

Introduction

Modelling parallel systems

Linear Time Properties

Regular Properties

Linear Temporal Logic (LTL)

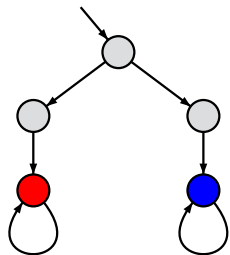
Computation-Tree Logic

Equivalences and Abstraction

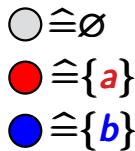
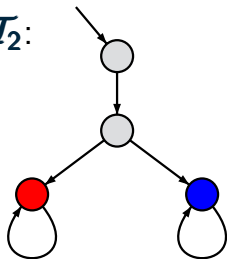
Trace equivalence

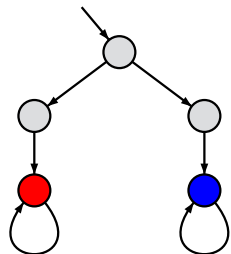
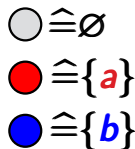
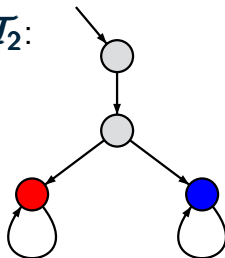
BSEQOR5.1-2

\mathcal{T}_1 :

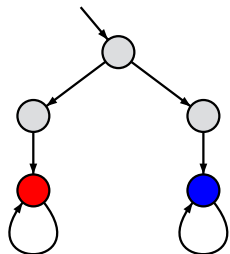
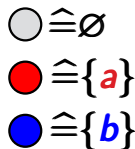
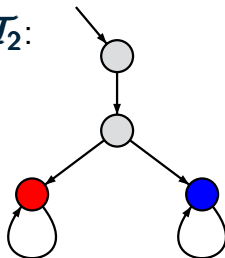


\mathcal{T}_2 :



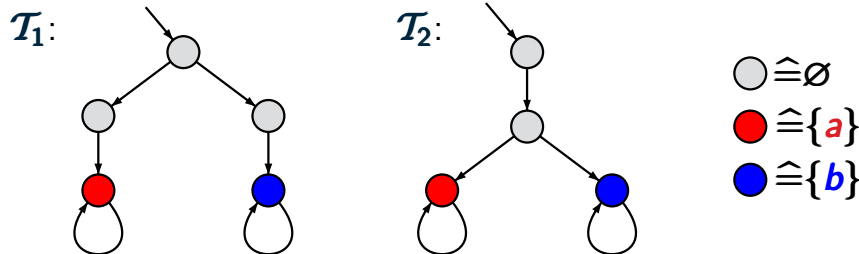
$\mathcal{T}_1:$  $\mathcal{T}_2:$ 

$$\text{Traces}(\mathcal{T}_1) = \{ \emptyset \emptyset a^\omega, \emptyset \emptyset b^\omega \} = \text{Traces}(\mathcal{T}_2)$$

$\mathcal{T}_1:$

 $\mathcal{T}_2:$


$$\text{Traces}(\mathcal{T}_1) = \{ \emptyset \emptyset a^\omega, \emptyset \emptyset b^\omega \} = \text{Traces}(\mathcal{T}_2)$$

$$\text{CTL-formula } \phi = \exists \text{O}(\exists \text{O} a \wedge \exists \text{O} b)$$



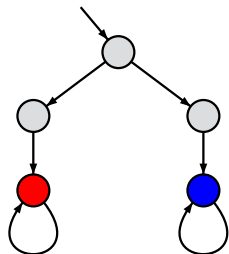
$$\text{Traces}(\mathcal{T}_1) = \{ \emptyset \emptyset a^\omega, \emptyset \emptyset b^\omega \} = \text{Traces}(\mathcal{T}_2)$$

$$\text{CTL-formula } \phi = \exists \text{O}(\exists \text{O}a \wedge \exists \text{O}b)$$

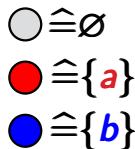
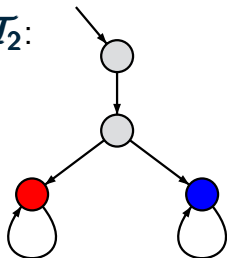
$$\mathcal{T}_1 \not\models \phi \quad \text{and} \quad \mathcal{T}_2 \models \phi$$

Trace equivalence is not compatible with CTL BSEQOR5.1-2

\mathcal{T}_1 :



\mathcal{T}_2 :



$$\text{Traces}(\mathcal{T}_1) = \{ \emptyset \emptyset a^\omega, \emptyset \emptyset b^\omega \} = \text{Traces}(\mathcal{T}_2)$$

$$\text{CTL-formula } \phi = \exists \text{O}(\exists \text{O}a \wedge \exists \text{O}b)$$

$$\mathcal{T}_1 \not\models \phi \quad \text{and} \quad \mathcal{T}_2 \models \phi$$

- for the **design** of complex systems
 - ↪ comparison of **2** transition systems

- for the **design** of complex systems
 - ↪ comparison of **2** transition systems
- for the **analysis** of complex systems

- for the **design** of complex systems
 - ↪ comparison of **2** transition systems
- for the **analysis** of complex systems
 - ↪ homogeneous model checking approach

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

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

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

- for the **design** of complex systems
 - ↪ comparison of **2** transition systems
- for the **analysis** of complex systems
 - ↪ homogeneous model checking approach
 - ↪ **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

finite trace inclusion and equivalence:

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

trace inclusion and trace equivalence:

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

finite trace inclusion and equivalence:

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

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

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

finite trace inclusion and equivalence:

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

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

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

preserves all **LTL** properties

finite trace inclusion and equivalence:

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

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

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

preserves all **LTL** properties

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

finite trace inclusion and equivalence:

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

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

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

preserves all **LTL** properties

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

finite trace inclusion and equivalence:

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

preserves all linear-time **safety** properties

trace inclusion and trace equivalence:

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

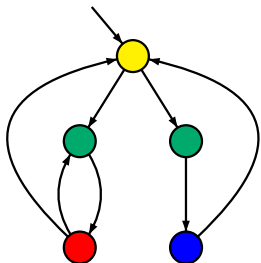
preserves all **LTL** properties

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

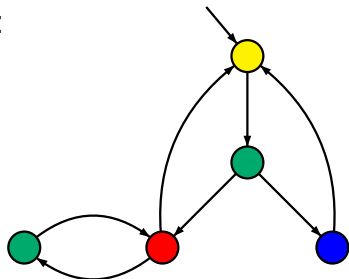
Minimization w.r.t. trace equivalence?

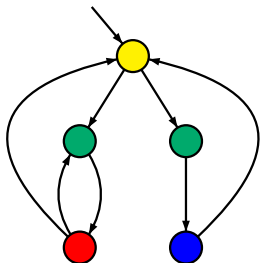
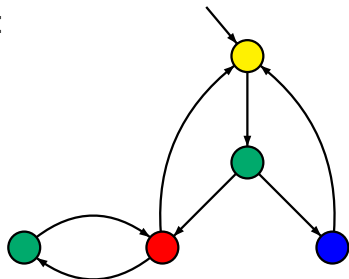
BSEQOR5.1-MIN-LT

\mathcal{T}_1 :

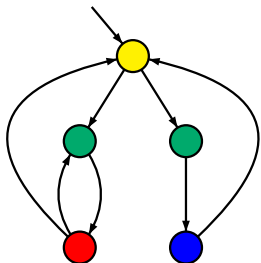
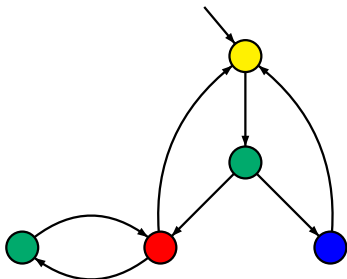


\mathcal{T}_2 :

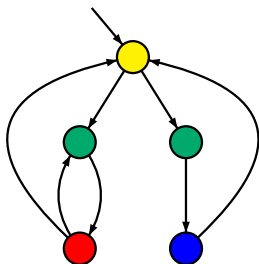
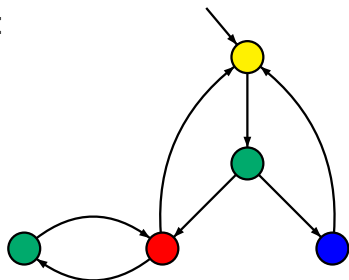


$\mathcal{T}_1:$  $\mathcal{T}_2:$ 

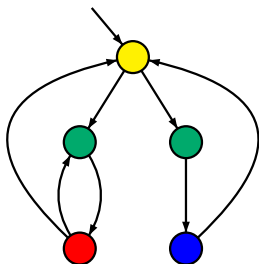
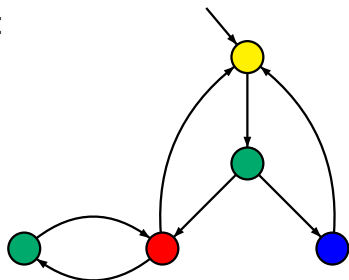
- $Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$

\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

$\mathcal{T}_1:$

 $\mathcal{T}_2:$


- $Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$
but \mathcal{T}_1 and \mathcal{T}_2 are not isomorphic
- $\mathcal{T}_1, \mathcal{T}_2$ have **5 states** and **7 transitions** each

$\mathcal{T}_1:$

 $\mathcal{T}_2:$


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

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

- **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**

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

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

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

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

be two transition systems

- with the same set AP

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.

