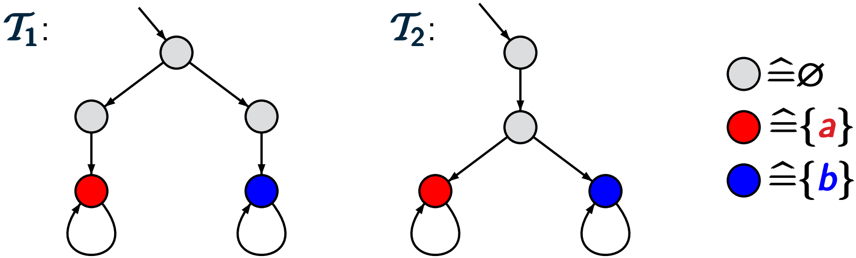
# Overview



- Introduction
- Modelling parallel systems
- Linear Time Properties
- Regular Properties
- Linear Temporal Logic (LTL)
- Computation-Tree Logic
- **Equivalences and Abstraction**

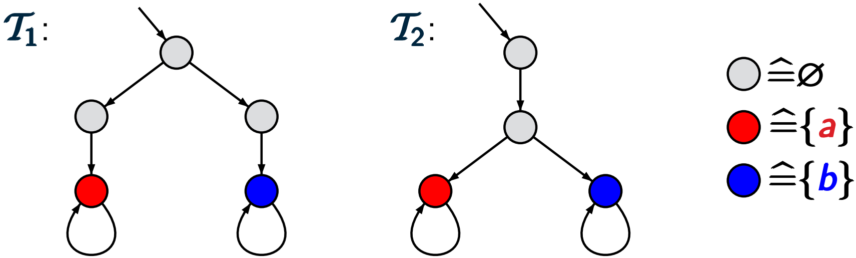
# Trace equivalence





# Trace equivalence

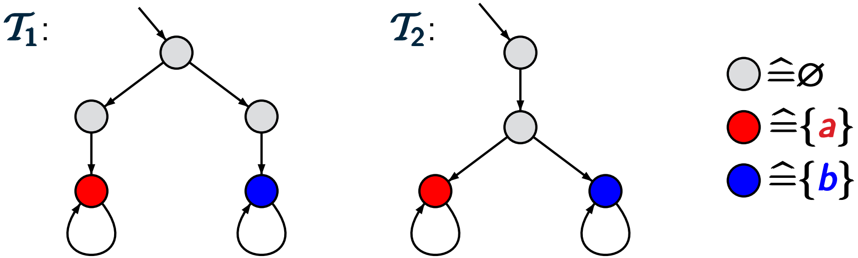





$$Traces(\mathcal{T}_1) = \{ \varnothing\, \varnothing\, a^\omega,\ \varnothing\, \varnothing\, b^\omega \} = Traces(\mathcal{T}_2)$$

# Trace equivalence



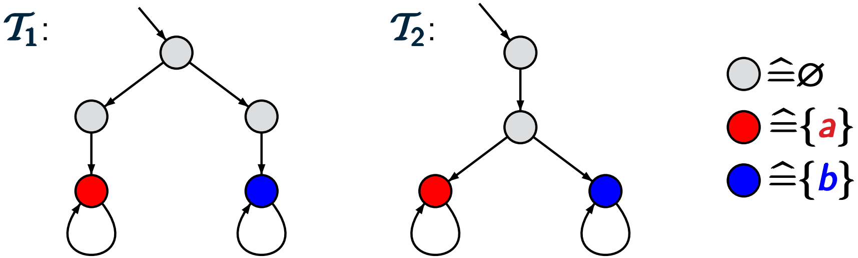



$$Traces(\mathcal{T}_1) = \{ \varnothing\,\varnothing\, a^\omega,\ \varnothing\,\varnothing\, b^\omega \} = Traces(\mathcal{T}_2)$$

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

# Trace equivalence



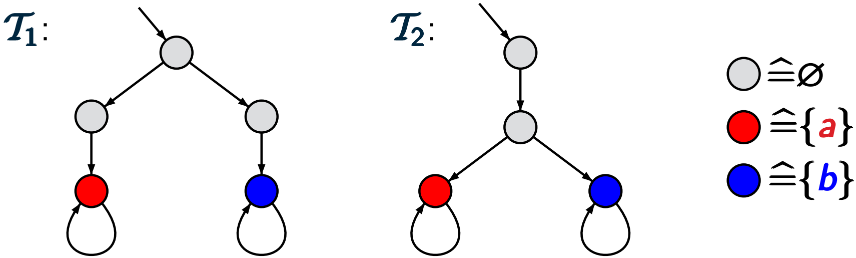



$$Traces(\mathcal{T}_1) = \{ \varnothing\varnothing a^\omega, \ \varnothing\varnothing b^\omega \} = Traces(\mathcal{T}_2)$$

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 




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

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

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

# 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 BSEQOR5.1-DEF-BIS-2TS

# 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

# 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

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 two transition systems

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

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

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

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

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

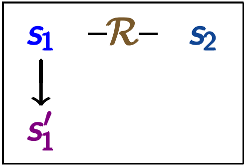
# Bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$



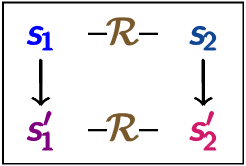
binary relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t. for all $(s_1, s_2) \in \mathcal{R}$:

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




can be completed to




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

$$\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \\ s_1' & & \end{array}$$

can be completed to

$$\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \downarrow \\ s_1' & -\mathcal{R}- & s_2' \end{array}$$

(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 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) } mutual stepwise simulation
(3)

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) } mutual stepwise simulation

and initial condition (I)

bisimulation equivalence $\sim$ for TS:

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) } 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 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) } mutual stepwise simulation

and initial condition (I)

bisimulation equivalence $\sim$ for TS:

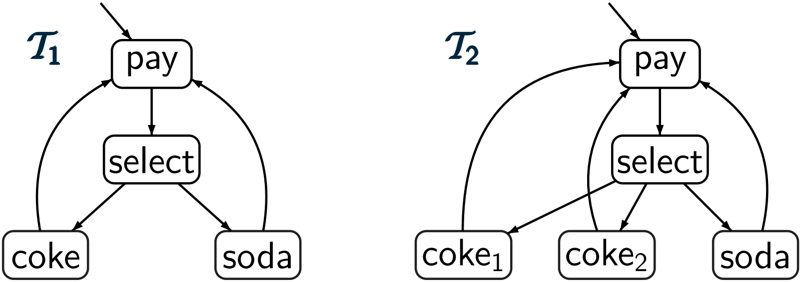
$\mathcal{T}_1 \sim \mathcal{T}_2$ iff there is a bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$

