}_2 \sim \mathcal{T}_3$ then $\mathcal{T}_1 \sim \mathcal{T}_3$

Let $\mathcal{R}_{1,2}$ be a bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$,
$\mathcal{R}_{2,3}$ be a bisimulation for $(\mathcal{T}_2, \mathcal{T}_3)$.

Properties of bisimulation equivalence

$\sim$ is an equivalence, i.e.,

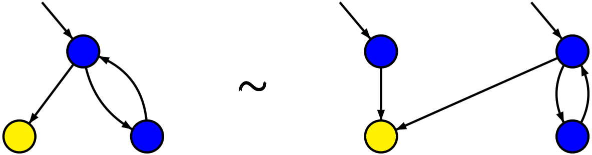
- reflexivity: $\mathcal{T} \sim \mathcal{T}$ for all transition systems $\mathcal{T}$
- symmetry: $\mathcal{T}_1 \sim \mathcal{T}_2$ implies $\mathcal{T}_2 \sim \mathcal{T}_1$
- transitivity: if $\mathcal{T}_1 \sim \mathcal{T}_2$ and $\mathcal{T}_2 \sim \mathcal{T}_3$ then $\mathcal{T}_1 \sim \mathcal{T}_3$

Let $\mathcal{R}_{1,2}$ be a bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$,
$\mathcal{R}_{2,3}$ be a bisimulation for $(\mathcal{T}_2, \mathcal{T}_3)$.

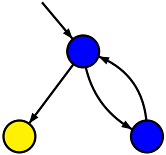
$$\mathcal{R} \stackrel{\text{def}}{=} \left\{ (s_1, s_3) \; : \; \exists s_2 \text{ s.t. } (s_1, s_2) \in \mathcal{R}_{1,2} \text{ and } (s_2, s_3) \in \mathcal{R}_{2,3} \right\}$$

is a bisimulation for $(\mathcal{T}_1, \mathcal{T}_3)$

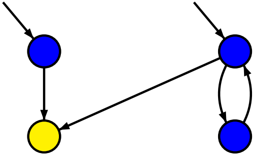
Correct or wrong?



Correct or wrong?

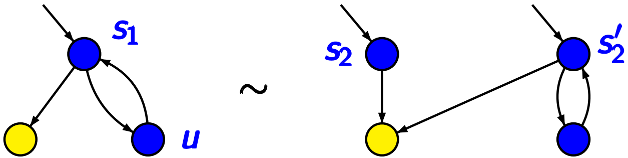


$\sim$



wrong

Correct or wrong?

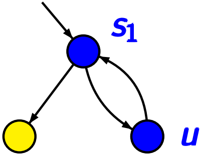


wrong

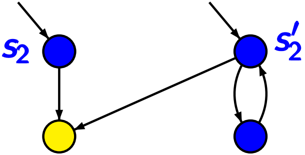
$s_1 \longrightarrow u$, but $s_2 \not\longrightarrow blue$ (thus $s_1 \not\sim s_2$)

Correct or wrong?

BSEQOR5.1-19

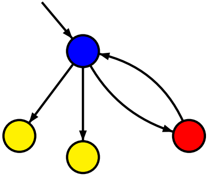


$\sim$

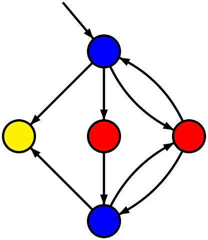


wrong

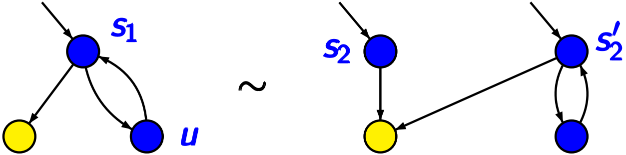
$s_1 \longrightarrow u$, but $s_2 \not\longrightarrow$ ***blue*** (thus $s_1 \not\sim s_2$)



$\sim$

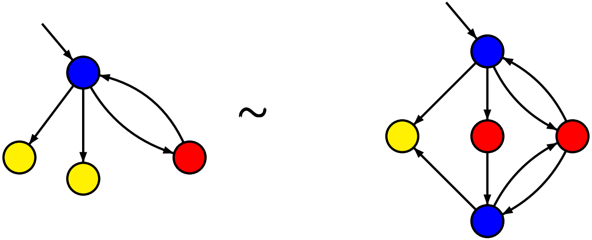


Correct or wrong?



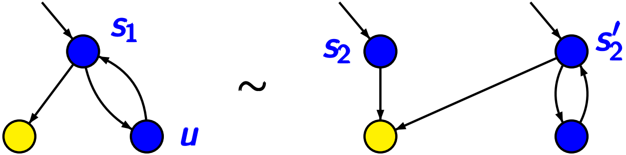
wrong

$s_1 \longrightarrow u$, but $s_2 \not\longrightarrow blue$ (thus $s_1 \not\sim s_2$)



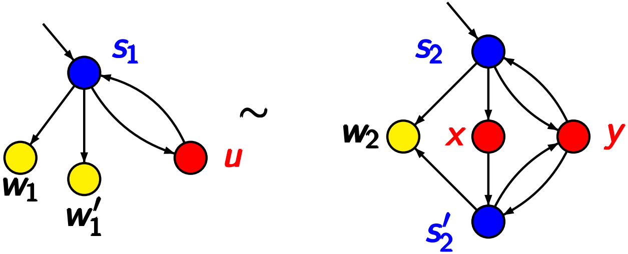
correct

Correct or wrong?



wrong

$s_1 \longrightarrow u$, but $s_2 \not\longrightarrow \mathit{blue}$ (thus $s_1 \not\sim s_2$)

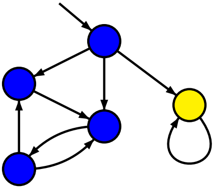


correct

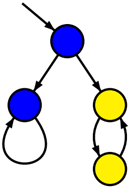
bisimulation:

$$\{(w_1, w_2), (w_1', w_2), (s_1, s_2), (s_1, s_2'), (u, x), (u, y)\}$$

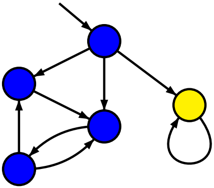
Correct or wrong?



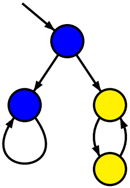
$\sim$



Correct or wrong?

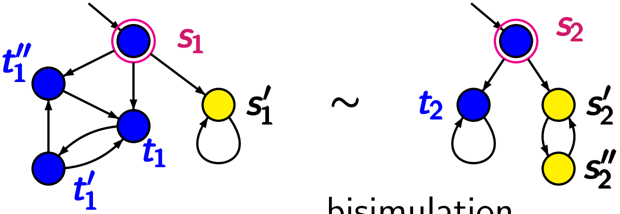


$\sim$



correct

Correct or wrong?

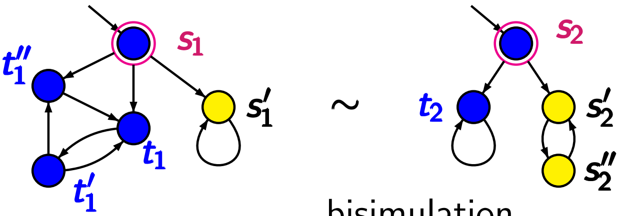


correct

bisimulation

$$\{(s_1, s_2), (s_1', s_2'), (s_1', s_2''), (t_1, t_2), (t_1', t_2), (t_1'', t_2)\}$$

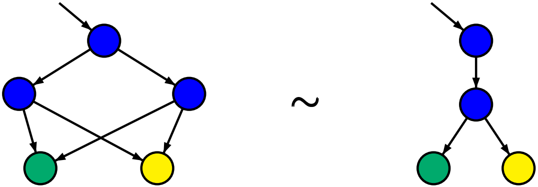
Correct or wrong?



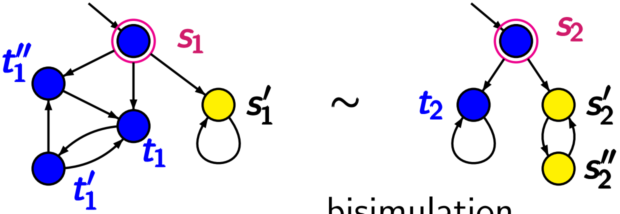
correct

bisimulation

$$\{(s_1, s_2), (s_1', s_2'), (s_1', s_2''), (t_1, t_2), (t_1', t_2), (t_1'', t_2)\}$$



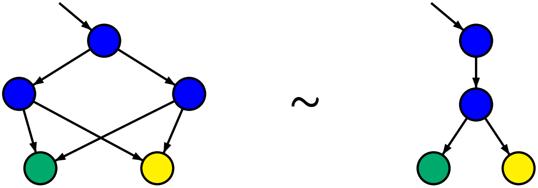
Correct or wrong?



correct

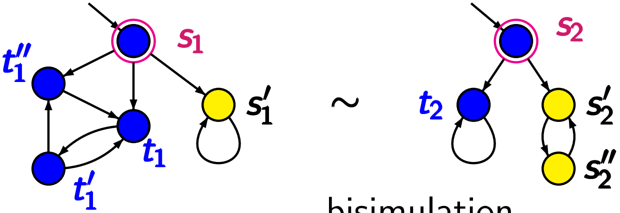
bisimulation

$$\{(s_1, s_2), (s_1', s_2'), (s_1', s_2''), (t_1, t_2), (t_1', t_2), (t_1'', t_2)\}$$



correct

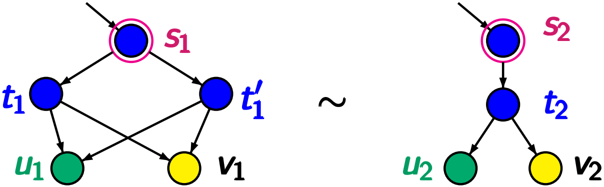
Correct or wrong?



correct

bisimulation

$$\{(s_1, s_2), (s_1', s_2'), (s_1', s_2''), (t_1, t_2), (t_1', t_2), (t_1'', t_2)\}$$



correct

bisimulation: $\{(s_1, s_2), (t_1, t_2), (t_1', t_2), (u_1, u_2), (v_1, v_2)\}$

Bisimulation vs. trace equivalence

Bisimulation vs. trace equivalence

$$\mathcal{T}_1 \sim \mathcal{T}_2 \implies \mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$$

Bisimulation vs. trace equivalence

$$\mathcal{T}_1 \sim \mathcal{T}_2 \Longrightarrow \mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$$

proof: ... path fragment lifting ...