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

be two transition systems

- with the same set AP ← 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.

binary relation $\mathcal{R} \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

binary relation $\mathcal{R} \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

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

binary relation $\mathcal{R} \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

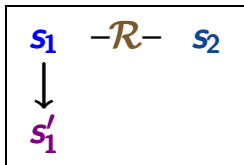
(1) $L_1(s_1) = L_2(s_2)$

(2) $\forall s'_1 \in \text{Post}(s_1) \exists s'_2 \in \text{Post}(s_2)$ s.t. $(s'_1, s'_2) \in \mathcal{R}$

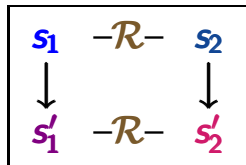
binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(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. $(s'_1, s'_2) \in \mathcal{R}$



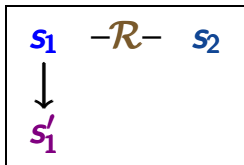
can be
completed to



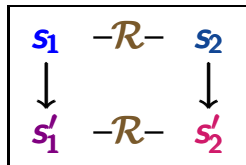
binary relation $\mathcal{R} \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

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

$$(2) \quad \forall s'_1 \in \text{Post}(s_1) \exists s'_2 \in \text{Post}(s_2) \text{ s.t. } (s'_1, s'_2) \in \mathcal{R}$$



can be
completed to

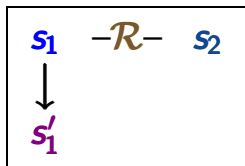


$$(3) \quad \forall s'_2 \in \text{Post}(s_2) \exists s'_1 \in \text{Post}(s_1) \text{ s.t. } (s'_1, s'_2) \in \mathcal{R}$$

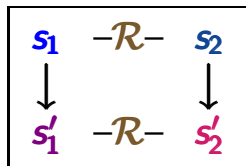
binary relation $\mathcal{R} \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

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

$$(2) \quad \forall s'_1 \in \text{Post}(s_1) \exists s'_2 \in \text{Post}(s_2) \text{ s.t. } (s'_1, s'_2) \in \mathcal{R}$$



can be
completed to



$$(3) \quad \forall s'_2 \in \text{Post}(s_2) \exists s'_1 \in \text{Post}(s_1) \text{ s.t. } (s'_1, s'_2) \in \mathcal{R}$$

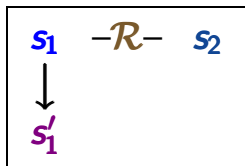
and such that the following initial condition holds:

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

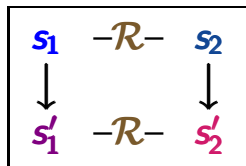
binary relation $\mathcal{R} \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

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

$$(2) \quad \forall s'_1 \in \text{Post}(s_1) \exists s'_2 \in \text{Post}(s_2) \text{ s.t. } (s'_1, s'_2) \in \mathcal{R}$$



can be
completed to



$$(3) \quad \forall s'_2 \in \text{Post}(s_2) \exists s'_1 \in \text{Post}(s_1) \text{ s.t. } (s'_1, s'_2) \in \mathcal{R}$$

and such that the following initial condition holds:

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

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

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

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

- (1) labeling condition
- (2) } mutual stepwise
- (3) } simulation

and initial condition (I)

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

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

- (1) labeling condition
- (2) } mutual stepwise
- (3) } simulation

and initial condition (I)

bisimulation equivalence \sim for TS:

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

- for all $(s_1, s_2) \in \mathcal{R}$:
- (1) labeling condition
 - (2) } mutual stepwise
 - (3) } 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 $\mathcal{R} \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ s.t.

- for all $(s_1, s_2) \in \mathcal{R}$:
- (1) labeling condition
 - (2) } mutual stepwise
 - (3) } simulation

and initial condition (I)

bisimulation equivalence \sim for TS:

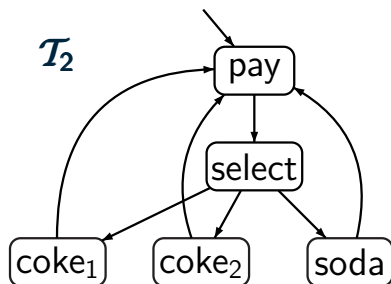
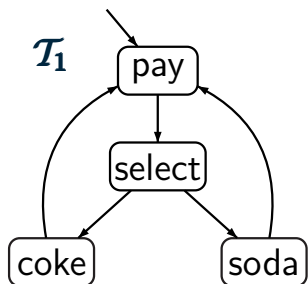
$\mathcal{T}_1 \sim \mathcal{T}_2$ iff there is a bisimulation \mathcal{R} for $(\mathcal{T}_1, \mathcal{T}_2)$

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 $(s_1, s_2) \in \mathcal{R}$

Two beverage machines

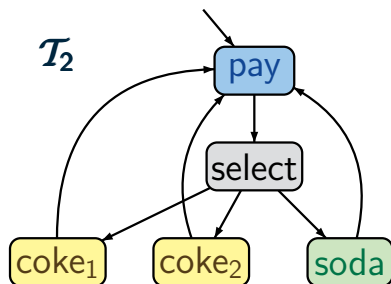
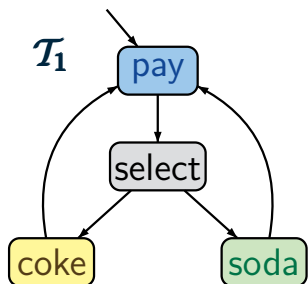
BSEQOR5.1-8-BIS



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

Two beverage machines

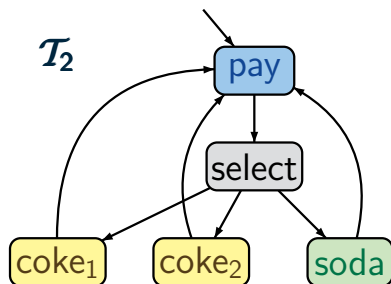
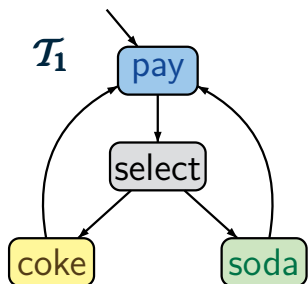
BSEQOR5.1-8-BIS



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

Two beverage machines

BSEQOR5.1-8-BIS

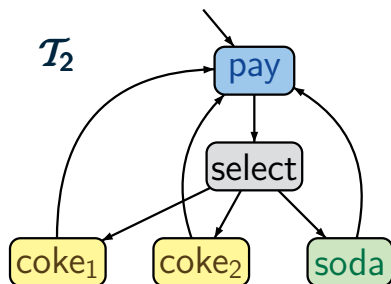
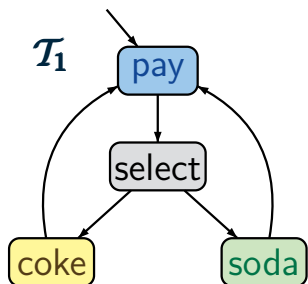


$$AP = \{ \textit{pay}, \textit{coke}, \textit{soda} \}$$

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

Two beverage machines

BSEQOR5.1-8-BIS

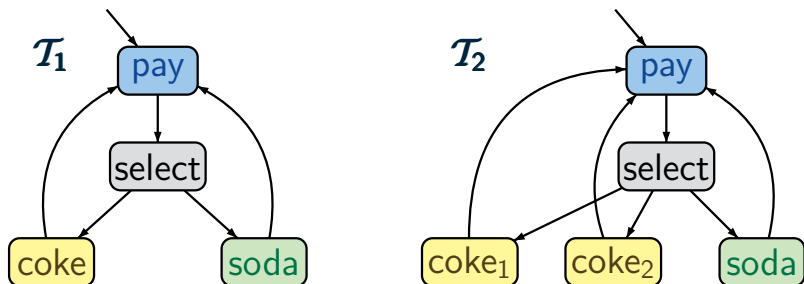


$$AP = \{ \text{pay}, \text{coke}, \text{soda} \}$$

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

Two beverage machines

BSEQOR5.1-8-BIS



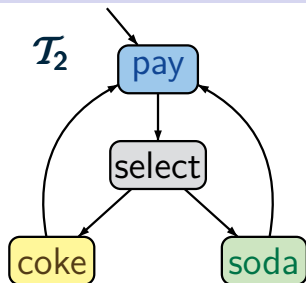
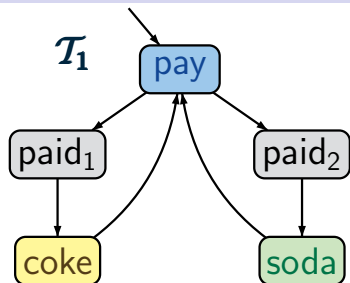
$$AP = \{ \text{pay}, \text{coke}, \text{soda} \}$$

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

$$\left\{ \begin{array}{l} (\text{pay}, \text{pay}), (\text{select}, \text{select}), (\text{soda}, \text{soda}) \\ (\text{coke}, \text{coke}_1), (\text{coke}, \text{coke}_2) \end{array} \right\}$$

Two beverage machines

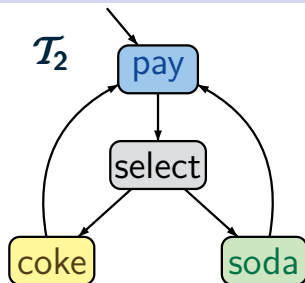
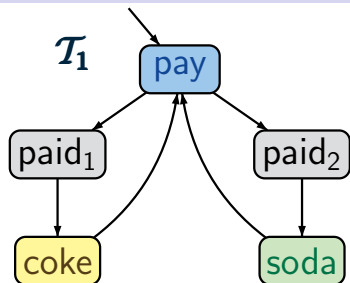
BSEQOR5.1-8-BIS-3



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

Two beverage machines

BSEQOR5.1-8-BIS-3

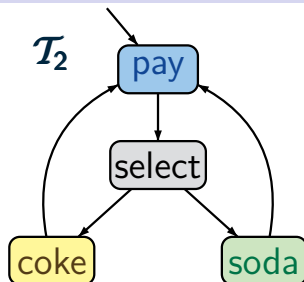
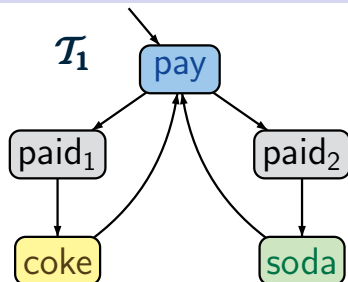