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

# Two beverage machines

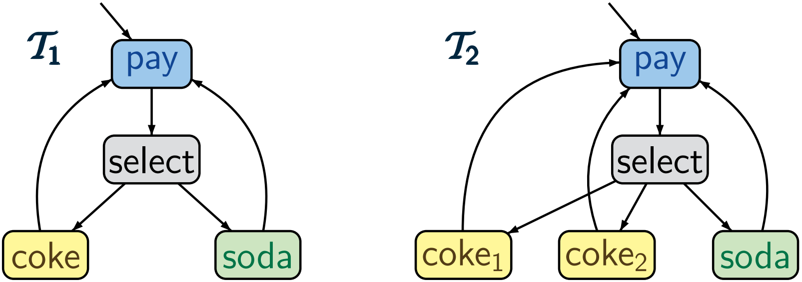





$AP = \{pay, coke, soda\}$

# Two beverage machines

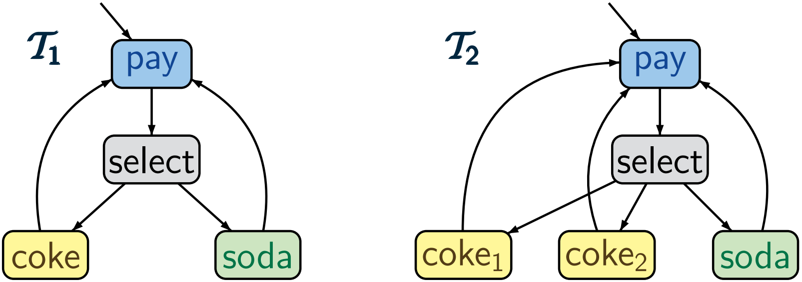





$$AP = \{pay, coke, soda\}$$

# Two beverage machines



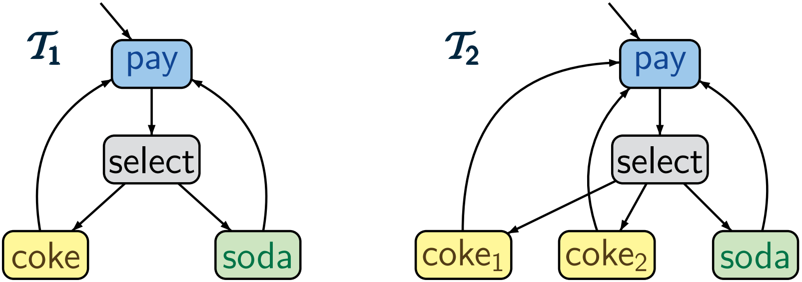



$AP = \{pay, coke, soda\}$

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

# Two beverage machines



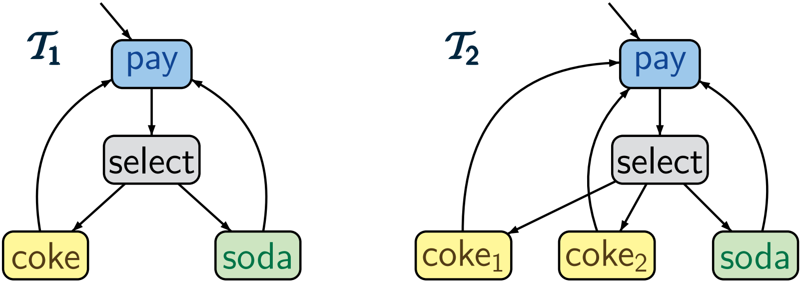



$AP = \{pay, coke, soda\}$

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

# Two beverage machines





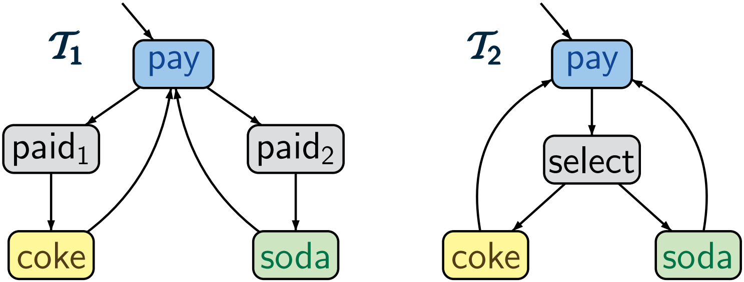

$AP = \{pay, coke, soda\}$

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

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

# Two beverage machines

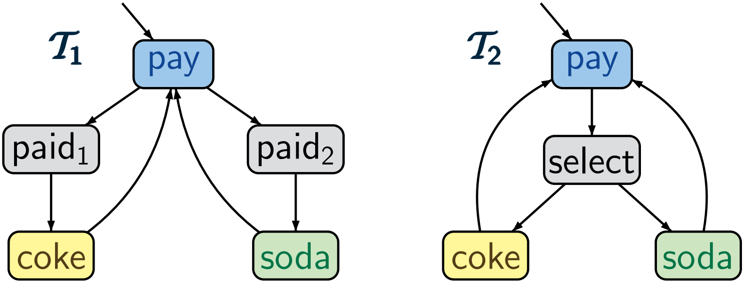





$$AP = \{pay, coke, soda\}$$

# Two beverage machines



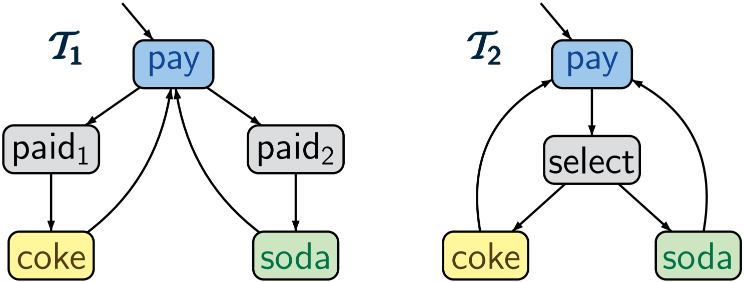



$AP = \{pay, coke, soda\}$

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

# 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

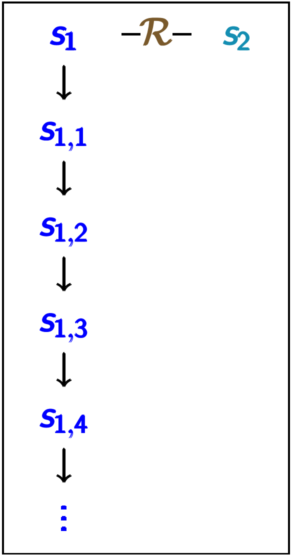
# Simulation condition of bisimulations



$$\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \\ s_1' & & \end{array}$$

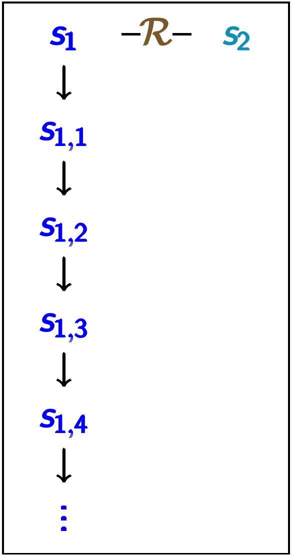
can be completed to

$$\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \downarrow \\ s_1' & -\mathcal{R}- & s_2' \end{array}$$



$s_1$ —$\mathcal{R}$— $s_2$
$s_{1,1}$
$s_{1,2}$
$s_{1,3}$
$s_{1,4}$
⋮

# Path lifting for bisimulation $\mathcal{R}$




can be
completed to

# 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

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

# 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

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

can be completed to

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

# 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 $\mathcal{T}$

$\sim$ 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 $(\mathcal{T}, \mathcal{T})$

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

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

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

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

↑

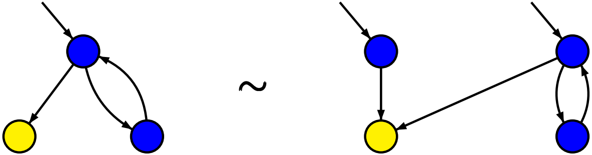
Let $\mathcal{R}_{1,2}$ be a bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$,
$\mathcal{R}_{2,3}$ be a bisimulation for $(\mathcal{T}_2, \mathcal{T}_3)$.

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

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

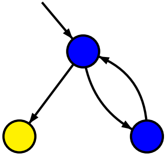
# Correct or wrong?

BSEQOR5.1-19

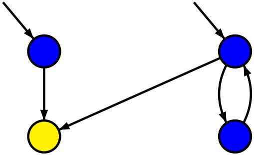


# Correct or wrong?





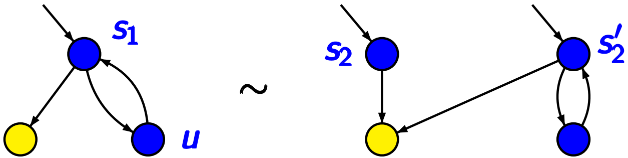
$\sim$



**wrong**

# Correct or wrong?



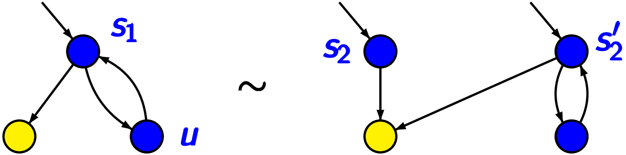



**wrong**

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

# Correct or wrong?

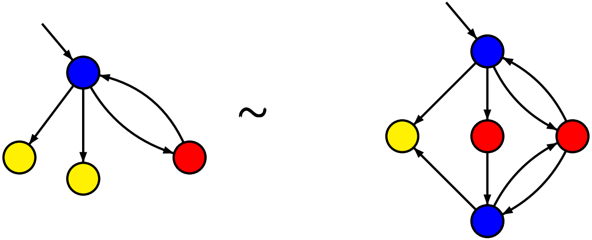





**wrong**

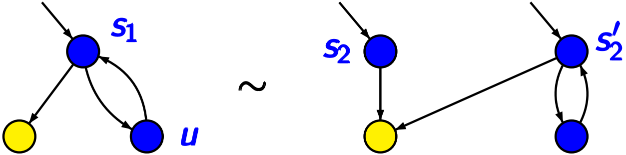
$s_1 \longrightarrow u$, but $s_2 \not\longrightarrow$ *blue* (thus $s_1 \not\sim s_2$)