Bisimulation vs. trace equivalence

$$\mathcal{T}_1 \sim \mathcal{T}_2 \Longrightarrow \mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$$

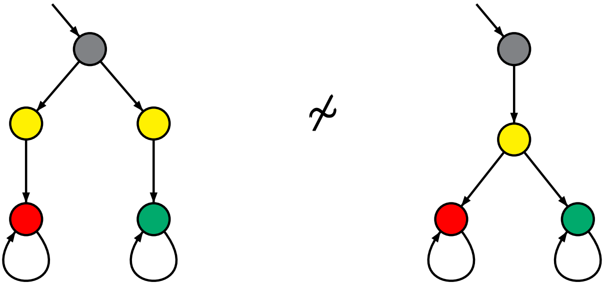
proof: ... path fragment lifting ...

$$\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2) \not\Longrightarrow \mathcal{T}_1 \sim \mathcal{T}_2$$

$$\mathcal{T}_1 \sim \mathcal{T}_2 \implies \mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$$

proof: ... path fragment lifting ...

$$\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2) \;\not\Longrightarrow\; \mathcal{T}_1 \sim \mathcal{T}_2$$



trace equivalent, but not bisimulation equivalent

Bisimulation vs. trace equivalence

$$\mathcal{T}_1 \sim \mathcal{T}_2 \implies \mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$$

proof: ... path fragment lifting ...

$$\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2) \not\Longrightarrow \mathcal{T}_1 \sim \mathcal{T}_2$$

Trace equivalence is strictly coarser than bisimulation equivalence.

$$\mathcal{T}_1 \sim \mathcal{T}_2 \implies Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$$

proof: ... path fragment lifting ...

$$Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2) \not\Longrightarrow \mathcal{T}_1 \sim \mathcal{T}_2$$

Trace equivalence is strictly coarser than bisimulation equivalence.

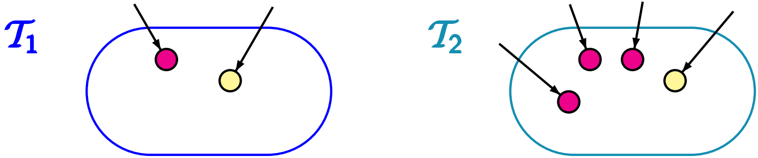
Bisimulation equivalent transition systems satisfy the same **LT properties** (e.g., **LTL** formulas).

Bisimulation equivalence ...

- as a relation that compares 2 transition systems

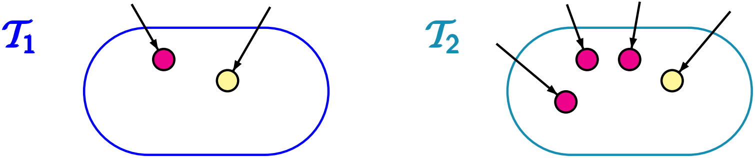
Bisimulation equivalence ...

- as a relation that compares **2** transition systems



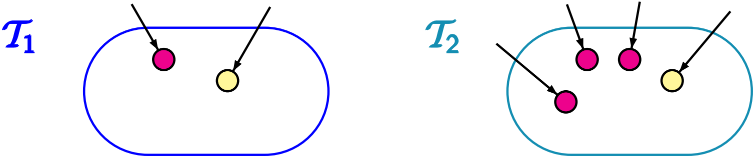
Bisimulation equivalence ...

- as a relation that compares **2** transition systems

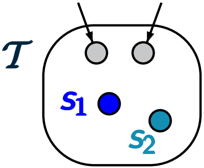


- as a relation on the states of **1** transition system

- as a relation that compares **2** transition systems

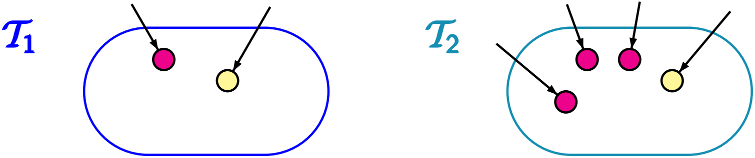


- as a relation on the states of **1** transition system

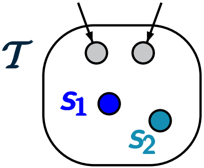


Bisimulation equivalence ...

- as a relation that compares **2** transition systems



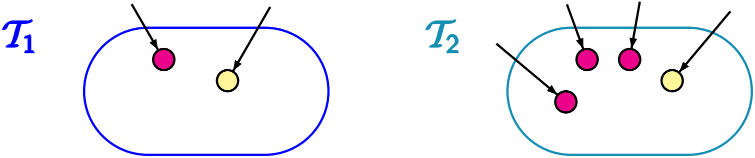
- as a relation on the states of **1** transition system



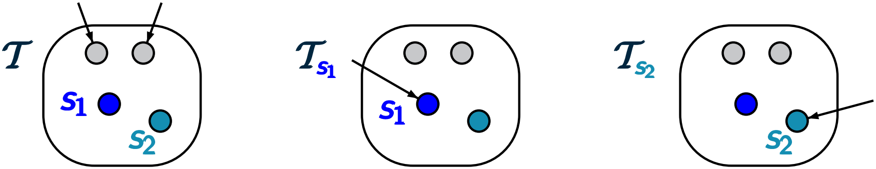
$s_1 \sim s_2$ iff $\mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$

Bisimulation equivalence ...

- as a relation that compares **2** transition systems

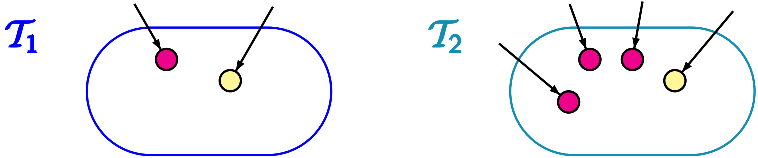


- as a relation on the states of **1** transition system

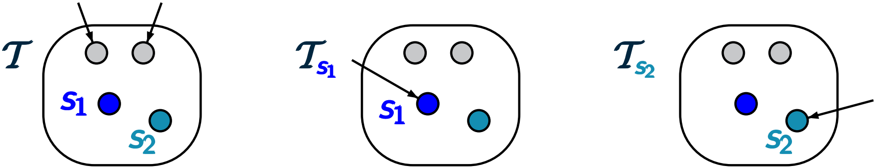


$s_1 \sim s_2$ iff $\mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$

- as a relation that compares **2** transition systems



- as a relation on the states of **1** transition system



$s_1 \sim s_2$ iff $\mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$ iff
there exists a bisimulation $\mathcal{R}$ for $\mathcal{T}$ s.t. $(s_1, s_2) \in \mathcal{R}$

Bisimulations on a single TS

Bisimulations on a single TS

Let $\mathcal{T}$ be a TS with proposition set AP.

Bisimulations on a single TS

Let $\mathcal{T}$ be a TS with proposition set AP.

A bisimulation for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on the state space of $\mathcal{T}$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L(s_1) = L(s_2)$

(2) $\forall\, s_1' \in Post(s_1)\ \exists\, s_2' \in Post(s_2)$ s.t. $(s_1', s_2') \in \mathcal{R}$

(3) $\forall\, s_2' \in Post(s_2)\ \exists\, s_1' \in Post(s_1)$ s.t. $(s_1', s_2') \in \mathcal{R}$

Let $\mathcal{T}$ be a TS with proposition set AP.

A bisimulation for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on the state space of $\mathcal{T}$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L(s_1) = L(s_2)$

(2) $\forall\, s_1' \in Post(s_1)\ \exists\, s_2' \in Post(s_2)$ s.t. $(s_1', s_2') \in \mathcal{R}$

(3) $\forall\, s_2' \in Post(s_2)\ \exists\, s_1' \in Post(s_1)$ s.t. $(s_1', s_2') \in \mathcal{R}$

bisimulation equivalence $\sim_{\mathcal{T}}$:

$s_1 \sim_{\mathcal{T}} s_2$ iff there exists a bisimulation $\mathcal{R}$ for $\mathcal{T}$ s.t. $(s_1, s_2) \in \mathcal{R}$

Let $\mathcal{T}$ be a TS with proposition set AP.

A bisimulation for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on the state space of $\mathcal{T}$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L(s_1) = L(s_2)$

(2) $\forall\, s_1' \in Post(s_1)\ \exists\, s_2' \in Post(s_2)$ s.t. $(s_1', s_2') \in \mathcal{R}$

(3) $\forall\, s_2' \in Post(s_2)\ \exists\, s_1' \in Post(s_1)$ s.t. $(s_1', s_2') \in \mathcal{R}$

coinductive definition of $\sim_{\mathcal{T}}$:

$s_1 \sim_{\mathcal{T}} s_2$ iff there exists a bisimulation $\mathcal{R}$ for $\mathcal{T}$ s.t. $(s_1, s_2) \in \mathcal{R}$

Let $\mathcal{T}$ be a transition system with state space S.