$AP = \{pay, coke, soda\}$

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

Two beverage machines

BSEQOR5.1-8-BIS-3

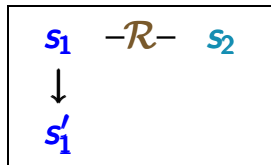


$AP = \{pay, coke, soda\}$

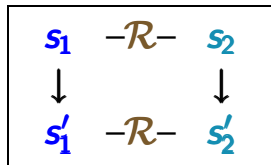
$\mathcal{T}_1 \not\sim \mathcal{T}_2$

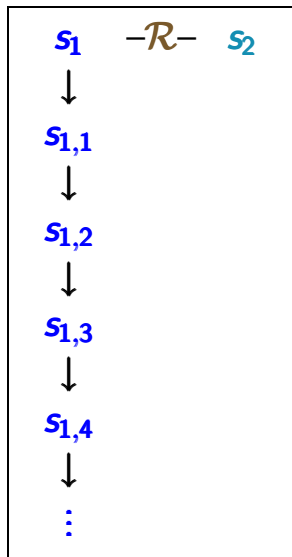
because there is no state in \mathcal{T}_1 that has both

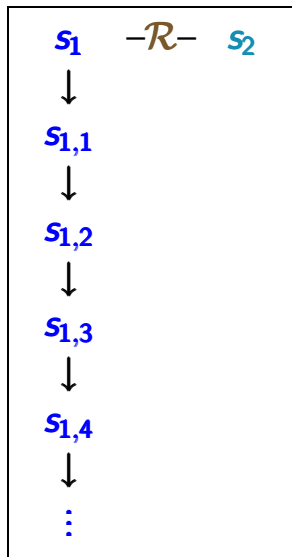
- a successor labeled with **coke** and
- a successor labeled with **soda**



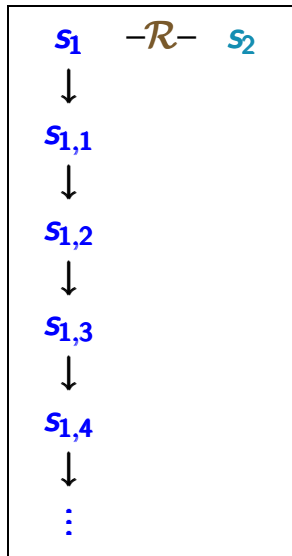
can be
completed to



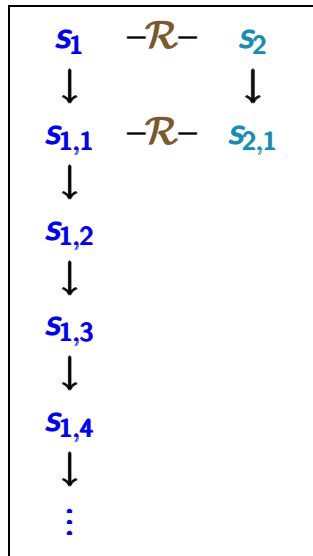


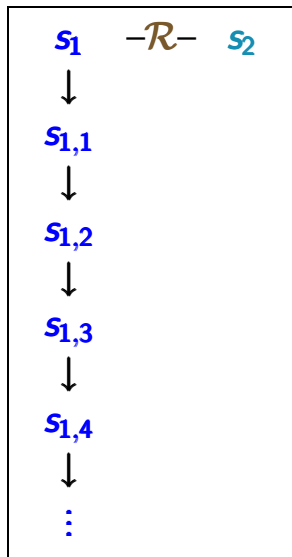


can be
completed to

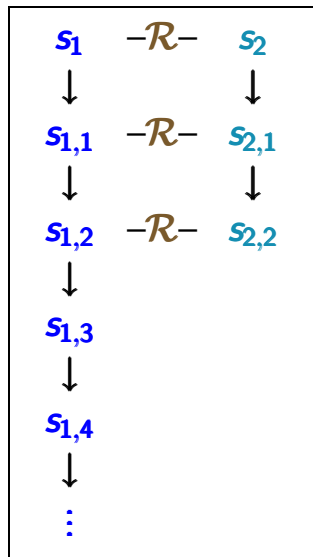


can be
completed to



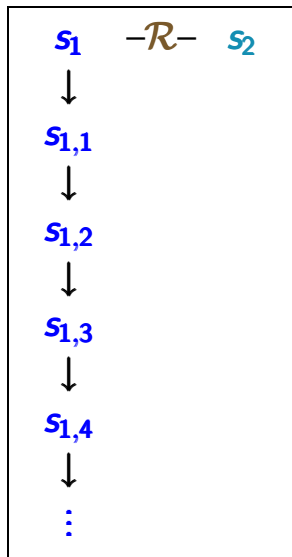


can be
completed to

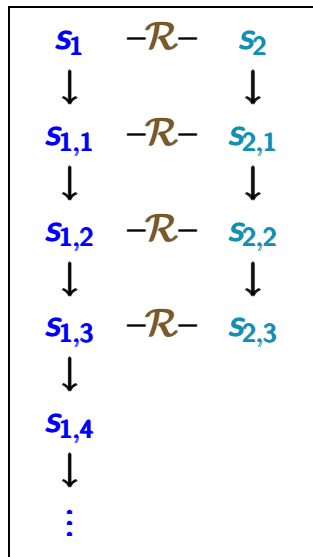


Path lifting for bisimulation \mathcal{R}

BSEQOR5.1-9-BIS

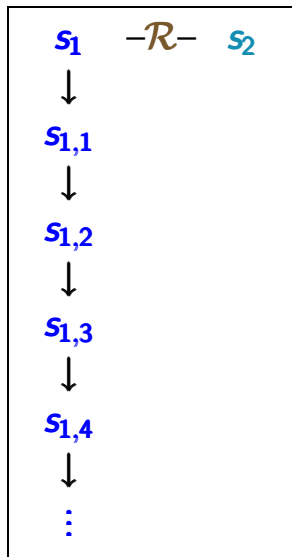


can be
completed to

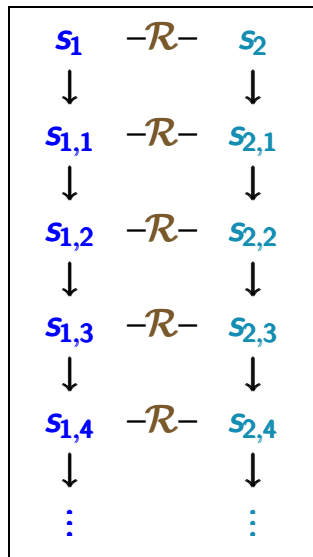


Path lifting for bisimulation \mathcal{R}

BSEQOR5.1-9-BIS



can be
completed to



\sim is an **equivalence**

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

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

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

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



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

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

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

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

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

\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 $(\mathcal{T}_2, \mathcal{T}_3)$.

\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 $\mathcal{T}_1 \sim \mathcal{T}_3$



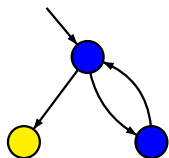
Let $\mathcal{R}_{1,2}$ be a bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$,
 $\mathcal{R}_{2,3}$ be a bisimulation for $(\mathcal{T}_2, \mathcal{T}_3)$.

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

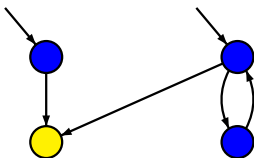
is a bisimulation for $(\mathcal{T}_1, \mathcal{T}_3)$

Correct or wrong?

BSEQOR5.1-19

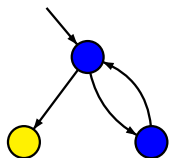


~

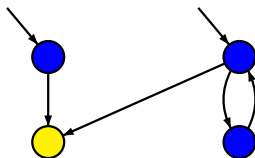


Correct or wrong?

BSEQOR5.1-19



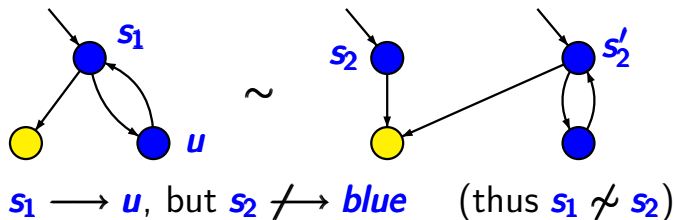
~



wrong

Correct or wrong?

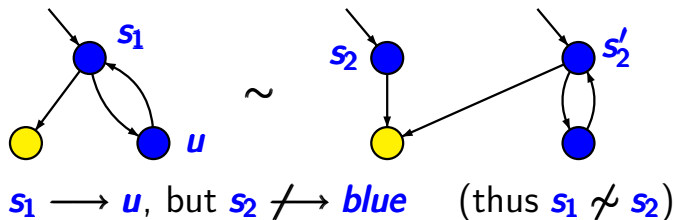
BSEQOR5.1-19



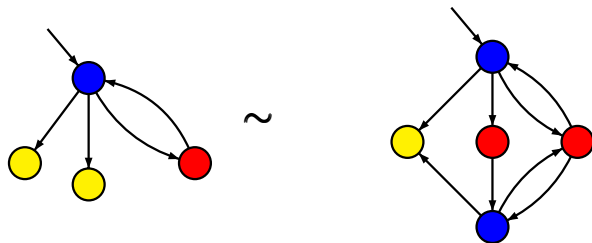
wrong

Correct or wrong?