---



# Correct or wrong?

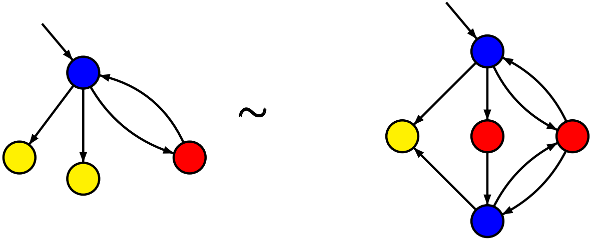





**wrong**

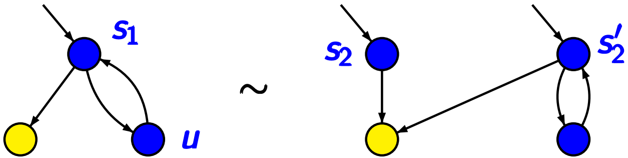
$s_1 \longrightarrow u$, but $s_2 \not\longrightarrow$ ***blue*** (thus $s_1 \not\sim s_2$)

---




**correct**

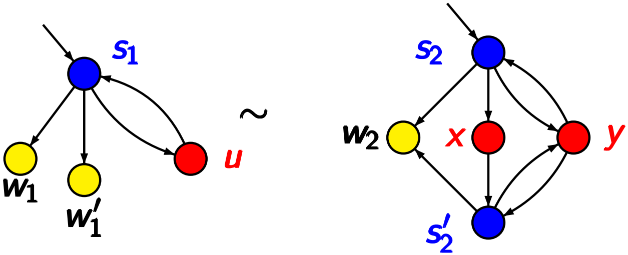
# Correct or wrong?




**wrong**

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

---



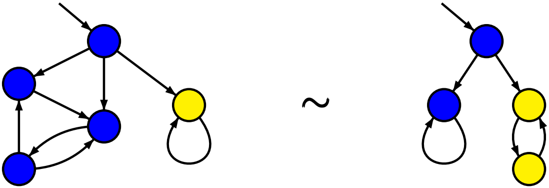

**correct**

bisimulation:

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

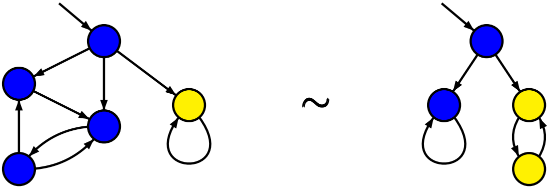
# Correct or wrong?

BSEQOR5.1-20



# Correct or wrong?

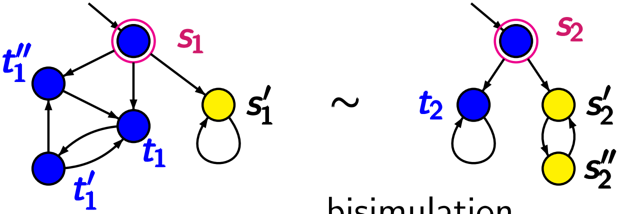




**correct**

# Correct or wrong?





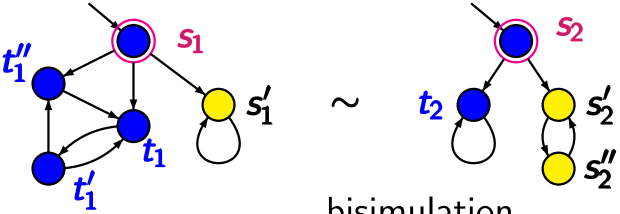

**correct**

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?



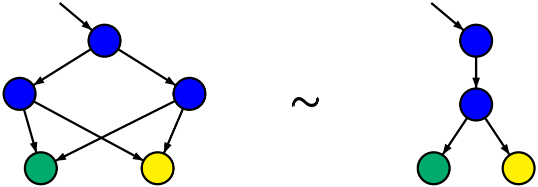



**correct**

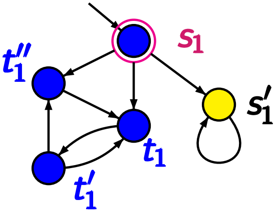
bisimulation

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

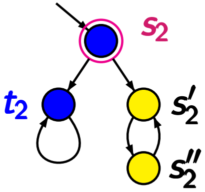
---



# Correct or wrong?




$\sim$

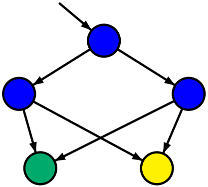



**correct**

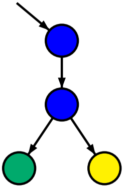
bisimulation

$$\{(s_1, s_2), (s'_1, s'_2), (s'_1, s''_2), (t_1, t_2), (t'_1, t_2), (t''_1, t_2)\}$$

---



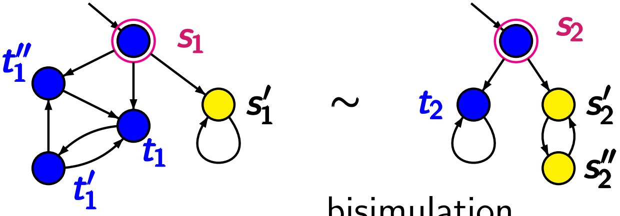
$\sim$



**correct**

# Correct or wrong?



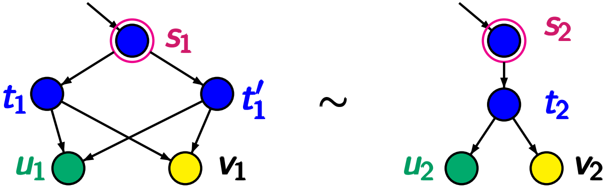



**correct**

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**

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



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

# 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 ...

# 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 ...

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

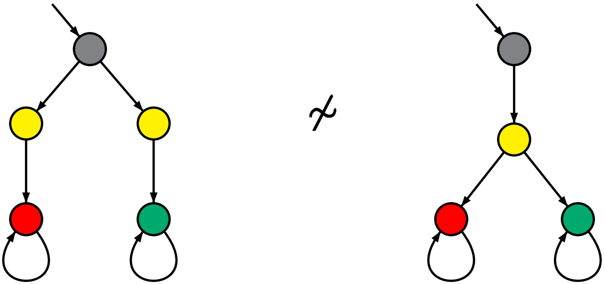
# Bisimulation vs. trace equivalence



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

*proof:* ... path fragment lifting ...

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



trace equivalent, but not bisimulation equivalent

# Bisimulation vs. trace equivalence



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

*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 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 ...

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

Trace equivalence is strictly coarser than bisimulation equivalence.

---

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

# Bisimulation equivalence ... 

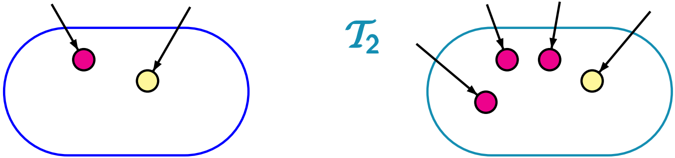
- as a relation that compares 2 transition systems

# Bisimulation equivalence ...

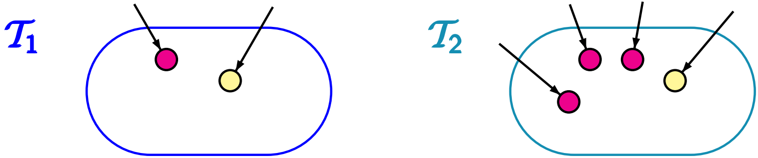


- as a relation that compares **2** transition systems

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



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

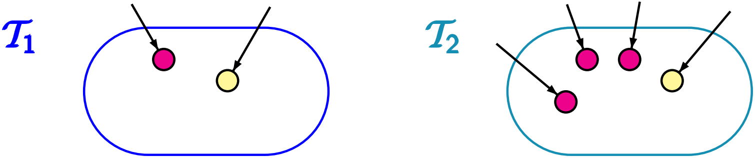



- as a relation on the states of **1** transition system

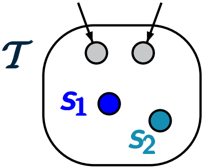
# Bisimulation equivalence ... 



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



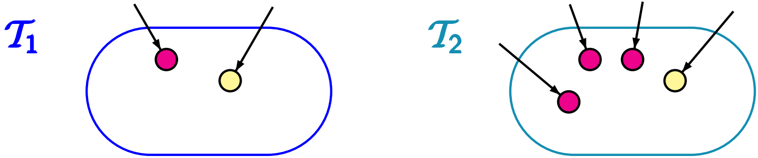

- as a relation on the states of **1** transition system



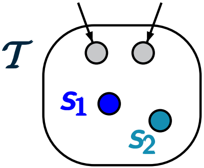
# Bisimulation equivalence ...