Bisimulation equivalence $\sim_{\mathcal{T}}$ is

Bisimulation equivalence

Let $\mathcal{T}$ be a transition system with state space S.

Bisimulation equivalence $\sim_{\mathcal{T}}$ is

- the coarsest bisimulation on $\mathcal{T}$

Bisimulation equivalence

Let $\mathcal{T}$ be a transition system with state space S.

Bisimulation equivalence $\sim_{\mathcal{T}}$ is

- the coarsest bisimulation on $\mathcal{T}$
- and an equivalence on S

Bisimulation equivalence

Let $\mathcal{T}$ be a transition system with state space S.

Bisimulation equivalence $\sim_{\mathcal{T}}$ is the coarsest equivalence on S s.t. for all states $s_1, s_2 \in S$ with $s_1 \sim_{\mathcal{T}} s_2$:

Bisimulation equivalence

Let $\mathcal{T}$ be a transition system with state space S.

Bisimulation equivalence $\sim_{\mathcal{T}}$ is the coarsest equivalence on S s.t. for all states $s_1, s_2 \in S$ with $s_1 \sim_{\mathcal{T}} s_2$:

(1) $L(s_1) = L(s_2)$

(2) each transition of s_1 can be mimicked by a transition of s_2:

Let $\mathcal{T}$ be a transition system with state space S.

Bisimulation equivalence $\sim_{\mathcal{T}}$ is the coarsest equivalence on S s.t. for all states $s_1, s_2 \in S$ with $s_1 \sim_{\mathcal{T}} s_2$:

(1) $L(s_1) = L(s_2)$

(2) each transition of s_1 can be mimicked by a transition of s_2:

$$\boxed{\begin{array}{ccc} s_1 & \sim_{\mathcal{T}} & s_2 \\ \downarrow & & \\ s_1' & & \end{array}} \quad \text{can be completed to} \quad \boxed{\begin{array}{ccc} s_1 & \sim_{\mathcal{T}} & s_2 \\ \downarrow & & \downarrow \\ s_1' & \sim_{\mathcal{T}} & s_2' \end{array}}$$

Two variants of bisimulation equivalence

$\sim$ relation that compares **2** transition systems

$\sim_{\mathcal{T}}$ equivalence on the state space of a single TS $\mathcal{T}$

Two variants of bisimulation equivalence

$\sim$ relation that compares **2** transition systems

$\sim_{\mathcal{T}}$ equivalence on the state space of a single TS $\mathcal{T}$

1. $\sim_{\mathcal{T}}$ can be derived from $\sim$

$\sim$ relation that compares 2 transition systems

$\sim_{\mathcal{T}}$ equivalence on the state space of a single TS $\mathcal{T}$

1. $\sim_{\mathcal{T}}$ can be derived from $\sim$

for all states s_1 and s_2 of $\mathcal{T}$:

$$s_1 \sim_{\mathcal{T}} s_2 \quad \text{iff} \quad \mathcal{T}_{s_1} \sim \mathcal{T}_{s_1}$$

Two variants of bisimulation equivalence

$\sim$ relation that compares **2** transition systems

$\sim_{\mathcal{T}}$ equivalence on the state space of a single TS $\mathcal{T}$

1. $\sim_{\mathcal{T}}$ can be derived from $\sim$

for all states s_1 and s_2 of $\mathcal{T}$:

$$s_1 \sim_{\mathcal{T}} s_2 \quad \text{iff} \quad \mathcal{T}_{s_1} \sim \mathcal{T}_{s_1}$$

where $\mathcal{T}_s$ agrees with $\mathcal{T}$, except that state s is declared to be the unique initial state

Two variants of bisimulation equivalence

$\sim$ relation that compares **2** transition systems

$\sim_{\mathcal{T}}$ equivalence on the state space of a single TS $\mathcal{T}$

1. $\sim_{\mathcal{T}}$ can be derived from $\sim$

for all states s_1 and s_2 of $\mathcal{T}$:

$$s_1 \sim_{\mathcal{T}} s_2 \quad \text{iff} \quad \mathcal{T}_{s_1} \sim \mathcal{T}_{s_1}$$

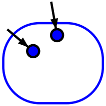
where $\mathcal{T}_s$ agrees with $\mathcal{T}$, except that state s is declared to be the unique initial state

2. $\sim$ can be derived from $\sim_{\mathcal{T}}$

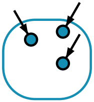
Derivation of $\sim$ from $\sim_{\mathcal{T}}$

given two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$

$\mathcal{T}_1$ with state space S_1



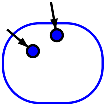
$\mathcal{T}_2$ with state space S_2



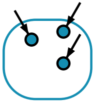
Derivation of $\sim$ from $\sim_{\mathcal{T}}$

given two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$

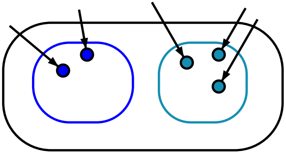
$\mathcal{T}_1$ with state space S_1



$\mathcal{T}_2$ with state space S_2



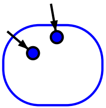
consider $\mathcal{T} = \mathcal{T}_1 \uplus \mathcal{T}_2$
(state space $S_1 \uplus S_2$)



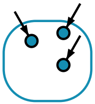
Derivation of $\sim$ from $\sim_{\mathcal{T}}$

given two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$

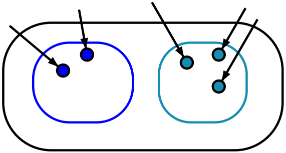
$\mathcal{T}_1$ with state space S_1



$\mathcal{T}_2$ with state space S_2



consider $\mathcal{T} = \mathcal{T}_1 \uplus \mathcal{T}_2$
(state space $S_1 \uplus S_2$)

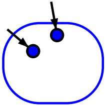


$\mathcal{T}_1 \sim \mathcal{T}_2$ iff $\forall$ initial states s_1 of $\mathcal{T}_1$
$\exists$ initial state s_2 of $\mathcal{T}_2$ s.t. $s_1 \sim_{\mathcal{T}} s_2$,

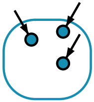
Derivation of $\sim$ from $\sim_{\mathcal{T}}$

given two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$

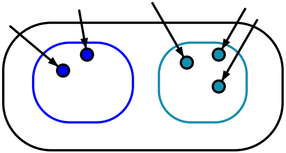
$\mathcal{T}_1$ with state space S_1



$\mathcal{T}_2$ with state space S_2



consider $\mathcal{T} = \mathcal{T}_1 \uplus \mathcal{T}_2$
(state space $S_1 \uplus S_2$)



$\mathcal{T}_1 \sim \mathcal{T}_2$ iff $\forall$ initial states s_1 of $\mathcal{T}_1$
$\exists$ initial state s_2 of $\mathcal{T}_2$ s.t. $s_1 \sim_{\mathcal{T}} s_2$,
and vice versa

Bisimulation quotient

Bisimulation quotient

Let $\mathcal{T} = (S, Act, \to, S_0, AP, L)$ be a TS.

Bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient $\mathcal{T}/\sim$ arises from $\mathcal{T}$
by collapsing bisimulation equivalent states

Bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/\!\sim\; = (S', Act', \rightarrow', S_0', AP, L')$$

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/\sim = (S', Act', \rightarrow', S'_0, AP, L')$$

- state space: $S' = S/\sim_{\mathcal{T}}$

set of bisimulation equivalence classes

Bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/\!\sim\; = (S', Act', \rightarrow', S'_0, AP, L')$$

- state space: $S' = S/\!\sim_{\mathcal{T}}$
- set of initial states: $S'_0 = \left\{ [s]_{\sim_{\mathcal{T}}} : s \in S_0 \right\}$

Bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/\sim = (S', Act', \rightarrow', S'_0, AP, L')$$

- state space: $S' = S/\sim_{\mathcal{T}}$
- set of initial states: $S'_0 = \{[s]_{\sim_{\mathcal{T}}} : s \in S_0\}$
- labeling function: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/{\sim} = (S', Act', \rightarrow', S'_0, AP, L')$$

- state space: $S' = S/{\sim_{\mathcal{T}}}$
- set of initial states: $S'_0 = \{[s]_{\sim_{\mathcal{T}}} : s \in S_0\}$
- labeling function: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$

well-defined

by the labeling condition of bisimulations

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/\!\sim\; = (S', Act', \rightarrow', S_0', AP, L')$$

- state space: $S' = S/\!\sim_{\mathcal{T}}$
- set of initial states: $S_0' = \{[s]_{\sim_{\mathcal{T}}} : s \in S_0\}$
- labeling function: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$
- transition relation:

$$\frac{s \longrightarrow s'}{[s]_{\sim_{\mathcal{T}}} \longrightarrow [s']_{\sim_{\mathcal{T}}}}$$

Bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/\!\sim\; = (S', Act', \;\rightarrow', S'_0, AP, L')$$

- state space: $S' = S/\!\sim_{\mathcal{T}}$
- set of initial states: $S'_0 = \big\{ [s]_{\sim_{\mathcal{T}}} : s \in S_0 \big\}$
- labeling function: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$
- transition relation:

$$\frac{s \longrightarrow s'}{[s]_{\sim_{\mathcal{T}}} \longrightarrow [s']_{\sim_{\mathcal{T}}}}$$

action labels irrelevant

Bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/\!\sim\; = (S', \{\tau\}, \rightarrow', S'_0, AP, L')$$