BSEQOR5.1-19

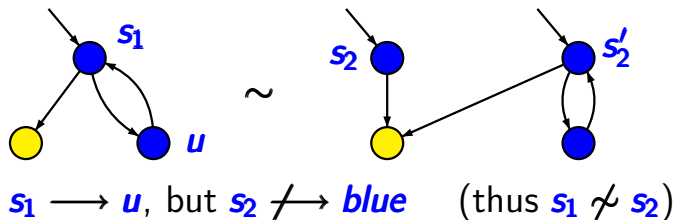


wrong

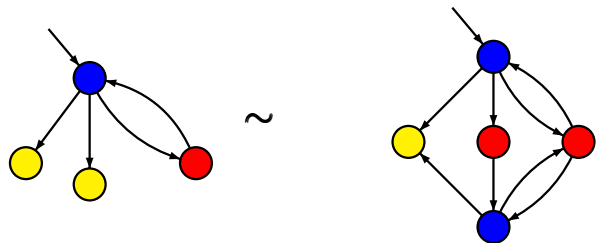


Correct or wrong?

BSEQOR5.1-19



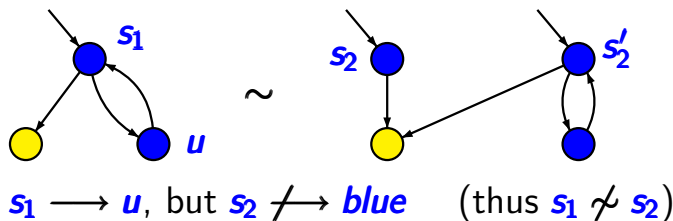
wrong



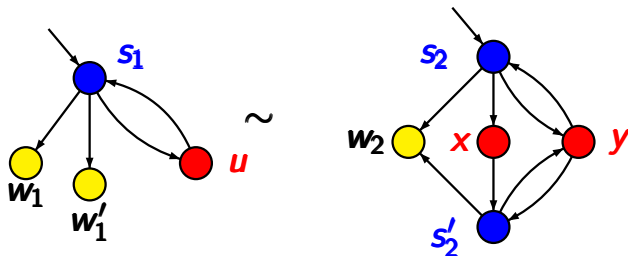
correct

Correct or wrong?

BSEQOR5.1-19



wrong

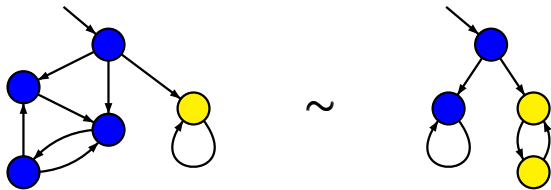


correct

$\{(w_1, w_2), (w'_1, w_2), (s_1, s_2), (s_1, s'_2), (u, x), (u, y)\}$

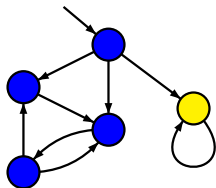
Correct or wrong?

BSEQOR5.1-20

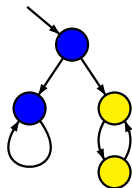


Correct or wrong?

BSEQOR5.1-20



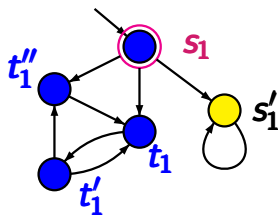
~



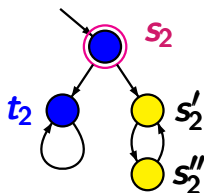
correct

Correct or wrong?

BSEQOR5.1-20



\sim



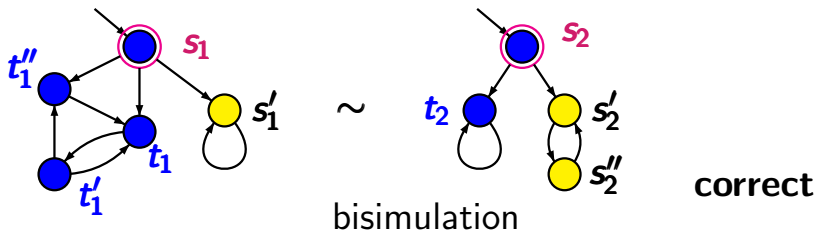
bisimulation

correct

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

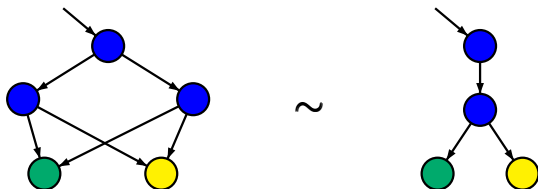
Correct or wrong?

BSEQOR5.1-20



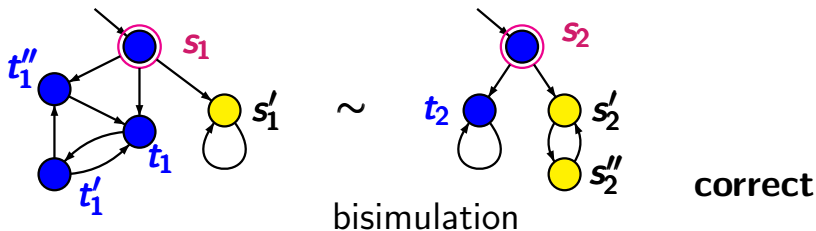
bisimulation

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



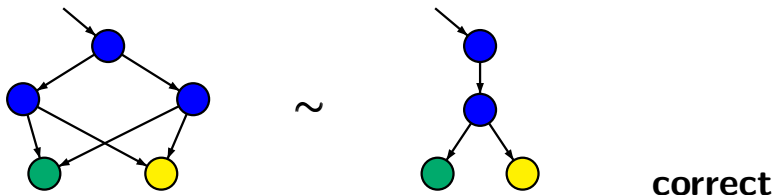
Correct or wrong?

BSEQOR5.1-20



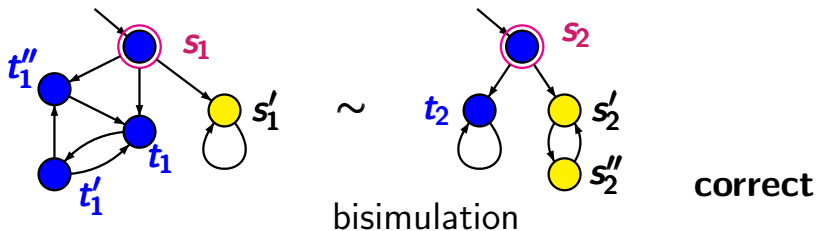
bisimulation

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



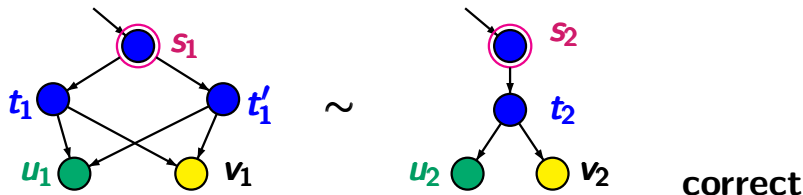
Correct or wrong?

BSEQOR5.1-20



bisimulation

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



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

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

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

proof: ... path fragment lifting ...

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

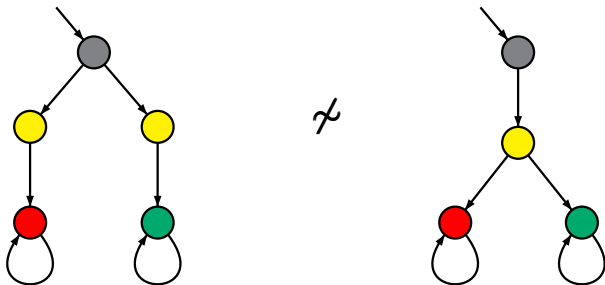
proof: ... path fragment lifting ...

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

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

proof: ... path fragment lifting ...

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



trace equivalent, but not bisimulation equivalent

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

proof: ... path fragment lifting ...

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

Trace equivalence is **strictly coarser** than bisimulation equivalence.

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

proof: ... path fragment lifting ...

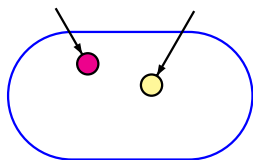
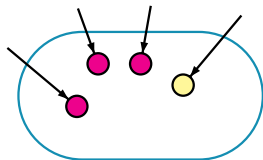
$$\text{Traces}(\mathcal{T}_1) = \text{Traces}(\mathcal{T}_2) \not\Rightarrow \mathcal{T}_1 \sim \mathcal{T}_2$$

Trace equivalence is **strictly coarser** than bisimulation equivalence.

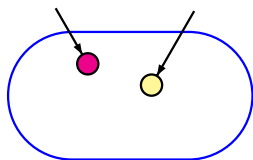
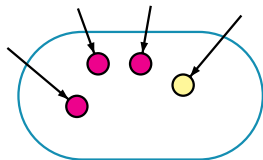
Bisimulation equivalent transition systems satisfy the **same LT properties** (e.g., **LTL formulas**).