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



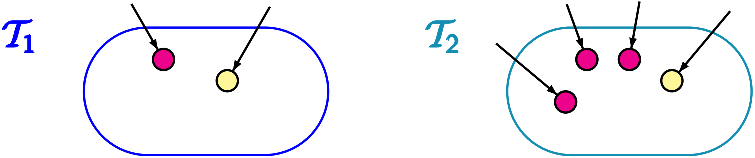

- as a relation on the states of **1** transition system



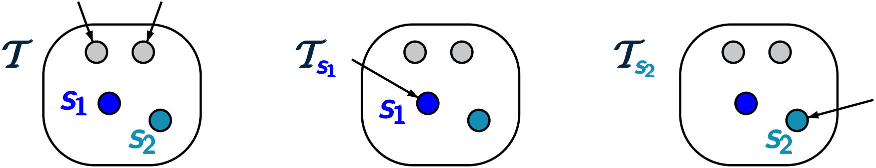

$s_1 \sim s_2$ iff $\mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$

# Bisimulation equivalence ...

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



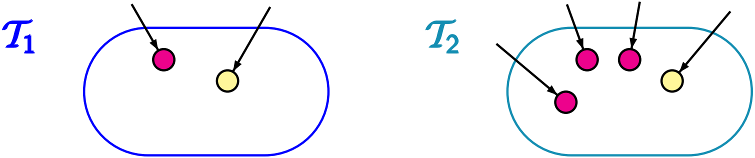

- as a relation on the states of **1** transition system



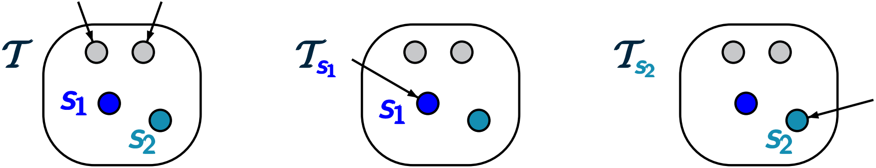

$s_1 \sim s_2$ iff $\mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$

# 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 $\mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$ 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 $AP$.

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 $\sim_{\mathcal{T}}$ 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. $(s_1', s_2') \in \mathcal{R}$

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

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

# Bisimulation equivalence $\sim_{\mathcal{T}}$ 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. $(s_1', s_2') \in \mathcal{R}$

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

# Bisimulation equivalence



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 $S$

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 $s_2$:

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

# 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 $\sim$

# 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 $\mathcal{T}$: |
| $s_1 \sim_{\mathcal{T}} s_2$ iff $\mathcal{T}_{s_1} \sim \mathcal{T}_{s_2}$ |

# 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 $\mathcal{T}$:

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

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

# 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

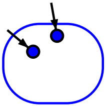
2. $\sim$ can be derived from $\sim_{\mathcal{T}}$

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

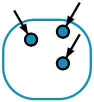


given two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$

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



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

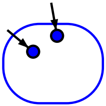


# Derivation of $\sim$ from $\sim_{\mathcal{T}}$

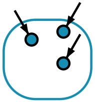


given two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$

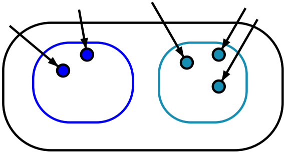
$\mathcal{T}_1$ with state space $S_1$



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



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

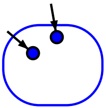


# Derivation of $\sim$ from $\sim_{\mathcal{T}}$

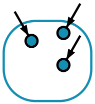


given two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$

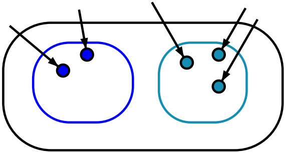
$\mathcal{T}_1$ with state space $S_1$



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



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



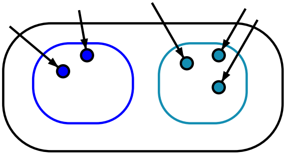
$\mathcal{T}_1 \sim \mathcal{T}_2$ iff $\forall$ initial states $s_1$ of $\mathcal{T}_1$
$\exists$ initial state $s_2$ of $\mathcal{T}_2$ s.t. $s_1 \sim_{\mathcal{T}} s_2$,

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

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

# 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 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: $S'_0 = \{[s]_{\sim_{\mathcal{T}}} : s \in S_0\}$

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: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$

# 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: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$

$\uparrow$

**well-defined**

by the labeling condition of bisimulations

# 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: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$
- 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: $S'_0 = \{[s]_{\sim_{\mathcal{T}}} : s \in S_0\}$
- labeling function: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$
- transition relation:

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

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: $S'_0 = \big\{ [s]_{\sim_{\mathcal{T}}} : s \in S_0 \big\}$
- labeling function: $L'([s]_{\sim_{\mathcal{T}}}) = L(s)$
- transition relation:

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

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

# Example: interleaving of $n$ printers



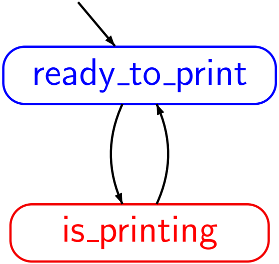
parallel system $\mathcal{T} = \underbrace{\textit{Printer} \parallel\!\mid \textit{Printer} \parallel\!\mid \ldots \parallel\!\mid \textit{Printer}}_{n \text{ printers}}$

# Example: interleaving of $n$ printers



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

transition system
for each printer

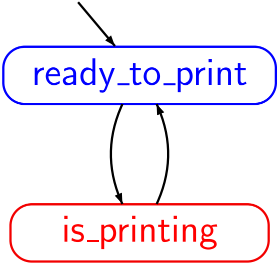


# Example: interleaving of *n* printers

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

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

transition system for each printer

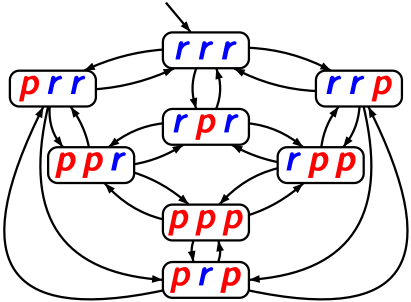


# Example: $n$=3 printers



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

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




***p***: is printing

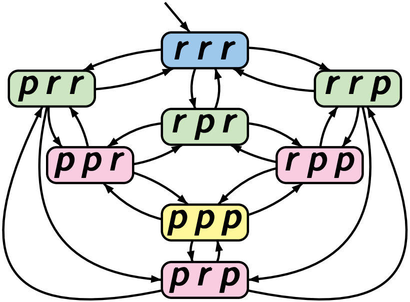
***r***: ready to print

# Example: $n$=3 printers



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

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




$p$: is printing

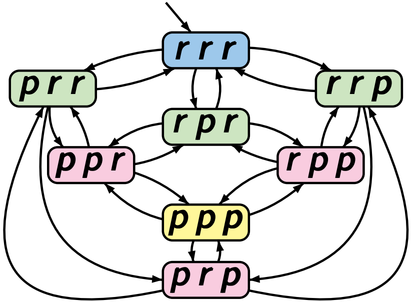
$r$: ready to print

# Example: $n$=3 printers



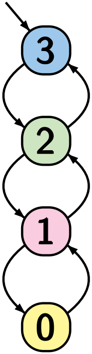
parallel system $\mathcal{T} = \underbrace{\textit{Printer} \,|||\, \textit{Printer} \,|||\, \ldots \,|||\, \textit{Printer}}_{n \text{ printers}}$

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




**p**: is printing

**r**: ready to print



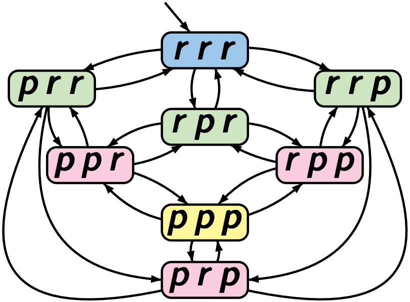

bisimulation quotient

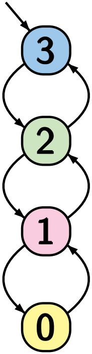
# Example: $n$=3 printers



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

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




$2^n$ states

$n+1$ states

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

# 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

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

**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**

**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

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

**CTL*** path formulas

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

**CTL**: sublogic of **CTL***

**CTL*** state formulas

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

**CTL*** path formulas

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

**CTL**: sublogic of **CTL***

- with path quantifiers ∃ and ∀

# Recall: CTL* and CTL



**CTL*** state formulas

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

**CTL*** path formulas

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

**CTL**: sublogic of **CTL***

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

**CTL*** state formulas

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

**CTL*** path formulas

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

**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

CTLEQ5.2-1

# CTL equivalence



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

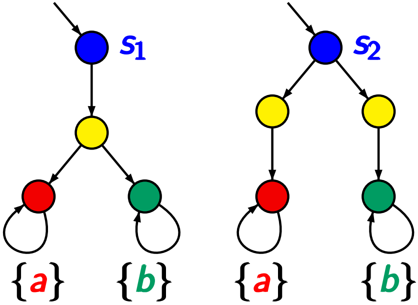
# CTL equivalence

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

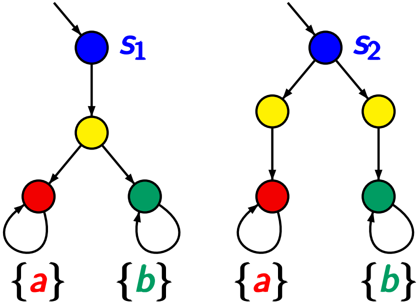
Let $s_1, s_2$ be states of a TS $\mathcal{T}$ without terminal states