- state space: $S' = S/\!\sim_{\mathcal{T}}$
- set of initial states: $S'_0 = \big\{ [s]_{\sim_{\mathcal{T}}} : s \in S_0 \big\}$
- labeling function: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$
- transition relation:

$$\frac{s \xrightarrow{\alpha} s'}{[s]_{\sim_{\mathcal{T}}} \xrightarrow{\tau} [s']_{\sim_{\mathcal{T}}}}$$

action labels irrelevant

Bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

bisimulation quotient:

$$\mathcal{T}/\!\sim \; = (S', \{\tau\}, \rightarrow', S'_0, AP, L')$$

- state space: $S' = S/\!\sim_{\mathcal{T}}$
- set of initial states: $S'_0 = \left\{ [s]_{\sim_{\mathcal{T}}} : s \in S_0 \right\}$
- labeling function: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$
- transition relation:

$$\frac{s \xrightarrow{\alpha} s'}{[s]_{\sim_{\mathcal{T}}} \xrightarrow{\tau} [s']_{\sim_{\mathcal{T}}}}$$

$$\boxed{\mathcal{T} \sim \mathcal{T}/\!\sim}$$

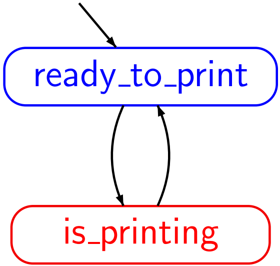
Example: interleaving of *n* printers

parallel system $\mathcal{T} = \underbrace{\mathit{Printer} \mid\mid\mid \mathit{Printer} \mid\mid\mid \ldots \mid\mid\mid \mathit{Printer}}_{\boldsymbol{n} \text{ printer}}$

Example: interleaving of *n* printers

parallel system $\mathcal{T} = \underbrace{\mathit{Printer} \,|||\, \mathit{Printer} \,|||\, \ldots \,|||\, \mathit{Printer}}_{n \text{ printer}}$

transition system for each printer

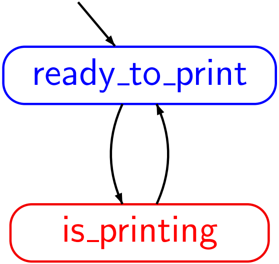


Example: interleaving of n printers

parallel system $\mathcal{T} = \underbrace{\mathit{Printer} \,|||\, \mathit{Printer} \,|||\, \ldots \,|||\, \mathit{Printer}}_{n \text{ printer}}$

$AP = \{0, 1, \ldots, n\}$ "number of available printers"

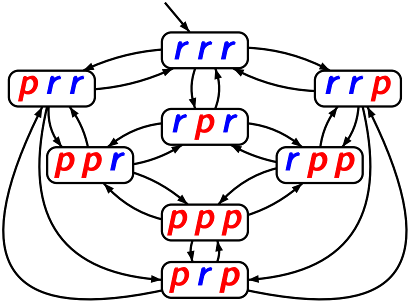
transition system for each printer



Example: n=3 printers

parallel system $\mathcal{T} = \underbrace{\mathit{Printer} \mid\mid\mid \mathit{Printer} \mid\mid\mid \ldots \mid\mid\mid \mathit{Printer}}_{n \text{ printer}}$

$AP = \{0, 1, 2, 3\}$



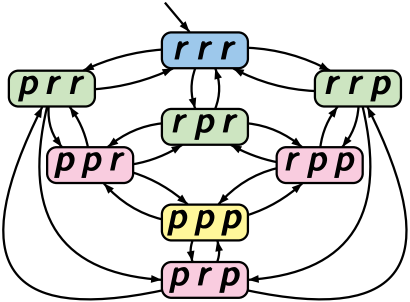
p: is printing

r: ready to print

Example: n=3 printers

parallel system $\mathcal{T} = \underbrace{Printer \,|||\, Printer \,|||\, \ldots \,|||\, Printer}_{n \text{ printer}}$

$AP = \{0, 1, 2, 3\}$



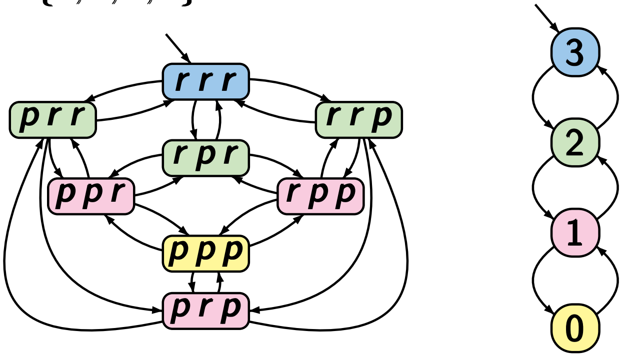
p: is printing

r: ready to print

Example: n=3 printers

parallel system $\mathcal{T} = \underbrace{\textit{Printer} \,|||\, \textit{Printer} \,|||\, \ldots \,|||\, \textit{Printer}}_{n \text{ printer}}$

$AP = \{0, 1, 2, 3\}$



p: is printing

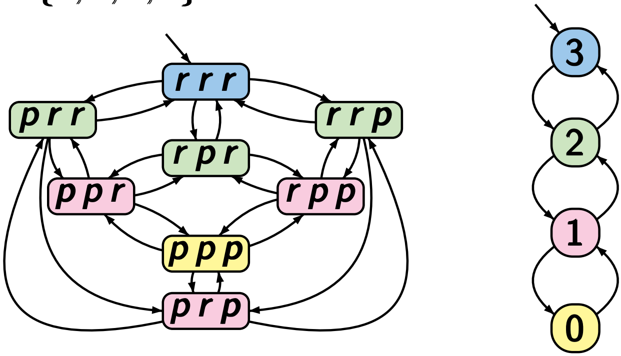
r: ready to print

bisimulation quotient

Example: n=3 printers

parallel system $\mathcal{T} = \underbrace{Printer \,|||\, Printer \,|||\, \ldots \,|||\, Printer}_{n \text{ printer}}$

$AP = \{0, 1, 2, 3\}$



2^n states

$n+1$ states

Mutual exclusion

Mutual exclusion

solutions for mutual exclusion problems:

- semaphore
- Peterson's algorithm

Mutual exclusion: Bakery algorithm

solutions for mutual exclusion problems:

- semaphore
- Peterson's algorithm
- Bakery algorithm

solutions for mutual exclusion problems:

- semaphore
- Peterson's algorithm
- Bakery algorithm

given two concurrent processes P_1 and P_2

Mutual exclusion: Bakery algorithm

solutions for mutual exclusion problems:

- semaphore
- Peterson's algorithm
- Bakery algorithm

given two concurrent processes P_1 and P_2

- two additional shared variables: $x_1, x_2 \in \mathbb{N}$

Mutual exclusion: Bakery algorithm

solutions for mutual exclusion problems:

- semaphore
- Peterson's algorithm
- Bakery algorithm

given two concurrent processes P_1 and P_2

- two additional shared variables: $x_1, x_2 \in \mathbb{N}$
- if P_1 and P_2 are waiting then:

solutions for mutual exclusion problems:

- semaphore
- Peterson's algorithm
- Bakery algorithm

given two concurrent processes P_1 and P_2

- two additional shared variables: $x_1, x_2 \in \mathbb{N}$
- if P_1 and P_2 are waiting then:

  if $x_1 < x_2$ then P_1 enters its critical section

  if $x_2 < x_1$ then P_2 enters its critical section

Mutual exclusion: Bakery algorithm

solutions for mutual exclusion problems:

- semaphore
- Peterson's algorithm
- Bakery algorithm

given two concurrent processes P_1 and P_2

- two additional shared variables: $x_1, x_2 \in \mathbb{N}$
- if P_1 and P_2 are waiting then:

  if $x_1 < x_2$ then P_1 enters its critical section

  if $x_2 < x_1$ then P_2 enters its critical section

  $x_1 = x_2$: cannot happen

Bakery algorithm

protocol for P_1:

```
LOOP FOREVER
    noncritical actions
    x1 := x2+1
    AWAIT (x1 < x2) ∨ (x2=0);
    critical section;
    x1 := 0
END LOOP
```

symmetric protocol for P_2

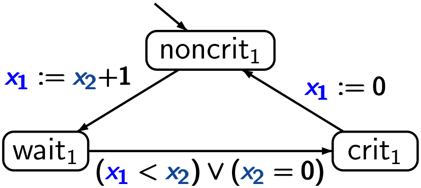
protocol for P_1:

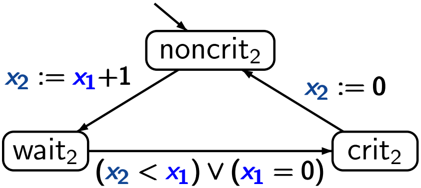
```
LOOP FOREVER
    noncritical actions
    x1 := x2+1
    AWAIT (x1 < x2) ∨ (x2=0);
    critical section;
    x1 := 0
END LOOP
```

initially: $x_1 = x_2 = 0$

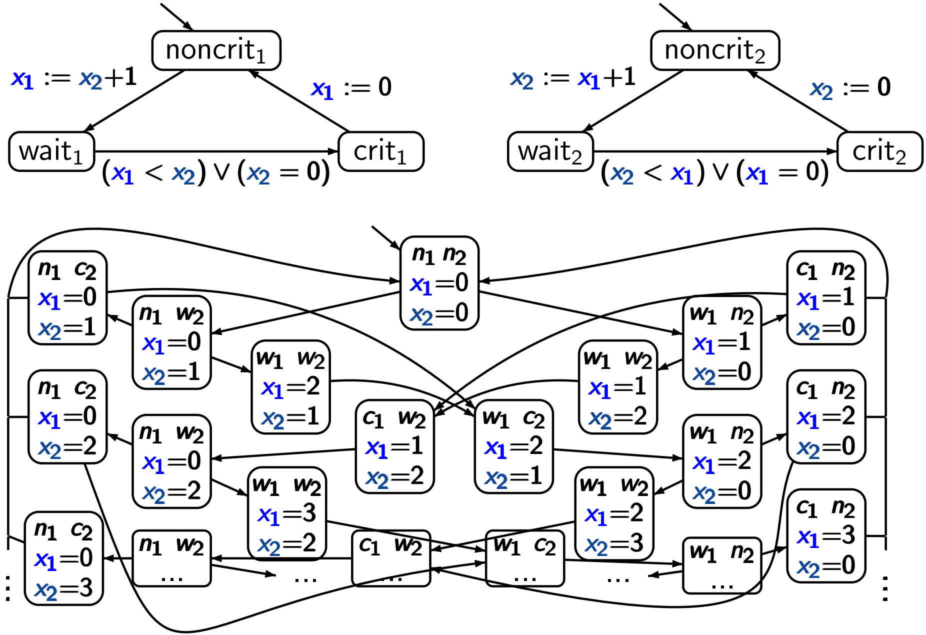
symmetric protocol for P_2

Program graphs for the Bakery algorithm

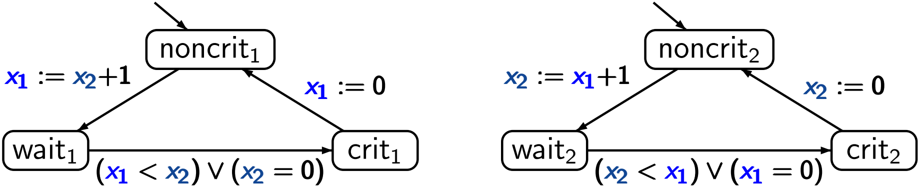


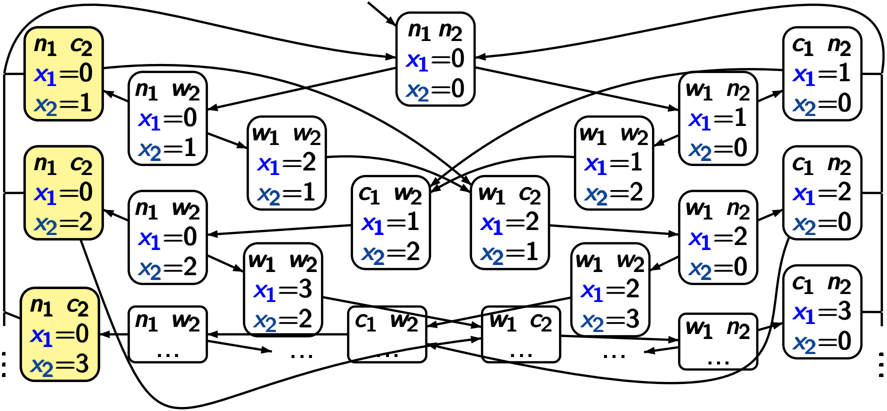


Transition system for the Bakery algorithm

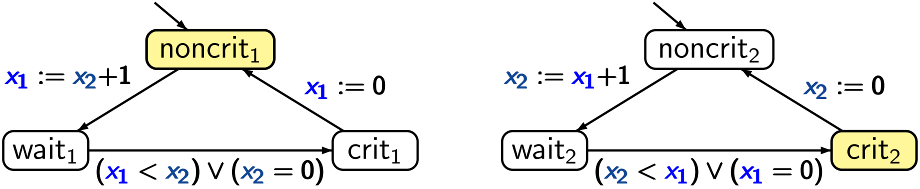


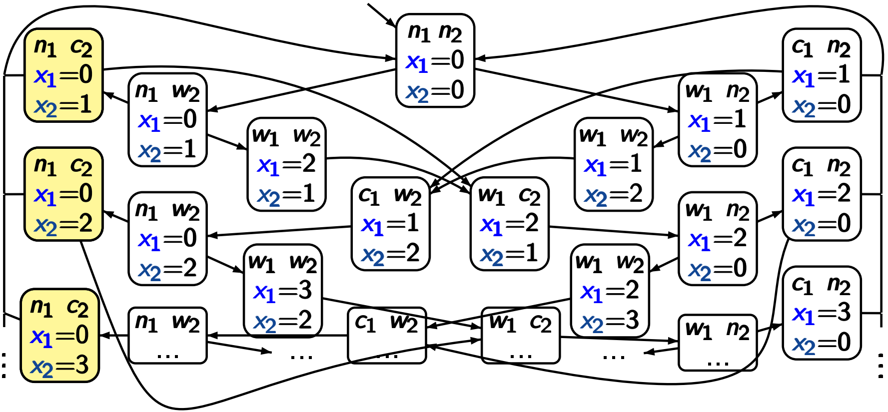
Transition system for the Bakery algorithm



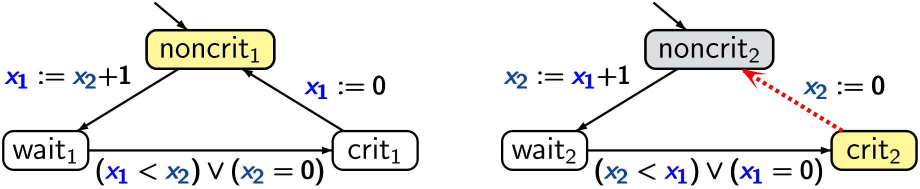


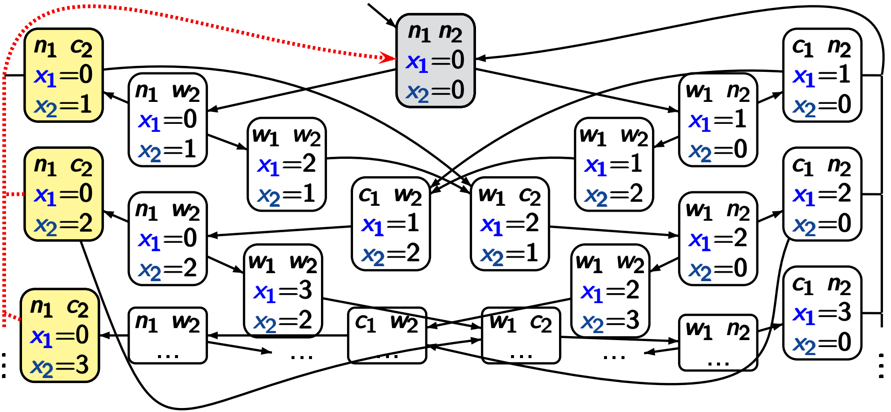
Transition system for the Bakery algorithm