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

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

 \mathcal{T}_1  \mathcal{T}_2 

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

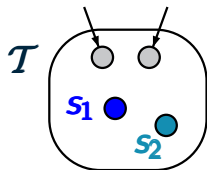
 \mathcal{T}_1  \mathcal{T}_2 

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

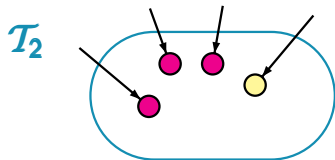
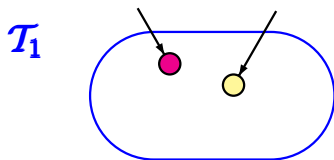
- as a relation that compares **2** transition systems



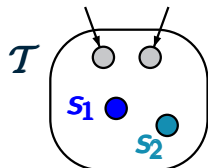
- as a relation on the **states** of **1** transition system



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



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

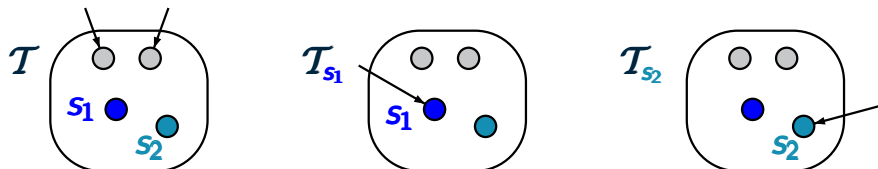


$$s_1 \sim s_2 \text{ iff } \mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$$

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



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

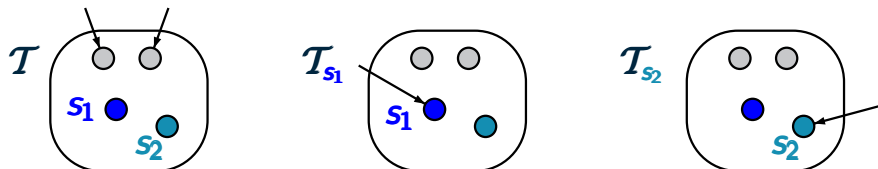


$$s_1 \sim s_2 \text{ iff } \mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$$

- 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. $(s_1, s_2) \in \mathcal{R}$

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

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) $\forall s'_1 \in \text{Post}(s_1) \exists s'_2 \in \text{Post}(s_2)$ s.t. $(s'_1, s'_2) \in \mathcal{R}$
- (3) $\forall s'_2 \in \text{Post}(s_2) \exists s'_1 \in \text{Post}(s_1)$ 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) $\forall s'_1 \in \text{Post}(s_1) \exists s'_2 \in \text{Post}(s_2)$ s.t. $(s'_1, s'_2) \in \mathcal{R}$
- (3) $\forall s'_2 \in \text{Post}(s_2) \exists s'_1 \in \text{Post}(s_1)$ 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 \mathcal{S} .

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

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

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

- the coarsest bisimulation on \mathcal{T}

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

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

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

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

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

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

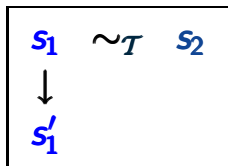
Bisimulation equivalence $\sim_{\mathcal{T}}$ is the coarsest equivalence on \mathcal{S} s.t. for all states $s_1, s_2 \in \mathcal{S}$ with $s_1 \sim_{\mathcal{T}} s_2$:

- (1) $L(s_1) = L(s_2)$
- (2) each transition of s_1 can be mimicked by a transition of s_2 :

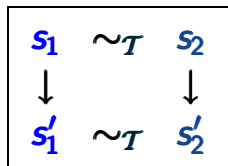
Let \mathcal{T} be a transition system with state space \mathcal{S} .

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



can be
completed to



- \sim relation that compares **2** transition systems
- $\sim_{\mathcal{T}}$ equivalence on the state space of a single TS \mathcal{T}

- \sim relation that compares **2** transition systems
 - $\sim_{\mathcal{T}}$ equivalence on the state space of a single TS \mathcal{T}
- $\sim_{\mathcal{T}}$ 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} :

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


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} :

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



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} :

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



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

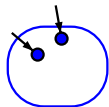
2. \sim can be derived from $\sim_{\mathcal{T}}$

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

BSEQOR5.1-31

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

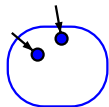


Derivation of \sim from $\sim_{\mathcal{T}}$

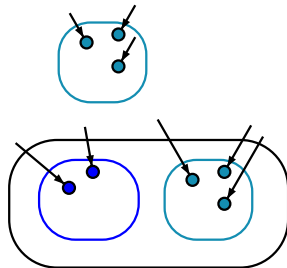
BSEQOR5.1-31

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

\mathcal{T}_1 with state space S_1



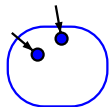
\mathcal{T}_2 with state space S_2



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

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

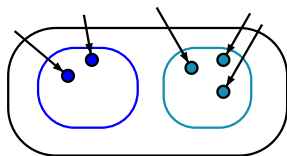
\mathcal{T}_1 with state space S_1



\mathcal{T}_2 with state space S_2



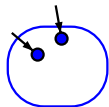
consider $\mathcal{T} = \mathcal{T}_1 \uplus \mathcal{T}_2$
(state space $S_1 \uplus S_2$)



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

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

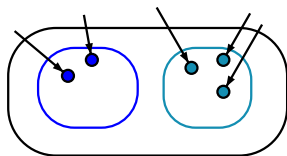
\mathcal{T}_1 with state space S_1



\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

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

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

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

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 $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, \text{AP}, L)$ be a TS.

bisimulation quotient:

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

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



set of bisimulation equivalence classes

Let $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, \text{AP}, L)$ be a TS.

bisimulation quotient:

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

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

Let $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, AP, L)$ be a TS.

bisimulation quotient:

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

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

Let $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, \text{AP}, L)$ be a TS.

bisimulation quotient:

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

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

well-defined

by the labeling condition
of bisimulations

Let $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, AP, L)$ be a TS.

bisimulation quotient:

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

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

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

Let $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, \text{AP}, L)$ be a TS.

bisimulation quotient:

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

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

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

action labels
irrelevant

Let $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, AP, L)$ be a TS.

bisimulation quotient:

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

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

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

action labels
irrelevant

Let $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, AP, L)$ be a TS.

bisimulation quotient:

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

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

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

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

Example: interleaving of n printers

BSEQOR5.1-34

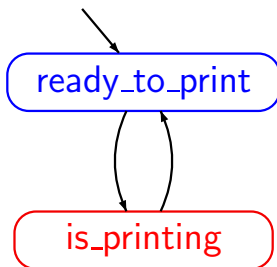
parallel system $\mathcal{T} = \underbrace{\textit{Printer} ||| \textit{Printer} ||| \dots ||| \textit{Printer}}_{n \text{ printer}}$

Example: interleaving of n printers

BSEQOR5.1-34

parallel system $\mathcal{T} = \underbrace{Printer \parallel\parallel Printer \parallel\parallel \dots \parallel\parallel Printer}_{n \text{ printer}}$

transition system
for each printer



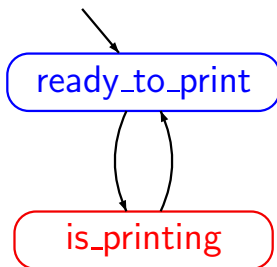
Example: interleaving of n printers

BSEQOR5.1-34

parallel system $\mathcal{T} = \underbrace{Printer \parallel\parallel Printer \parallel\parallel \dots \parallel\parallel Printer}_{n \text{ printer}}$

$AP = \{0, 1, \dots, n\}$ “number of available printers”

transition system
for each printer

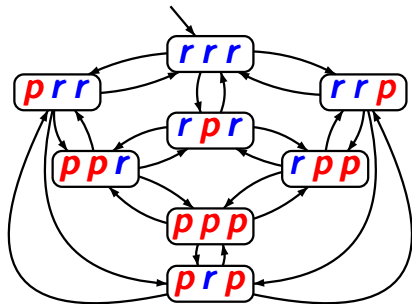


Example: $n=3$ printers

BSEQOR5.1-34

parallel system $\mathcal{T} = \underbrace{Printer \parallel\parallel Printer \parallel\parallel \dots \parallel\parallel Printer}_{n \text{ printer}}$

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



p: is printing

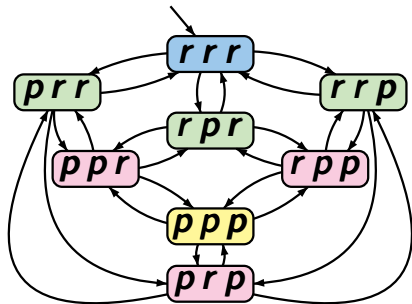
r: ready to print

Example: $n=3$ printers

BSEQOR5.1-34

parallel system $\mathcal{T} = \underbrace{\text{Printer} \parallel \text{Printer} \parallel \dots \parallel \text{Printer}}_{n \text{ printer}}$

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



p : is printing

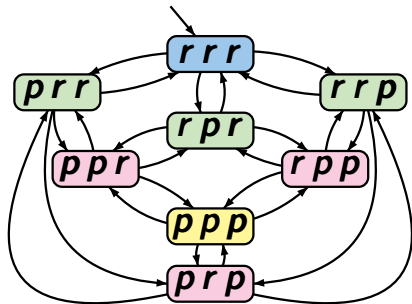
r : ready to print

Example: $n=3$ printers

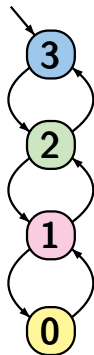
BSEQOR5.1-34

parallel system $\mathcal{T} = \underbrace{\text{Printer} \parallel \text{Printer} \parallel \dots \parallel \text{Printer}}_{n \text{ printer}}$

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



p : is printing
 r : ready to print



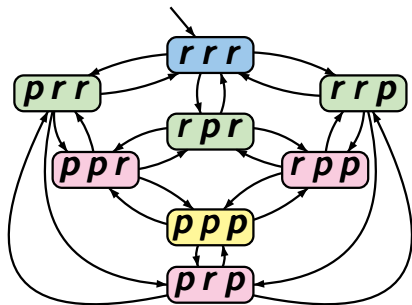
bisimulation
quotient

Example: $n=3$ printers

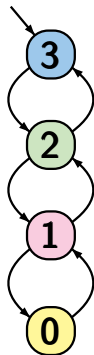
BSEQOR5.1-34

parallel system $\mathcal{T} = \underbrace{Printer \parallel\parallel Printer \parallel\parallel \dots \parallel\parallel Printer}_{n \text{ printer}}$

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



2^n states



$n+1$ states

solutions for mutual exclusion problems:

- semaphore
- Peterson's 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

solutions for mutual exclusion problems:

- semaphore
- Peterson's algorithm
- Bakery algorithm



given two concurrent processes P_1 and P_2

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

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

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

protocol for P_1 :

```
LOOP FOREVER
```

```
  noncritical actions
```

```
   $x_1 := x_2 + 1$ 
```

```
  AWAIT ( $x_1 < x_2$ )  $\vee$  ( $x_2 = 0$ );
```

```
  critical section;
```

```
   $x_1 := 0$ 
```

```
END LOOP
```

symmetric protocol for P_2

protocol for P_1 :

```
LOOP FOREVER
```

```
  noncritical actions
```

```
   $x_1 := x_2 + 1$ 
```

```
  AWAIT ( $x_1 < x_2$ )  $\vee$  ( $x_2 = 0$ );
```

```
  critical section;
```

```
   $x_1 := 0$ 
```

```
END LOOP
```

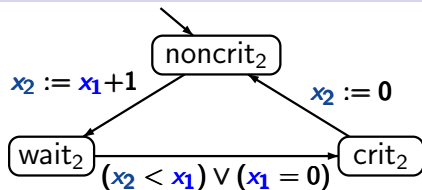
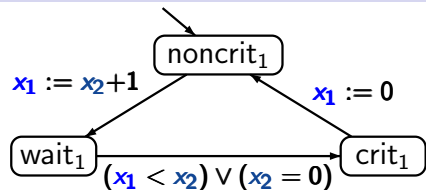
initially:

```
 $x_1 = x_2 = 0$ 
```

symmetric protocol for P_2

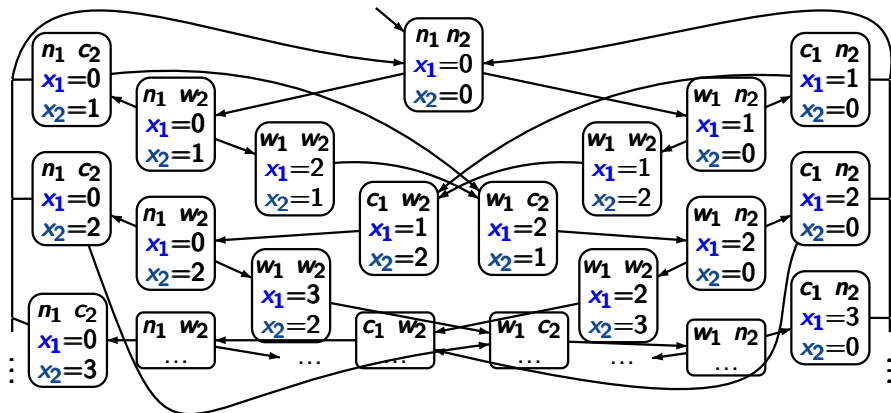
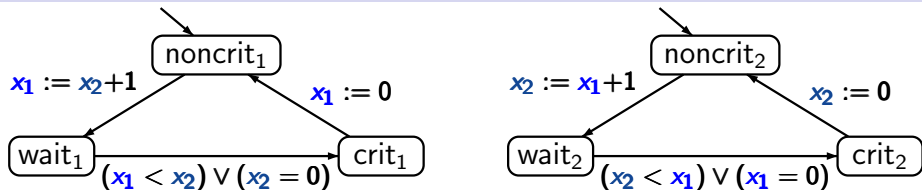
Program graphs for the Bakery algorithm

BSEQOR5.1-37



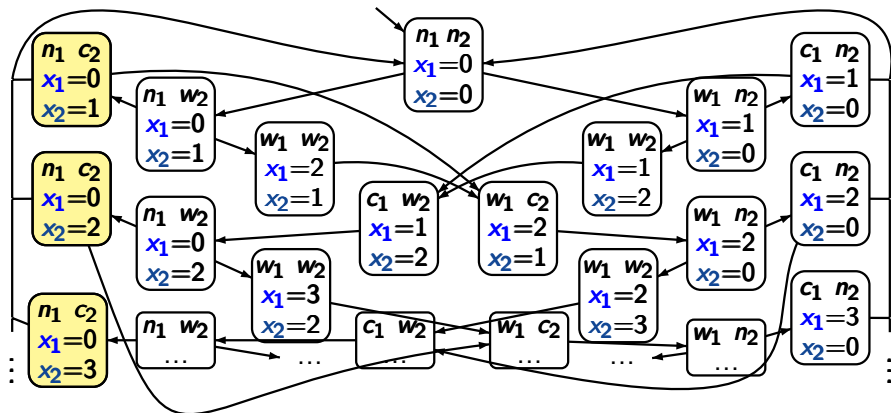
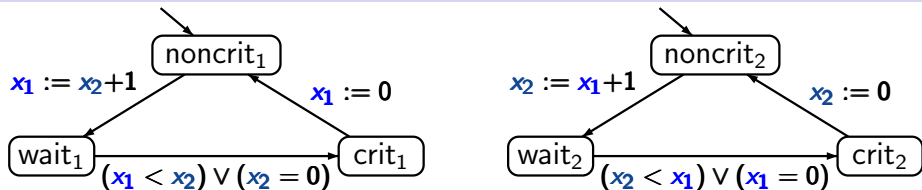
Transition system for the Bakery algorithm

BSEQOR5.1-37



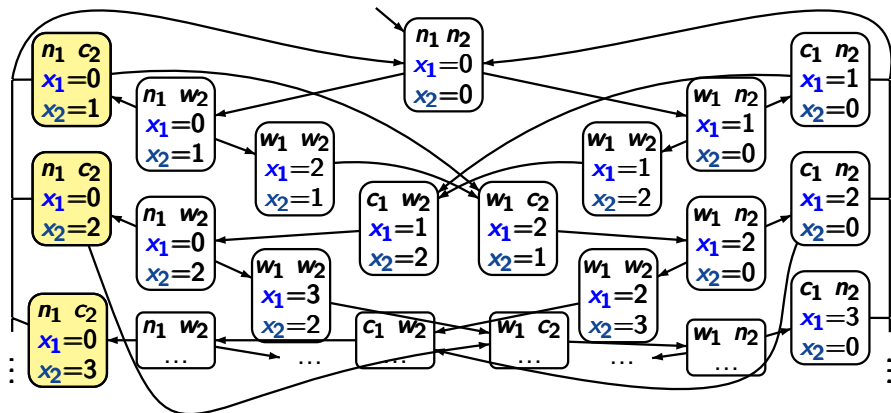
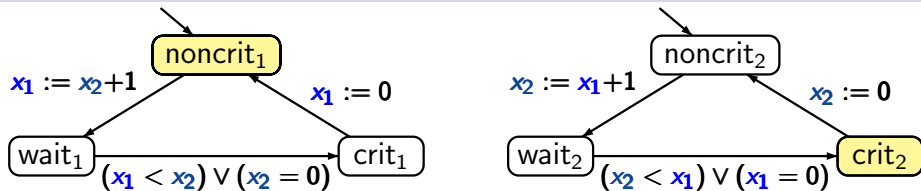
Transition system for the Bakery algorithm

BSEQOR5.1-37



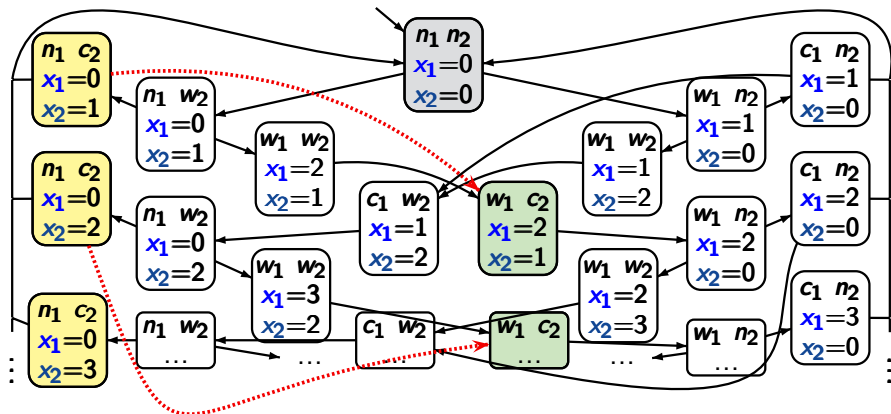
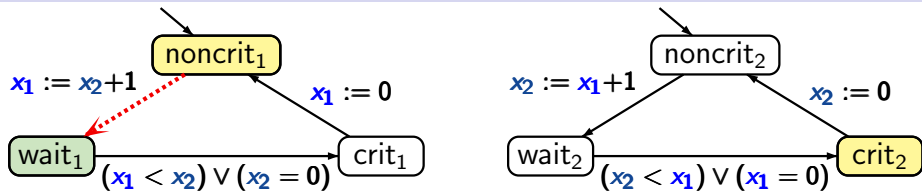
Transition system for the Bakery algorithm

BSEQOR5.1-37



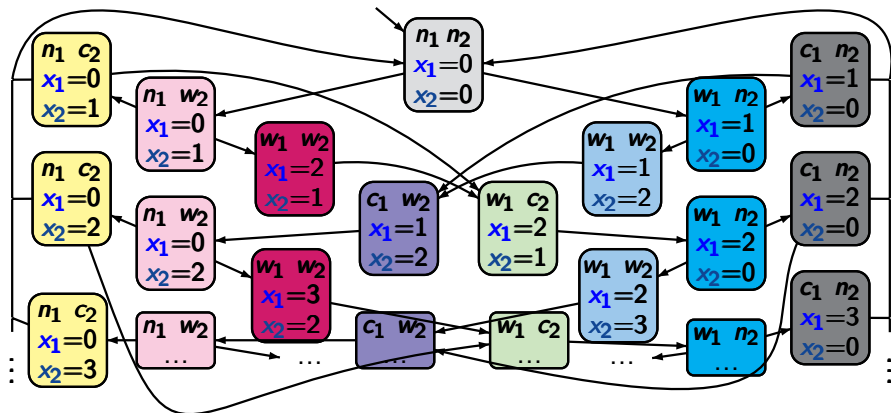
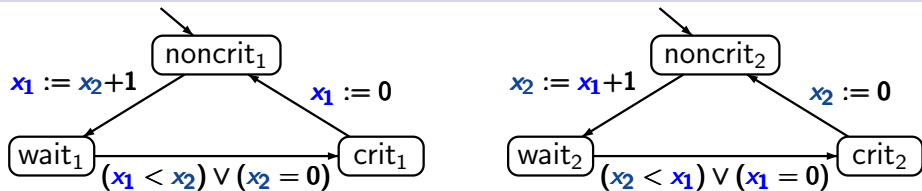
Transition system for the Bakery algorithm