> $s_1, s_2$ are **CTL** equivalent if for all **CTL** formulas $\Phi$:
>
> $s_1 \models \Phi$ iff $s_2 \models \Phi$



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




$s_1, s_2$ 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 $\Phi$:

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

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 $\Phi$:

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

$s_1, s_2$ are **CTL*** equivalent if for all **CTL*** formulas $\Phi$:

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

$s_1, s_2$ are **LTL** equivalent if for all **LTL** formulas $\varphi$:

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

# CTL/CTL* and bisimulation

# CTL/CTL* and bisimulation



bisimulation equivalence

= **CTL** equivalence

= **CTL*** equivalence

# CTL/CTL* and bisimulation



| | |
|---|---|
| | bisimulation equivalence |
| = | **CTL** equivalence |
| = | **CTL*** equivalence |

⟵ for finite TS

| | | |
|---|---|---|
| | bisimulation equivalence | |
| = | **CTL** equivalence | ⟵ 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:

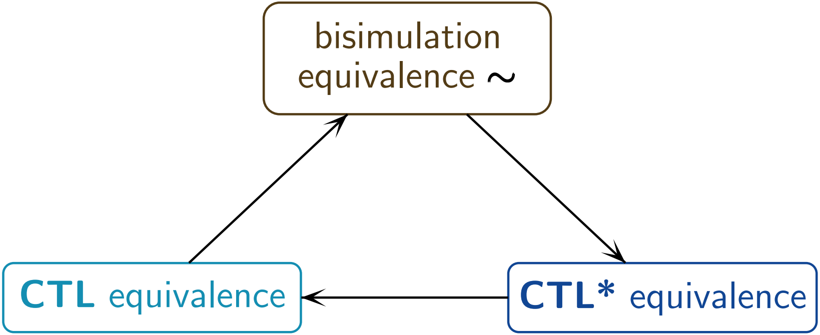
$$s_1 \sim_{\mathcal{T}} s_2$$

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

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

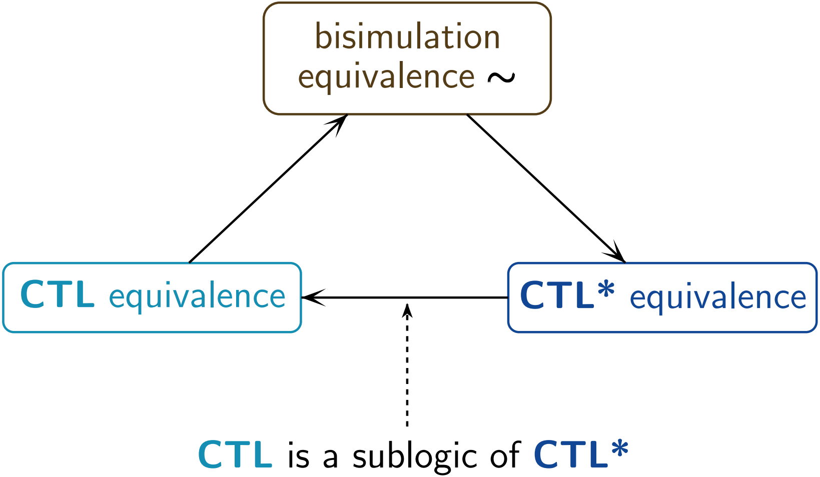
# CTL/CTL* and bisimulation





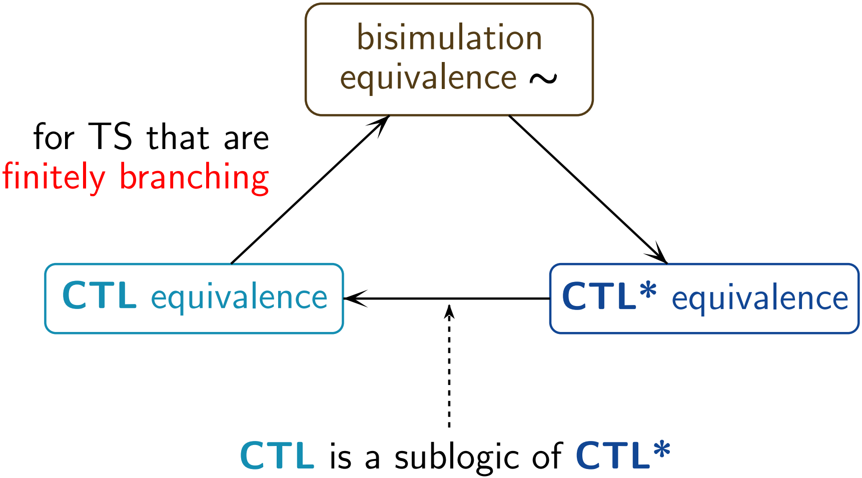
# CTL/CTL* and bisimulation





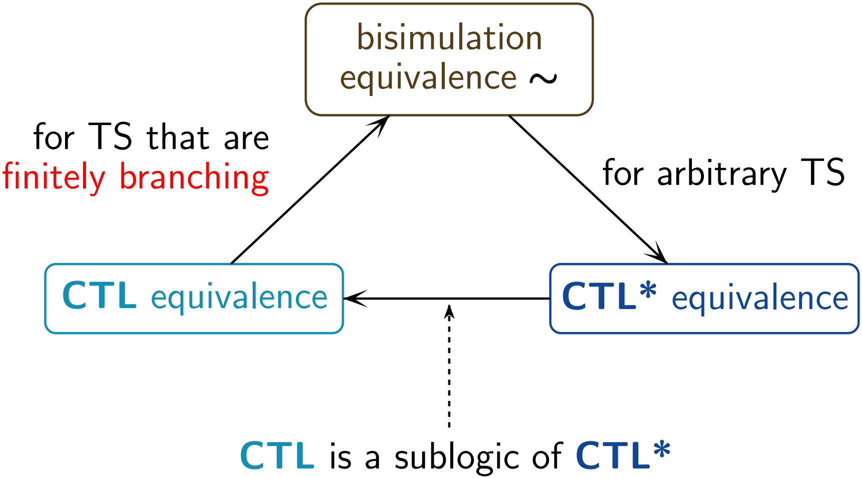
# CTL/CTL* and bisimulation





# CTL/CTL* and bisimulation





# Bisimulation equivalence ⇒ CTL* equivalence



For arbitrary (possibly infinite) transition systems without terminal states:

# Bisimulation equivalence ⇒ CTL* equivalence



For arbitrary (possibly infinite) transition systems without terminal states:

> If $s_1$, $s_2$ are states with $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL*** formulas $\Phi$:
>
> $$s_1 \models \Phi \quad \text{iff} \quad s_2 \models \Phi$$

show by structural induction on **CTL*** formulas:

(a) if $s_1$, $s_2$ are states with $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL*** state formulas $\Phi$:

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

(b) if $\pi_1$, $\pi_2$ are paths with $\pi_1 \sim_{\mathcal{T}} \pi_2$ then for all **CTL*** path formulas $\varphi$:

$$\pi_1 \models \varphi \quad \text{iff} \quad \pi_2 \models \varphi$$

# Bisimulation equivalence ⇒ CTL* equivalence



show by structural induction on **CTL*** formulas:

(a) if $s_1$, $s_2$ are states with $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL*** state formulas $\Phi$:

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

(b) if $\pi_1$, $\pi_2$ are paths with $\pi_1 \sim_{\mathcal{T}} \pi_2$ then for all **CTL*** path formulas $\varphi$:

$$\pi_1 \models \varphi \quad \text{iff} \quad \pi_2 \models \varphi$$

$\pi_1 \sim_{\mathcal{T}} \pi_2 \quad \overset{\text{def}}{\iff} \quad$ $\pi_1$ and $\pi_2$ are statewise bisimulation equivalent