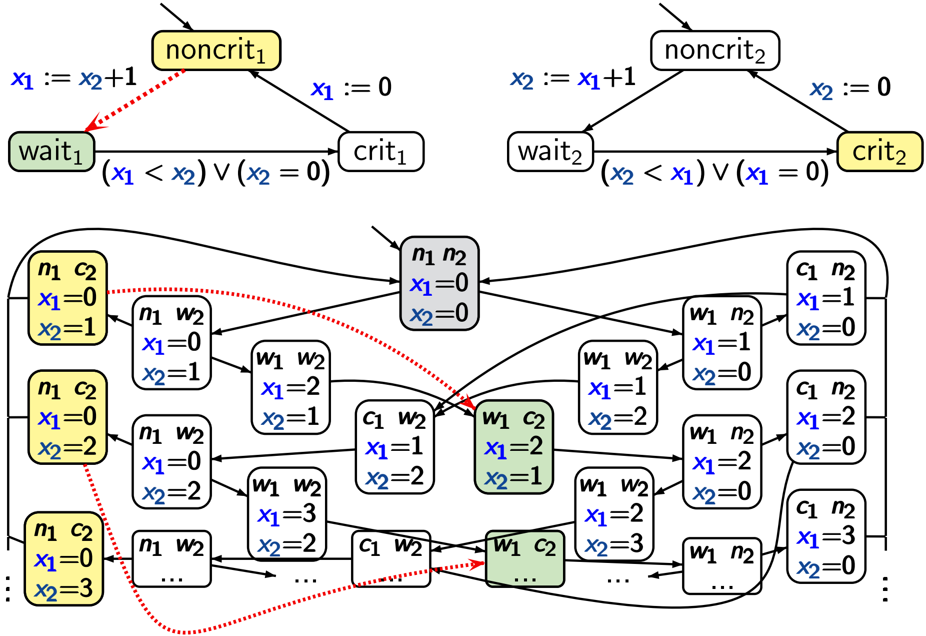


Transition system for the Bakery algorithm

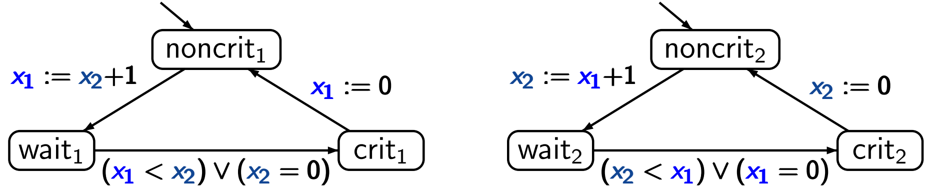


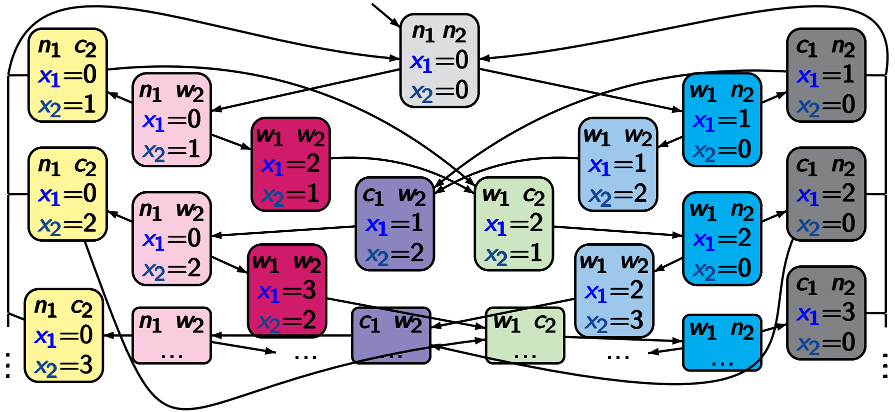


Transition system for the Bakery algorithm

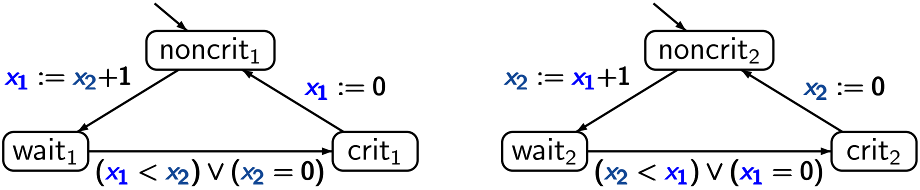


Transition system for the Bakery algorithm



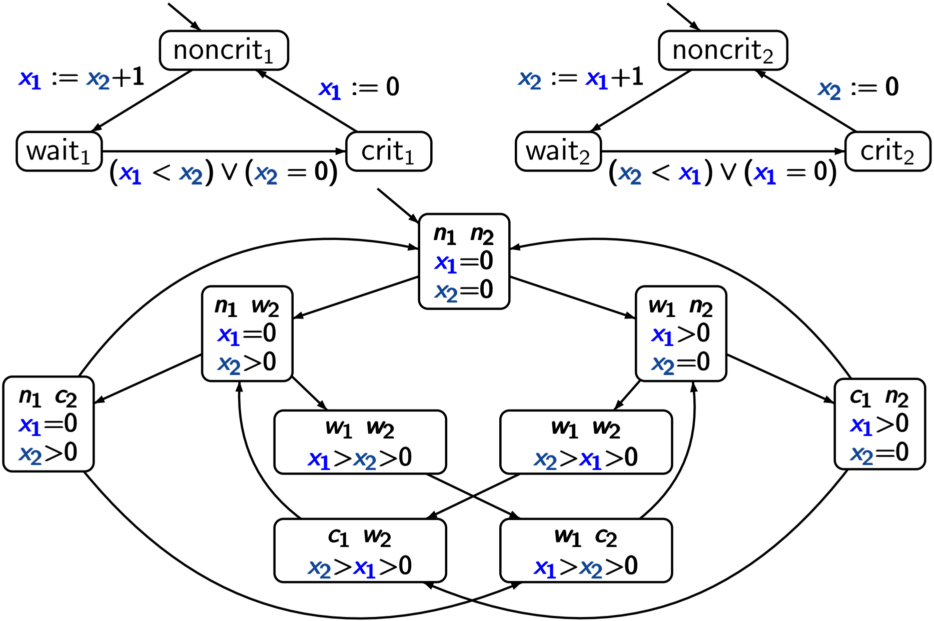


Bakery algorithm: bisimulation quotient

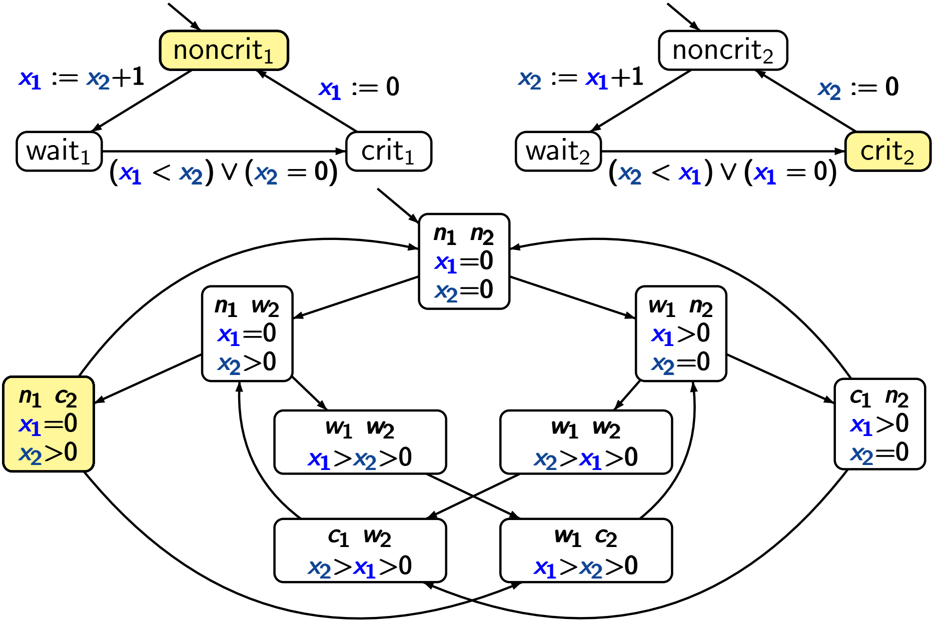


infinite transition system with a
finite bisimulation quotient

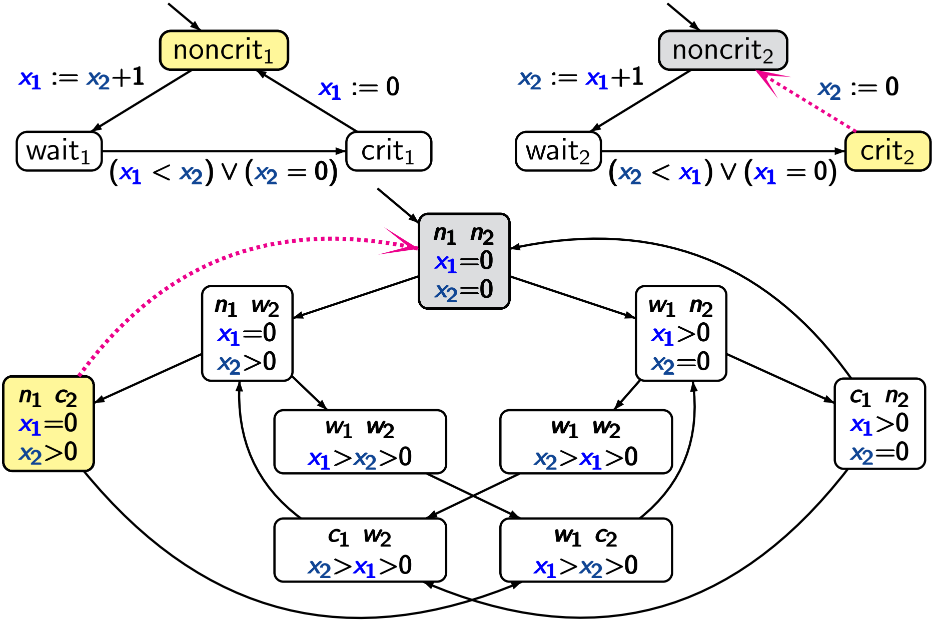
Bakery algorithm: bisimulation quotient



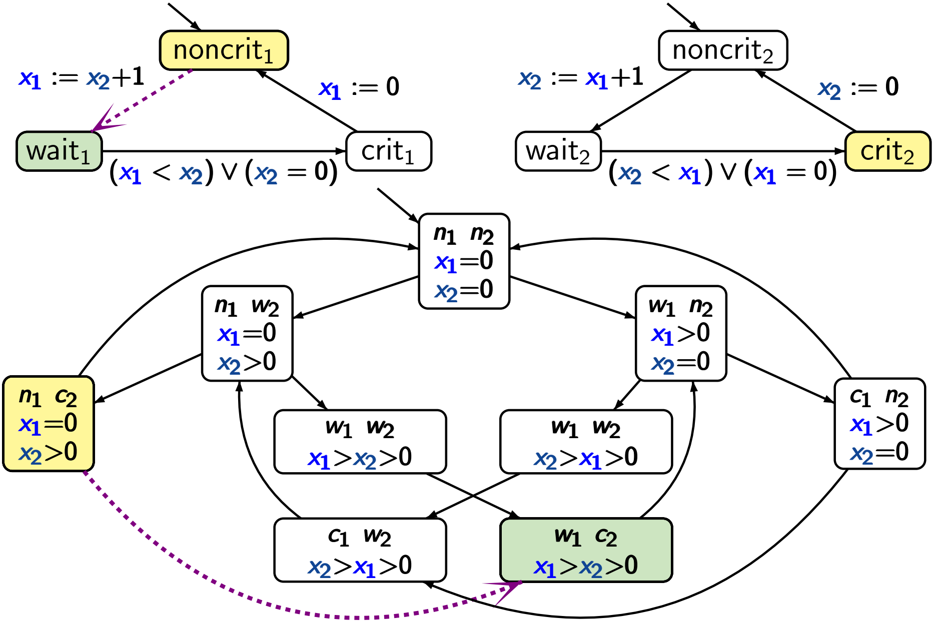
Bakery algorithm: bisimulation quotient



Bakery algorithm: bisimulation quotient



Bakery algorithm: bisimulation quotient



Introduction

Modelling parallel systems

Linear Time Properties

Regular Properties

Linear Temporal Logic (LTL)