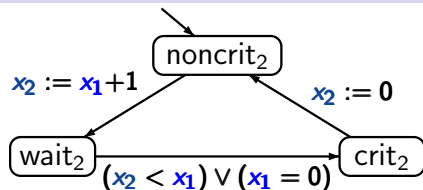
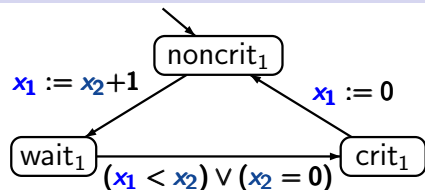
BSEQOR5.1-37



Transition system for the Bakery algorithm

BSEQOR5.1-37

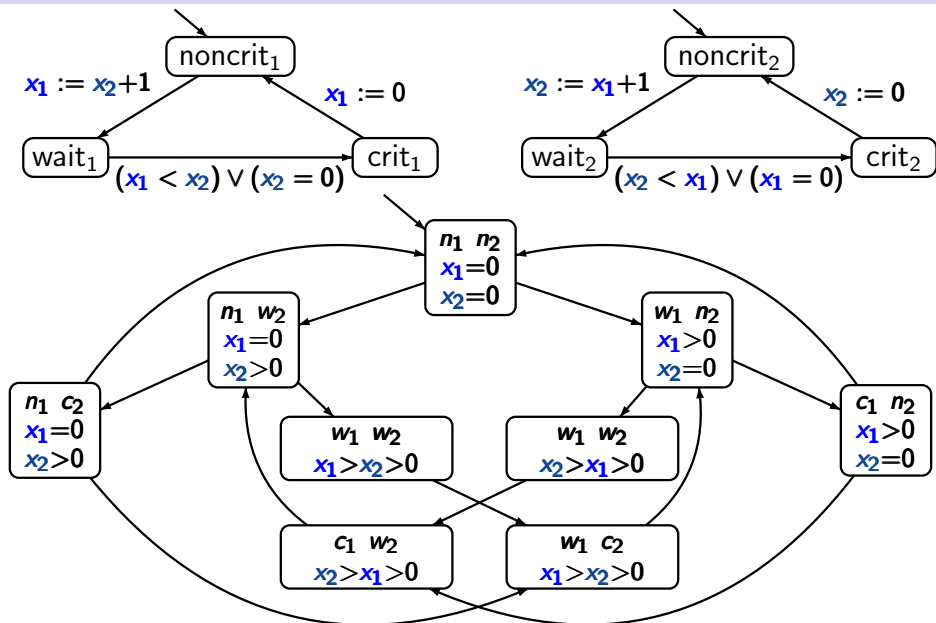




infinite transition system with a
finite bisimulation quotient

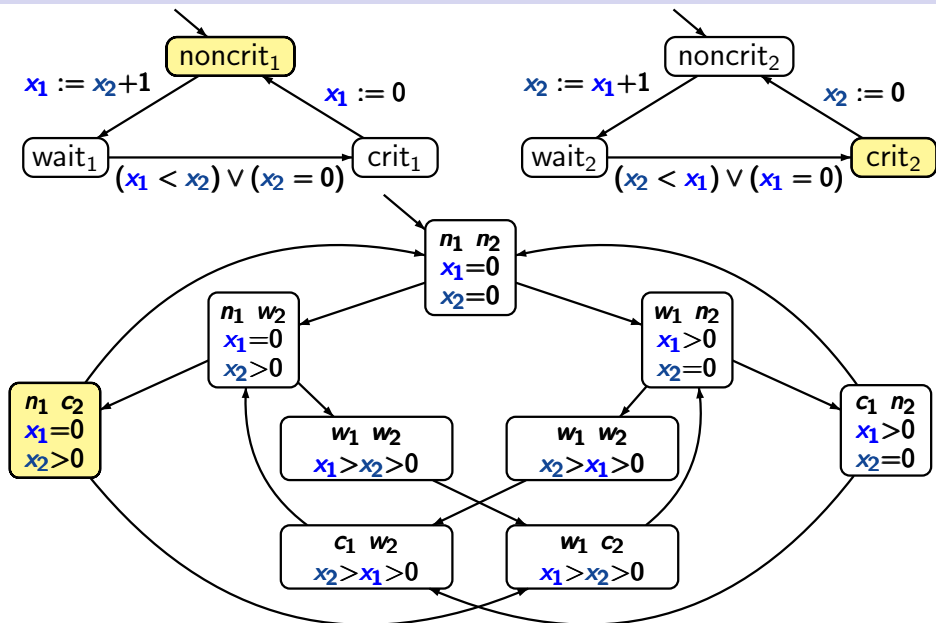
Bakery algorithm: bisimulation quotient

BSEQOR5.1-38



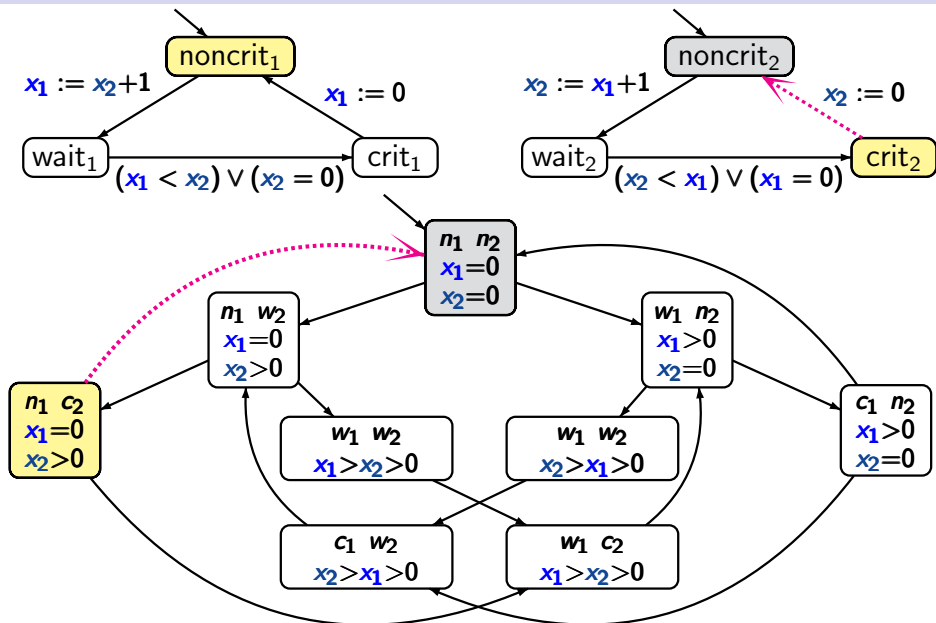
Bakery algorithm: bisimulation quotient

BSEQOR5.1-38



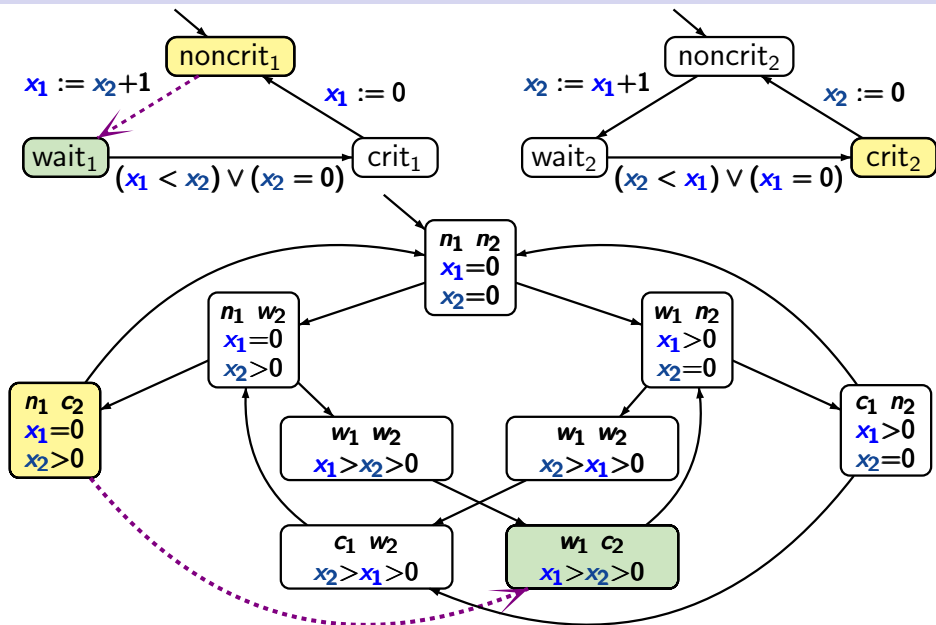
Bakery algorithm: bisimulation quotient

BSEQOR5.1-38



Bakery algorithm: bisimulation quotient

BSEQOR5.1-38



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

CTL* state formulas

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\psi$$

CTL* path formulas

$$\psi ::= \Phi \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \bigcirc\psi \mid \psi_1 \mathbf{U} \psi_2$$

CTL* state formulas

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\psi$$

CTL* path formulas

$$\psi ::= \Phi \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \bigcirc\psi \mid \psi_1 \mathbf{U} \psi_2$$

derived operators:

- \diamond, \square, \dots as in **LTL**

CTL* state formulas

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\psi$$

CTL* path formulas

$$\psi ::= \Phi \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \bigcirc\psi \mid \psi_1 \text{U} \psi_2$$

derived operators:

- \diamond, \square, \dots as in **LTL**
- universal quantification: $\forall\psi \stackrel{\text{def}}{=} \neg\exists\neg\psi$