# Bisimulation equivalence $\Rightarrow$ CTL* equivalence



statewise bisimulation equivalent paths:

$$
\begin{array}{ccc}
s_1 & \sim_{\mathcal{T}} & s_2 \\
\downarrow & & \downarrow \\
s_{11} & \sim_{\mathcal{T}} & s_{12} \\
\downarrow & & \downarrow \\
s_{21} & \sim_{\mathcal{T}} & s_{22} \\
\downarrow & & \downarrow \\
s_{31} & \sim_{\mathcal{T}} & s_{32} \\
\downarrow & & \downarrow \\
\ldots & & \ldots \\
\uparrow & & \uparrow \\
\text{path } \pi_1 & & \text{path } \pi_2
\end{array}
$$

# Bisimulation equivalence $\Rightarrow$ CTL* equivalence



For all CTL* state formulas $\Phi$ and path formulas $\varphi$:

(a) if $s_1 \sim_{\mathcal{T}} s_2$ then: $s_1 \models \Phi$ iff $s_2 \models \Phi$

(b) if $\pi_1 \sim_{\mathcal{T}} \pi_2$ then: $\pi_1 \models \varphi$ iff $\pi_2 \models \varphi$

# Bisimulation equivalence $\Rightarrow$ CTL* equivalence 

For all CTL* state formulas $\Phi$ and path formulas $\varphi$:

(a) if $s_1 \sim_{\mathcal{T}} s_2$ then: $s_1 \models \Phi$ iff $s_2 \models \Phi$

(b) if $\pi_1 \sim_{\mathcal{T}} \pi_2$ then: $\pi_1 \models \varphi$ iff $\pi_2 \models \varphi$

Proof by structural induction

# Bisimulation equivalence $\Rightarrow$ CTL* equivalence 

For all CTL* state formulas $\Phi$ and path formulas $\varphi$:

(a) if $s_1 \sim_{\mathcal{T}} s_2$ then: $s_1 \models \Phi$ iff $s_2 \models \Phi$

(b) if $\pi_1 \sim_{\mathcal{T}} \pi_2$ then: $\pi_1 \models \varphi$ iff $\pi_2 \models \varphi$

Proof by structural induction

base of induction:

(a) $\Phi = \mathit{true}$ or $\Phi = a \in AP$

(b) $\varphi = \Phi$ for some state formula $\Phi$
s.t. statement (a) holds for $\Phi$

For all CTL* state formulas $\Phi$ and path formulas $\varphi$:

(a) if $s_1 \sim_{\mathcal{T}} s_2$ then: $s_1 \models \Phi$ iff $s_2 \models \Phi$

(b) if $\pi_1 \sim_{\mathcal{T}} \pi_2$ then: $\pi_1 \models \varphi$ iff $\pi_2 \models \varphi$

Proof by structural induction

step of induction:

(a) consider $\Phi = \Phi_1 \wedge \Phi_2$, $\neg\Psi$ or $\exists\varphi$ s.t.

- (a) holds for $\Phi_1, \Phi_2, \Psi$
- (b) holds for $\varphi$

(b) consider $\varphi = \varphi_1 \wedge \varphi_2$, $\neg\varphi'$, $\bigcirc\varphi'$, $\varphi_1 \,\mathbf{U}\, \varphi_2$ s.t.

- (b) holds for $\varphi_1, \varphi_2, \varphi'$

# Path lifting for $\sim_{\mathcal{T}}$



$$
\begin{array}{ccc}
s_1 & \sim_{\mathcal{T}} & s_2 \\
\downarrow & & \\
s_{11} & & \\
\downarrow & & \\
s_{21} & & \\
\downarrow & & \\
s_{31} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

can be completed to

$$
\begin{array}{ccc}
s_1 & \sim_{\mathcal{T}} & s_2 \\
\downarrow & & \downarrow \\
s_{11} & \sim_{\mathcal{T}} & s_{12} \\
\downarrow & & \downarrow \\
s_{21} & \sim_{\mathcal{T}} & s_{22} \\
\downarrow & & \downarrow \\
s_{31} & \sim_{\mathcal{T}} & s_{32} \\
\downarrow & & \downarrow \\
\vdots & & \vdots
\end{array}
$$

# Path lifting for $\sim_{\mathcal{T}}$



$$
\begin{array}{ccc}
s_1 & \sim_{\mathcal{T}} & s_2 \\
\downarrow & & \\
s_{11} & & \\
\downarrow & & \\
s_{21} & & \\
\downarrow & & \\
s_{31} & & \\
\downarrow & & \\
\vdots & &
\end{array}
$$

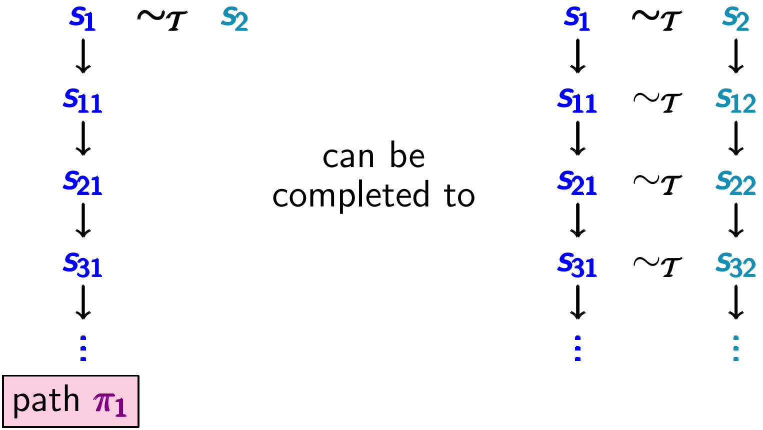
can be
completed to

$$
\begin{array}{ccc}
s_1 & \sim_{\mathcal{T}} & s_2 \\
\downarrow & & \downarrow \\
s_{11} & \sim_{\mathcal{T}} & s_{12} \\
\downarrow & & \downarrow \\
s_{21} & \sim_{\mathcal{T}} & s_{22} \\
\downarrow & & \downarrow \\
s_{31} & \sim_{\mathcal{T}} & s_{32} \\
\downarrow & & \downarrow \\
\vdots & & \vdots
\end{array}
$$

If $s_1 \sim_{\mathcal{T}} s_2$ then for all $\pi_1 \in Paths(s_1)$
there exists $\pi_2 \in Paths(s_2)$ with $\pi_1 \sim_{\mathcal{T}} \pi_2$

# Path lifting for $\sim_{\mathcal{T}}$

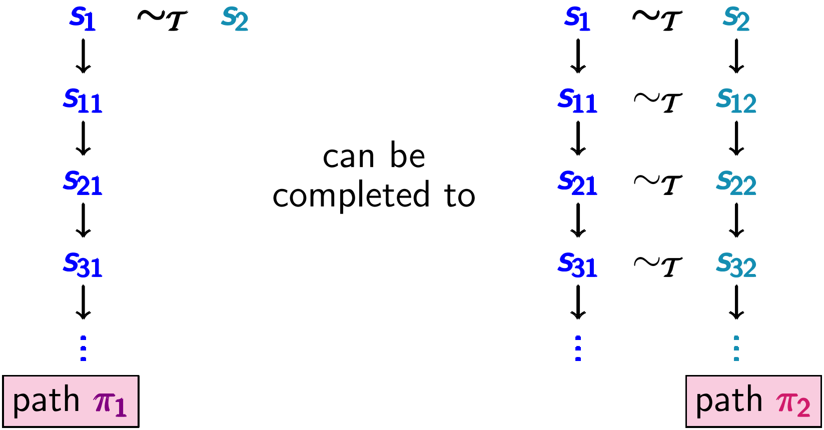





If $s_1 \sim_{\mathcal{T}} s_2$ then for all $\pi_1 \in Paths(s_1)$ there exists $\pi_2 \in Paths(s_2)$ with $\pi_1 \sim_{\mathcal{T}} \pi_2$

# Path lifting for $\sim_{\mathcal{T}}$






If $s_1 \sim_{\mathcal{T}} s_2$ then for all $\pi_1 \in \mathit{Paths}(s_1)$
there exists $\pi_2 \in \mathit{Paths}(s_2)$ with $\pi_1 \sim_{\mathcal{T}} \pi_2$

# Correct or wrong?



If $s_1, s_2$ are not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

If $s_1, s_2$ are not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

**correct**.

If $s_1, s_2$ are not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

**correct**.

If $s_1$, $s_2$ not **CTL** equivalent then
there exists a **CTL** formula $\Phi$ with

$$s_1 \models \Phi \wedge s_2 \not\models \Phi$$

$$\text{or} \quad s_1 \not\models \Phi \wedge s_2 \models \Phi$$

If $s_1, s_2$ are <u>not</u> **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