Computation-Tree Logic

Equivalences and Abstraction

- bisimulation
- CTL, CTL*-equivalence $\longleftarrow$
- computing the bisimulation quotient
- abstraction stutter steps
- simulation relations

Recall: CTL*

CTL* state formulas

$$\Phi ::= \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi$$

CTL* path formulas

$$\varphi ::= \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

Recall: CTL*

CTL* state formulas

$$\Phi ::= \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi$$

CTL* path formulas

$$\varphi ::= \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \mathsf{U} \varphi_2$$

derived operators:

- $\Diamond$, $\Box$, ... as in **LTL**

Recall: CTL*

CTL* state formulas

$$\Phi \ ::= \ \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi$$

CTL* path formulas

$$\varphi \ ::= \ \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \, \mathsf{U} \, \varphi_2$$

derived operators:

- $\Diamond$, $\Box$, ... as in **LTL**
- universal quantification: $\forall\varphi \stackrel{\text{def}}{=} \neg\exists\neg\varphi$

Recall: CTL* and CTL

CTL* state formulas

$$\Phi \;::=\; \mathit{true} \;\big|\; a \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; \neg\Phi \;\big|\; \exists\varphi$$

CTL* path formulas

$$\varphi \;::=\; \Phi \;\big|\; \varphi_1 \wedge \varphi_2 \;\big|\; \neg\varphi \;\big|\; \bigcirc\varphi \;\big|\; \varphi_1 \,\mathsf{U}\, \varphi_2$$

CTL: sublogic of **CTL***

Recall: CTL* and CTL

CTL* state formulas

$$\Phi \;::=\; \mathit{true} \;\big|\; a \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; \neg\Phi \;\big|\; \exists\varphi$$

CTL* path formulas

$$\varphi \;::=\; \Phi \;\big|\; \varphi_1 \wedge \varphi_2 \;\big|\; \neg\varphi \;\big|\; \bigcirc\varphi \;\big|\; \varphi_1 \,\mathsf{U}\, \varphi_2$$

CTL: sublogic of **CTL***

- with path quantifiers $\exists$ and $\forall$

Recall: CTL* and CTL

CTL* state formulas

$$\Phi \ ::= \ \mathit{true} \ \big| \ a \ \big| \ \Phi_1 \wedge \Phi_2 \ \big| \ \neg\Phi \ \big| \ \exists\varphi$$

CTL* path formulas

$$\varphi \ ::= \ \Phi \ \big| \ \varphi_1 \wedge \varphi_2 \ \big| \ \neg\varphi \ \big| \ \bigcirc\varphi \ \big| \ \varphi_1 \, \mathsf{U} \, \varphi_2$$

CTL: sublogic of **CTL***

- with path quantifiers $\exists$ and $\forall$
- restricted syntax of path formulas:

CTL* state formulas

$$\Phi \ ::= \ \mathit{true} \ \big| \ a \ \big| \ \Phi_1 \wedge \Phi_2 \ \big| \ \neg\Phi \ \big| \ \exists\varphi$$

CTL* path formulas

$$\varphi \ ::= \ \Phi \ \big| \ \varphi_1 \wedge \varphi_2 \ \big| \ \neg\varphi \ \big| \ \bigcirc\varphi \ \big| \ \varphi_1 \, \mathbf{U} \, \varphi_2$$

CTL: sublogic of **CTL***

- with path quantifiers $\exists$ and $\forall$
- restricted syntax of path formulas:
  - *no* boolean combinations of path formulas
  - arguments of temporal operators $\bigcirc$ and $\mathbf{U}$ are state formulas

CTL equivalence

Let s_1, s_2 be states of a TS $\mathcal{T}$ without terminal states

CTL equivalence

Let s_1, s_2 be states of a TS $\mathcal{T}$ without terminal states

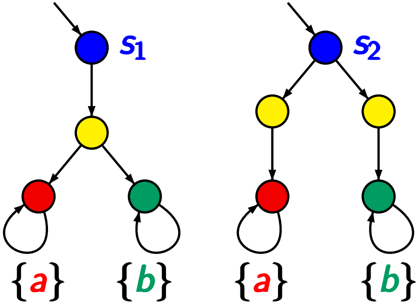
s_1, s_2 are **CTL** equivalent if for all **CTL** formulas Φ:

$$s_1 \models \Phi \quad \text{iff} \quad s_2 \models \Phi$$

Let s_1, s_2 be states of a TS $\mathcal{T}$ without terminal states

s_1, s_2 are **CTL** equivalent if for all **CTL** formulas Φ:

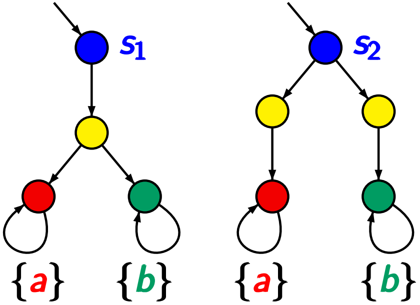
$$s_1 \models \Phi \quad \text{iff} \quad s_2 \models \Phi$$



Let s_1, s_2 be states of a TS $\mathcal{T}$ without terminal states

s_1, s_2 are **CTL** equivalent if for all **CTL** formulas Φ:

$$s_1 \models \Phi \quad \text{iff} \quad s_2 \models \Phi$$



s_1, s_2 are not **CTL** equivalent

$s_1 \models \exists\bigcirc(\exists\bigcirc a \wedge \exists\bigcirc b)$

$s_2 \not\models \exists\bigcirc(\exists\bigcirc a \wedge \exists\bigcirc b)$

CTL, CTL* and LTL equivalence

Let s_1, s_2 be states of a TS $\mathcal{T}$ without terminal states

s_1, s_2 are **CTL** equivalent if for all **CTL** formulas Φ:

$$s_1 \models \Phi \quad \text{iff} \quad s_2 \models \Phi$$

analogous definition for **CTL*** and **LTL**

CTL, CTL* and LTL equivalence

Let s_1, s_2 be states of a TS $\mathcal{T}$ without terminal states

s_1, s_2 are **CTL** equivalent if for all **CTL** formulas Φ:

$$s_1 \models \Phi \quad \text{iff} \quad s_2 \models \Phi$$

s_1, s_2 are **CTL*** equivalent if for all **CTL*** formulas Φ:

$$s_1 \models \Phi \quad \text{iff} \quad s_2 \models \Phi$$

s_1, s_2 are **LTL** equivalent if for all **LTL** formulas φ:

$$s_1 \models \varphi \quad \text{iff} \quad s_2 \models \varphi$$

CTL/CTL* and bisimulation

CTL/CTL* and bisimulation

bisimulation equivalence

= **CTL** equivalence

= **CTL*** equivalence

CTL/CTL* and bisimulation

	bisimulation equivalence	
$=$	**CTL** equivalence	$\longleftarrow$ for finite TS
$=$	**CTL*** equivalence	

CTL/CTL* and bisimulation

	bisimulation equivalence	
=	**CTL** equivalence	$\longleftarrow$ for finite TS
=	**CTL*** equivalence	

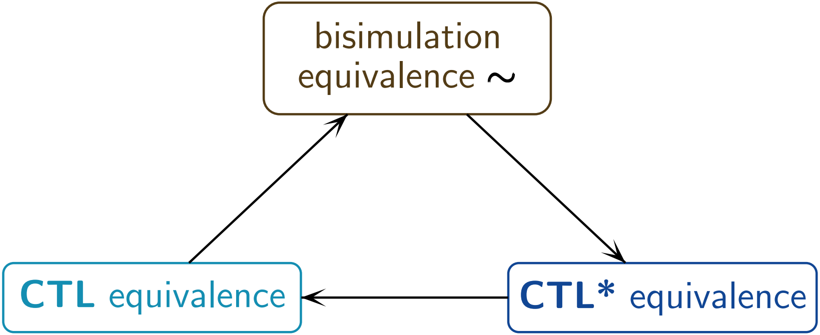
Let $\mathcal{T}$ be a finite TS without terminal states, and s_1, s_2 states in $\mathcal{T}$. Then:

$$s_1 \sim_{\mathcal{T}} s_2$$

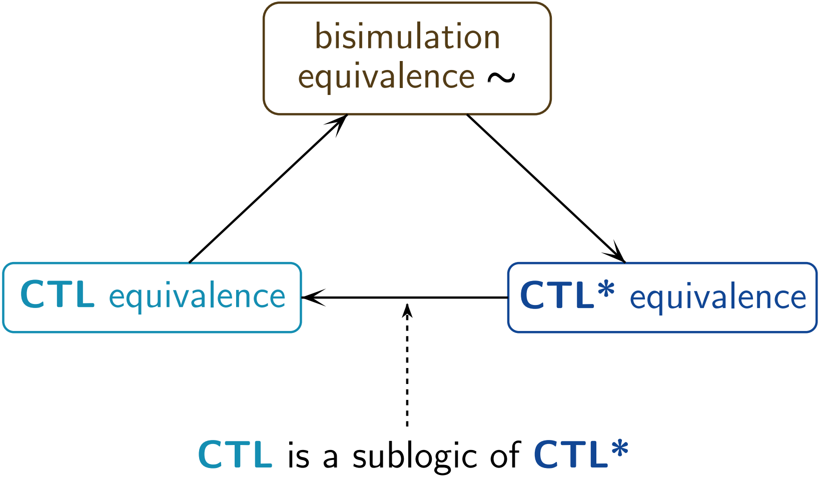
iff s_1 and s_2 are **CTL** equivalent

iff s_1 and s_2 are **CTL*** equivalent

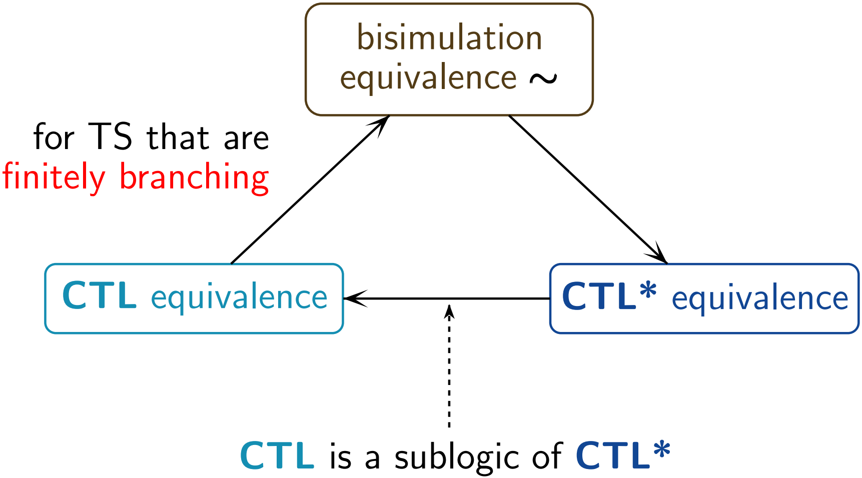
CTL/CTL* and bisimulation



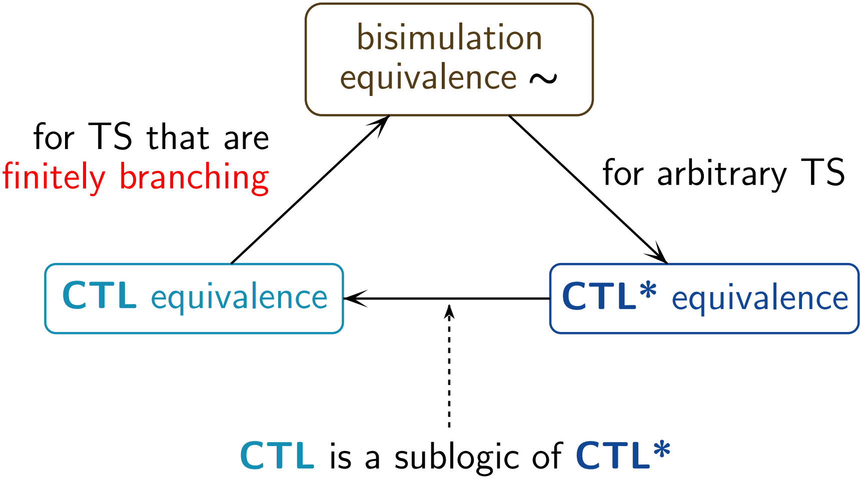
CTL/CTL* and bisimulation



CTL/CTL* and bisimulation



CTL/CTL* and bisimulation



CTL/CTL* and bisimulation for TS

If $\mathcal{T}_1$, $\mathcal{T}_2$ are finitely branching TS over AP without terminal states then:

$$\mathcal{T}_1 \sim \mathcal{T}_2$$

iff $\mathcal{T}_1$ and $\mathcal{T}_2$ satisfy the same **CTL** formulas

iff $\mathcal{T}_1$ and $\mathcal{T}_2$ satisfy the same **CTL*** formulas

Correct or wrong?

CTLEQ5.2-9

CTL equivalence is finer than **LTL** equivalence

Correct or wrong?

CTL equivalence is finer than **LTL** equivalence

correct.

Correct or wrong?

CTL equivalence is finer than **LTL** equivalence

correct.

$\uparrow$

CTL equivalence = **CTL*** equivalence

LTL is sublogic of **CTL***

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

Correct or wrong?

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

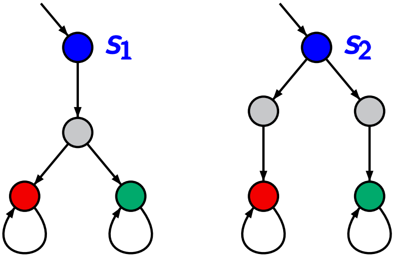
wrong.

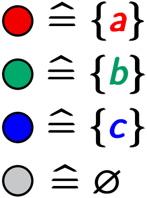
CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.



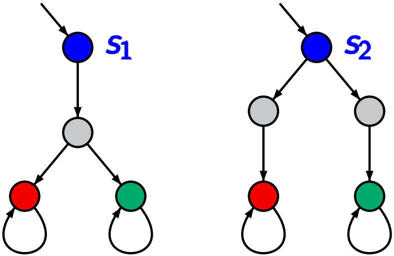


CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.



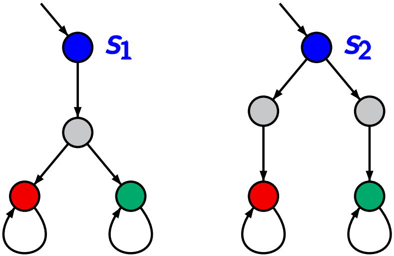
s_1, s_2 are trace equivalent

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.



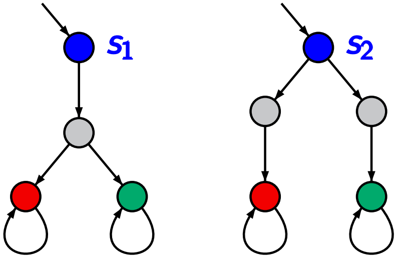
s_1, s_2 are trace equivalent and **LTL** equivalent

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

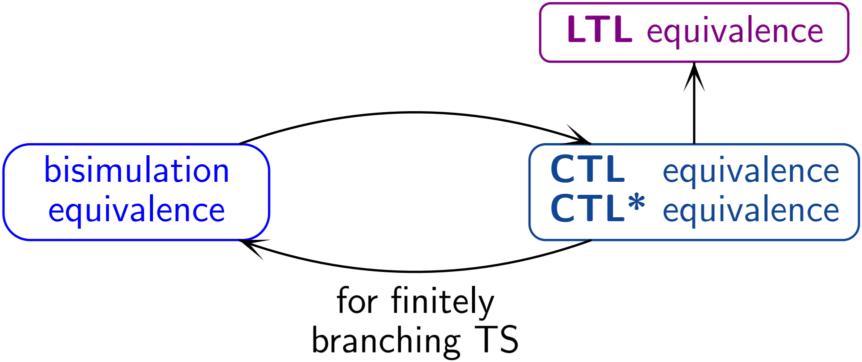
wrong.



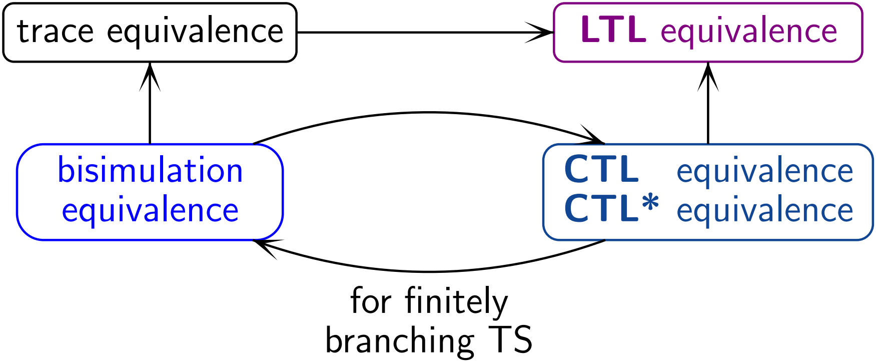
s_1, s_2 are trace equivalent and **LTL** equivalent

$$s_1 \models \exists\bigcirc(\exists\bigcirc a \wedge \exists\bigcirc b)$$
$$s_2 \not\models \exists\bigcirc(\exists\bigcirc a \wedge \exists\bigcirc b)$$

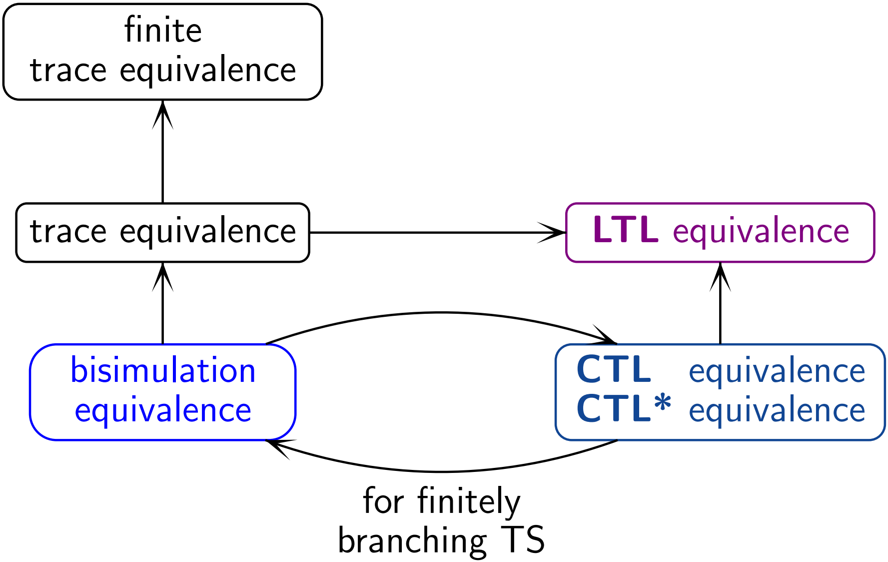
Summary: equivalences



Summary: equivalences



Summary: equivalences



Summary: equivalences