CTL* state formulas

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\psi$$

CTL* path formulas

$$\psi ::= \Phi \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \bigcirc\psi \mid \psi_1 \mathbf{U} \psi_2$$

CTL: sublogic of **CTL***

CTL* state formulas

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \exists \psi$$

CTL* path formulas

$$\psi ::= \Phi \mid \psi_1 \wedge \psi_2 \mid \neg \psi \mid \bigcirc \psi \mid \psi_1 \mathbf{U} \psi_2$$

CTL: sublogic of **CTL***

- with path quantifiers \exists and \forall

CTL* state formulas

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\psi$$

CTL* path formulas

$$\psi ::= \Phi \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \bigcirc\psi \mid \psi_1 \text{U} \psi_2$$

CTL: sublogic of **CTL***

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

CTL* state formulas

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\psi$$

CTL* path formulas

$$\psi ::= \Phi \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \bigcirc\psi \mid \psi_1 \mathbf{U} \psi_2$$

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**

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

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

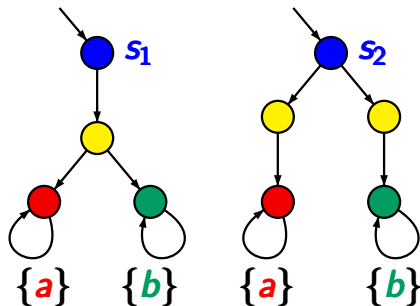
s_1, s_2 are **CTL** equivalent if for all **CTL** formulas ϕ :

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

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 :

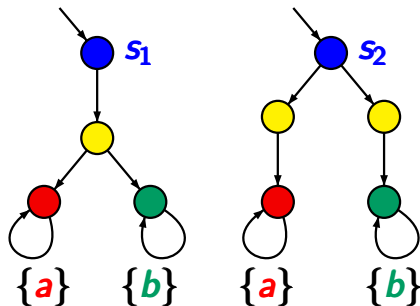
$$s_1 \models \Phi \quad \text{iff} \quad s_2 \models \Phi$$



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

$$s_1 \models \text{EO}(\text{EO}a \wedge \text{EO}b)$$

$$s_2 \not\models \text{EO}(\text{EO}a \wedge \text{EO}b)$$

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**

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 ψ :

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

bisimulation equivalence
= **CTL** equivalence
= **CTL*** equivalence

bisimulation equivalence
= CTL equivalence
= CTL* equivalence

← for finite TS

bisimulation equivalence
= CTL equivalence
= CTL* equivalence

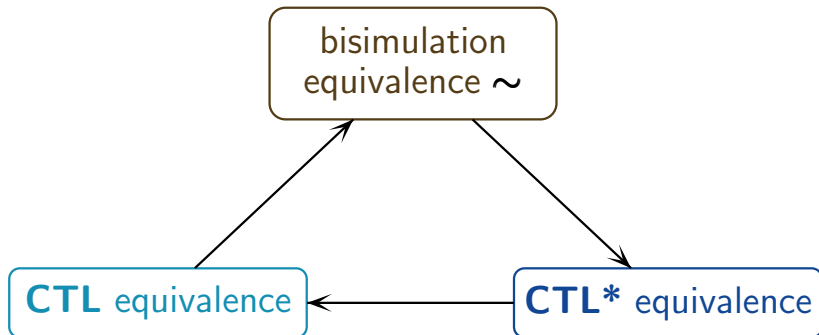
← for finite TS

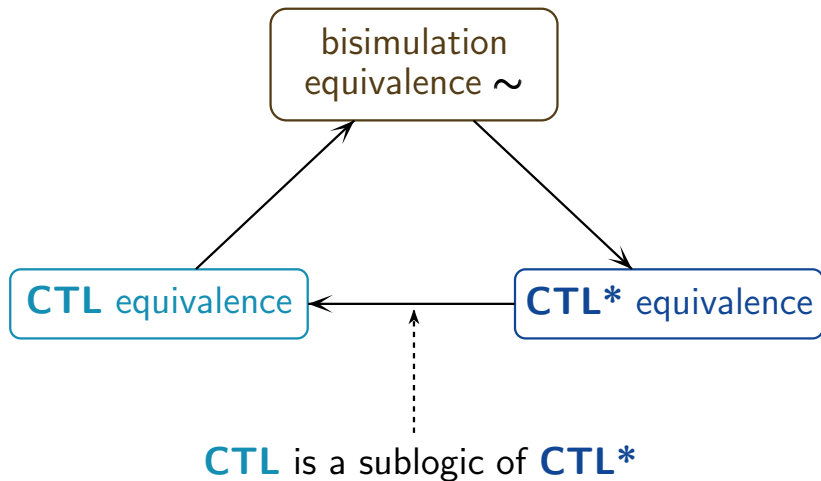
Let \mathcal{T} be a finite TS without terminal states,
and s_1, s_2 states in \mathcal{T} . Then:

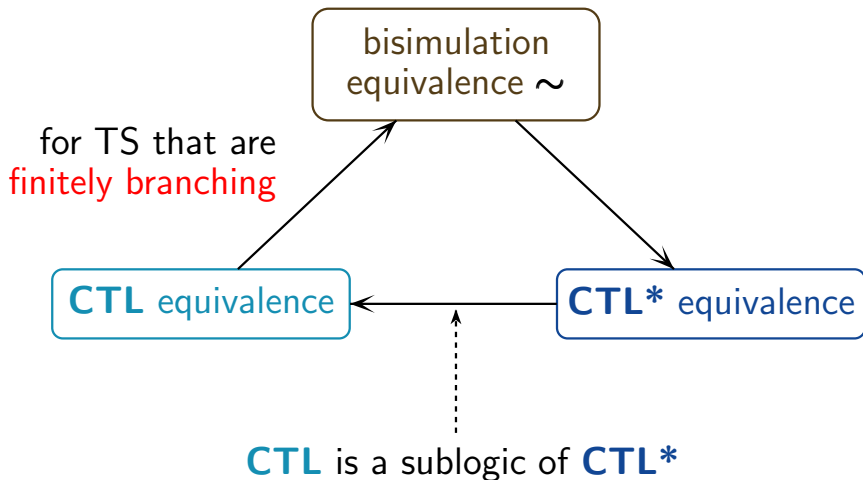
$$s_1 \sim_{\mathcal{T}} s_2$$

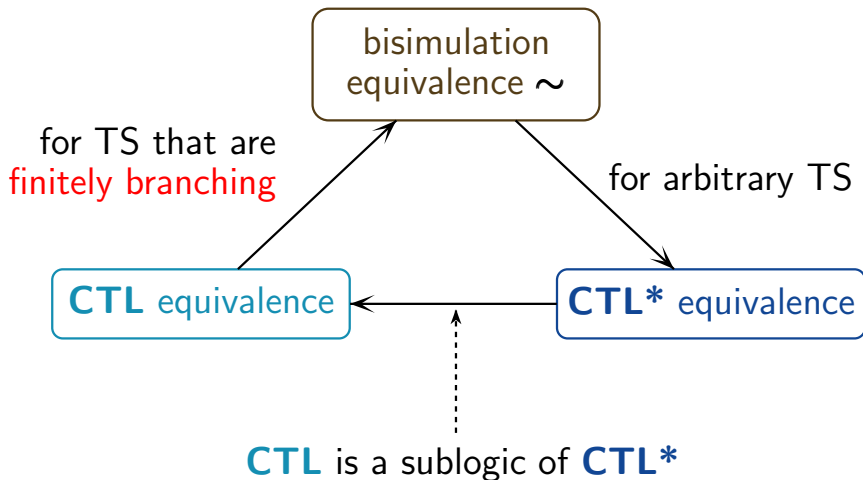
iff s_1 and s_2 are CTL equivalent

iff s_1 and s_2 are CTL* equivalent









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

CTL equivalence is finer than **LTL** equivalence

CTL equivalence is finer than **LTL** equivalence

correct.

CTL equivalence is finer than **LTL** equivalence

correct.



CTL equivalence = **CTL*** equivalence

LTL is sublogic of **CTL***

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

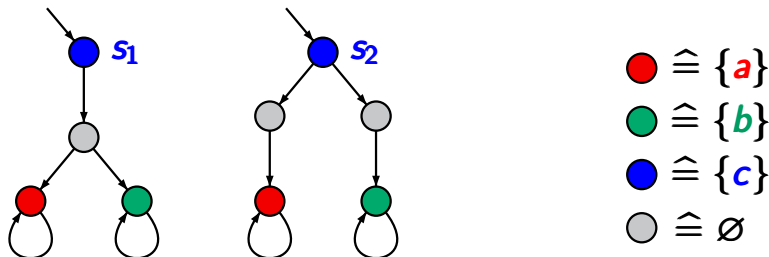
wrong.

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.

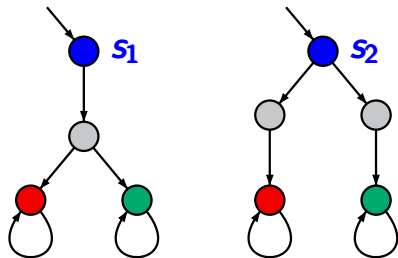


CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.



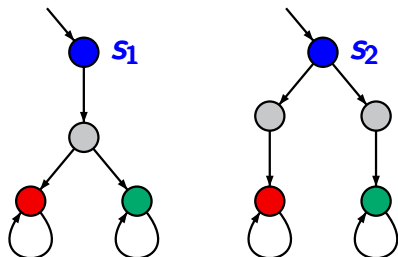
s_1, s_2 are trace equivalent

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.



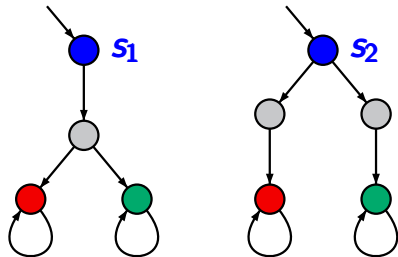
s_1 , s_2 are trace equivalent
and **LTL** equivalent

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

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