**correct**.

If $s_1$, $s_2$ not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with

$$s_1 \models \Phi \wedge s_2 \not\models \Phi$$

$$\text{or} \quad s_1 \not\models \Phi \wedge s_2 \models \Phi \quad \Longrightarrow \quad s_1 \models \neg\Phi \wedge s_2 \not\models \neg\Phi$$

# Correct or wrong?

If $s_1, s_2$ are not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

**correct**.

If $s_1$, $s_2$ are not **LTL** equivalent then there exists a **LTL** formula $\varphi$ with $s_1 \models \varphi$ and $s_2 \not\models \varphi$

# Correct or wrong?

If $s_1, s_2$ are not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

**correct**.

If $s_1$, $s_2$ are not **LTL** equivalent then there exists a **LTL** formula $\varphi$ with $s_1 \models \varphi$ and $s_2 \not\models \varphi$

**wrong.**

# Correct or wrong?

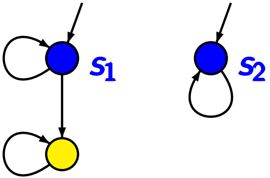


If $s_1, s_2$ are not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

**correct**.

If $s_1$, $s_2$ are not **LTL** equivalent then there exists a **LTL** formula $\varphi$ with $s_1 \models \varphi$ and $s_2 \not\models \varphi$

**wrong.**



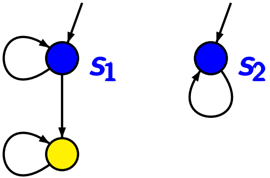
If $s_1, s_2$ are not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

**correct**.

If $s_1$, $s_2$ are not **LTL** equivalent then there exists a **LTL** formula $\varphi$ with $s_1 \models \varphi$ and $s_2 \not\models \varphi$

**wrong.**

$Traces(s_2) \subset Traces(s_1)$



If $s_1, s_2$ are not **CTL** equivalent then there exists a **CTL** formula $\Phi$ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

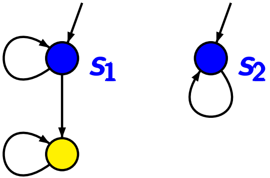
**correct**.

If $s_1$, $s_2$ are not **LTL** equivalent then there exists a **LTL** formula $\varphi$ with $s_1 \models \varphi$ and $s_2 \not\models \varphi$

**wrong.**

$Traces(s_2) \subset Traces(s_1)$

hence: $s_1 \models \varphi$ implies $s_2 \models \varphi$



# CTL equivalence $\Longrightarrow$ bisimulation equivalence 

# CTL equivalence $\Longrightarrow$ bisimulation equivalence



If $\mathcal{T}$ is a finite TS then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

# CTL equivalence $\Longrightarrow$ bisimulation equivalence

If $\mathcal{T}$ is a finite TS then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

# CTL equivalence $\Longrightarrow$ bisimulation equivalence



If $\mathcal{T}$ is a finite TS then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

*Proof:* show that

$$\mathcal{R} \stackrel{\text{def}}{=} \{ (s_1, s_2) : s_1, s_2 \text{ satisfy the same } \mathbf{CTL} \text{ formulas} \}$$

is a bisimulation

If $\mathcal{T}$ is a finite TS then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

*Proof:* show that

$$\mathcal{R} \stackrel{\text{def}}{=} \{ (s_1, s_2) : s_1,\ s_2 \text{ satisfy the same } \textbf{CTL} \text{ formulas} \}$$

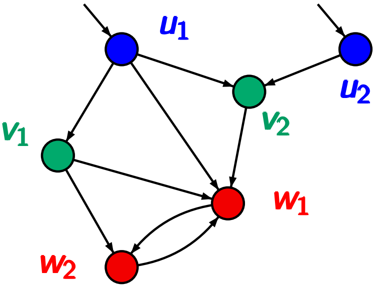
is a bisimulation, i.e., for all $(s_1, s_2) \in \mathcal{R}$:

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

(2) if $s_1 \to t_1$ then there exists a transition $s_2 \to t_2$ s.t. $(t_1, t_2) \in \mathcal{R}$

# Example: CTL master formulas





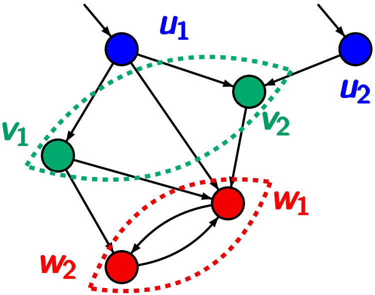


$\widehat{=} \{a\}$


$\widehat{=} \{b\}$


$\widehat{=} \varnothing$

# Example: CTL master formulas



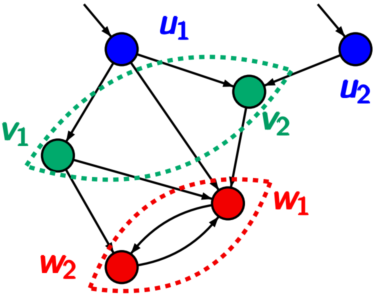

bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \dots\}$

$\widehat{=} \{a\}$

$\widehat{=} \{b\}$

$\widehat{=} \varnothing$

# Example: CTL master formulas




bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

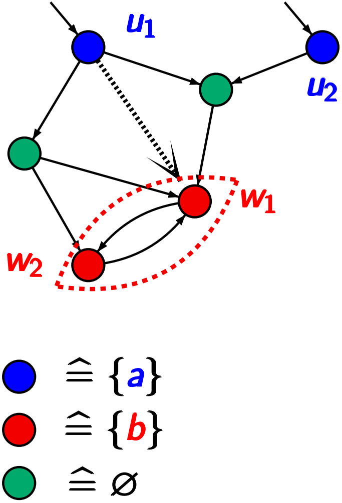
but $u_1 \not\sim_{\mathcal{T}} u_2$

 $\widehat{=} \{a\}$

 $\widehat{=} \{b\}$

 $\widehat{=} \varnothing$

# Example: CTL master formulas




bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

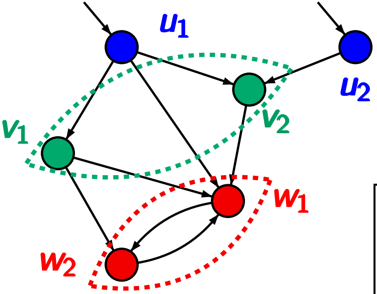
but $u_1 \not\sim_{\mathcal{T}} u_2$

as $u_1 \rightarrow \{w_1, w_2\}$

$u_2 \not\rightarrow \{w_1, w_2\}$

# Example: CTL master formulas







bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

**CTL** master formulas:

$w_1, w_2 \models$ ?

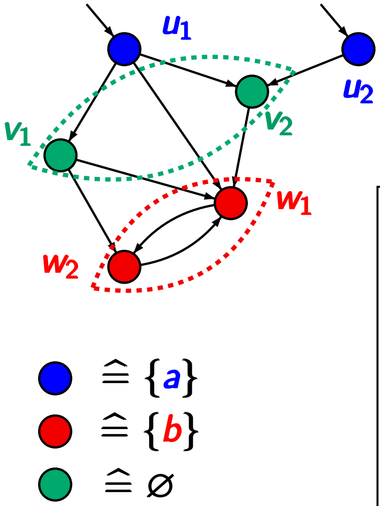
$v_1, v_2 \models$ ?

$u_1 \models$ ?

$u_2 \models$ ?

# Example: CTL master formulas






bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

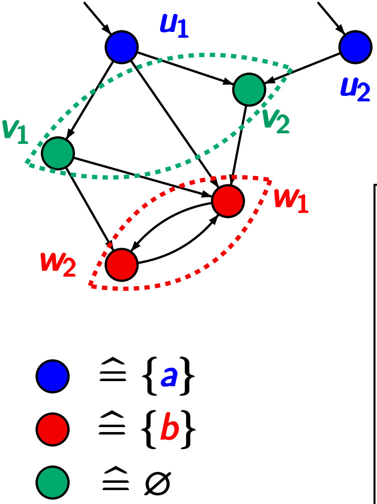
**CTL** master formulas:

$w_1, w_2 \models b$

$v_1, v_2 \models ?$

$u_1 \models ?$

$u_2 \models ?$




bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

**CTL** master formulas:

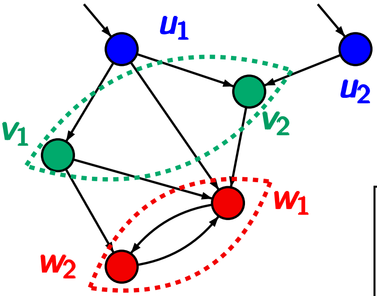
$w_1, w_2 \models b$

$v_1, v_2 \models \neg a \wedge \neg b$

$u_1 \models$ ?

$u_2 \models$ ?

# Example: CTL master formulas




bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

● $\widehat{=} \{a\}$
● $\widehat{=} \{b\}$
● $\widehat{=} \varnothing$

**CTL** master formulas:

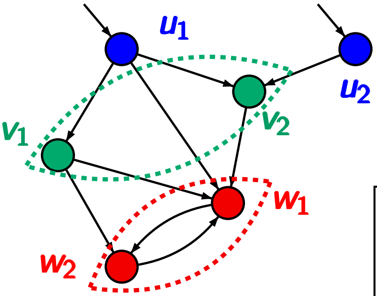
$w_1, w_2 \models b$

$v_1, v_2 \models \neg a \wedge \neg b$

$u_1 \models (\exists \bigcirc b) \wedge a$

$u_2 \models$ ?

# Example: CTL master formulas




bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

$\hat{=} \{a\}$

$\hat{=} \{b\}$

$\hat{=} \varnothing$

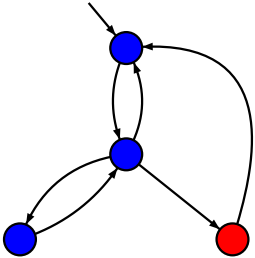
**CTL** master formulas:

$w_1, w_2 \models b$

$v_1, v_2 \models \neg a \wedge \neg b$

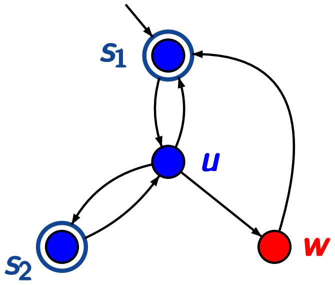
$u_1 \models (\exists \bigcirc b) \wedge a$

$u_2 \models (\neg \exists \bigcirc b) \wedge a$



$AP = \{blue, red\}$

# ...master formulas for $\sim_{\mathcal{T}}$-classes?

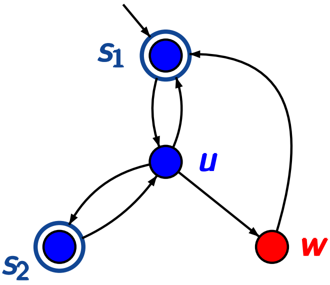



$AP = \{blue, red\}$

$s_1 \sim_{\mathcal{T}} s_2 \not\sim_{\mathcal{T}} u$

# ...master formulas for $\sim_{\mathcal{T}}$-classes?






$AP = \{blue, red\}$

$s_1 \sim_{\mathcal{T}} s_2 \not\sim_{\mathcal{T}} u$

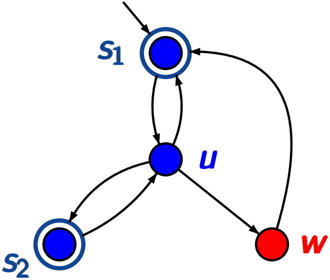
$$\Phi_w = ?$$

$$\Phi_C = ? \qquad \text{where } C = \{s_1, s_2\}$$

$$\Phi_u = ?$$

# ...master formulas for $\sim_{\mathcal{T}}$-classes?






$AP = \{blue, red\}$

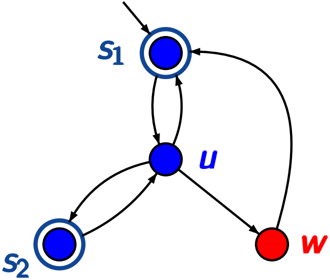
$s_1 \sim_{\mathcal{T}} s_2 \not\sim_{\mathcal{T}} u$

$\Phi_w = red$

$\Phi_C = ?$ where $C = \{s_1, s_2\}$

$\Phi_u = ?$

# ...master formulas for $\sim_{\mathcal{T}}$-classes? 



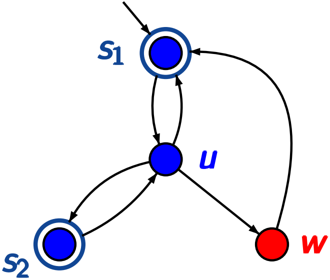

$AP = \{blue, red\}$

$s_1 \sim_{\mathcal{T}} s_2 \not\sim_{\mathcal{T}} u$

$$\Phi_w = red$$

$$\Phi_C = blue \wedge \forall \bigcirc blue \quad \text{where } C = \{s_1, s_2\}$$

$$\Phi_u = \ ?$$




$AP = \{blue, red\}$

$s_1 \sim_{\mathcal{T}} s_2 \not\sim_{\mathcal{T}} u$

$$\Phi_w = red$$

$$\Phi_C = blue \wedge \forall \bigcirc blue \quad \text{where } C = \{s_1, s_2\}$$

$$\Phi_u = \exists \bigcirc red$$

# CTL equivalence $\Longrightarrow$ bisimulation equivalence



If $\mathcal{T}$ is a finite TS then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

# CTL equivalence $\Longrightarrow$ bisimulation equivalence



If $\mathcal{T}$ is a finite TS then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

- wrong for infinite TS

# CTL equivalence $\Longrightarrow$ bisimulation equivalence 

If $\mathcal{T}$ is a finite TS then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

- wrong for infinite TS
- but also holds for finitely branching TS

If $\mathcal{T}$ is a finite TS then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

- wrong for infinite TS
- but also holds for finitely branching TS

$\uparrow$

possibly infinite-state TS such that

- $*$ the number of initial states is finite
- $*$ for each state the number of successors is finite

# CTL equivalence $\Longrightarrow$ bisimulation equivalence 

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

# CTL equivalence $\Longrightarrow$ bisimulation equivalence 

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

* $S_0$ is finite
* $Post(s)$ is finite for all $s \in S$

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

↑

* $S_0$ is finite
* $Post(s)$ is finite for all $s \in S$

Then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

# CTL equivalence $\Longrightarrow$ bisimulation equivalence 

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

- $*$ $S_0$ is finite
- $*$ $Post(s)$ is finite for all $s \in S$

Then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

*Proof:* as for finite TS.

# CTL equivalence $\Longrightarrow$ bisimulation equivalence 

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

- $S_0$ is finite
- $Post(s)$ is finite for all $s \in S$

Then, for all states $s_1$, $s_2$ in $\mathcal{T}$:

> if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

*Proof:* as for finite TS. Amounts showing that

$$\mathcal{R} \stackrel{\text{def}}{=} \{ (s_1, s_2) : s_1, s_2 \text{ satisfy the same } \textbf{CTL} \text{ formulas} \}$$

is a bisimulation.

# CTL equivalence $\Longrightarrow$ bisimulation equivalence 

If $\mathcal{T}$ is a finitely branching TS then for all states $s_1$, $s_2$:

if $s_1$, $s_2$ are **CTL** equivalent then $s_1 \sim_{\mathcal{T}} s_2$

*Proof:* show that

$$\mathcal{R} \overset{\text{def}}{=} \{ (s_1, s_2) : s_1,\ s_2 \text{ satisfy the same } \mathbf{CTL} \text{ formulas} \}$$

is a bisimulation, i.e., for $(s_1, s_2) \in \mathcal{R}$:

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

(2) if $s_1 \to t_1$ then there exists a transition $s_2 \to t_2$ s.t. $(t_1, t_2) \in \mathcal{R}$

# Summary: CTL/CTL* and bisimulation

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

# Summary: CTL/CTL* and bisimulation



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

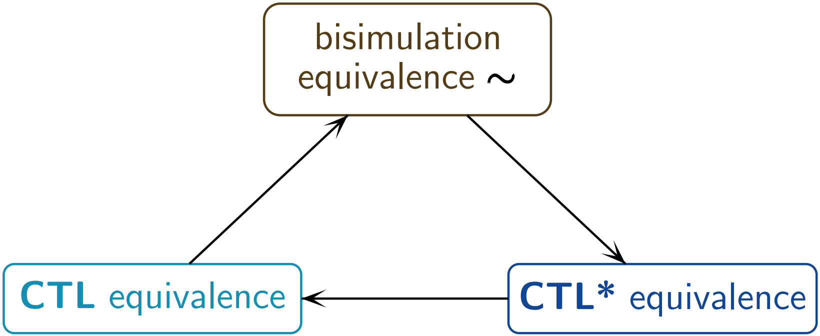
$$s_1 \sim_{\mathcal{T}} s_2$$

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

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

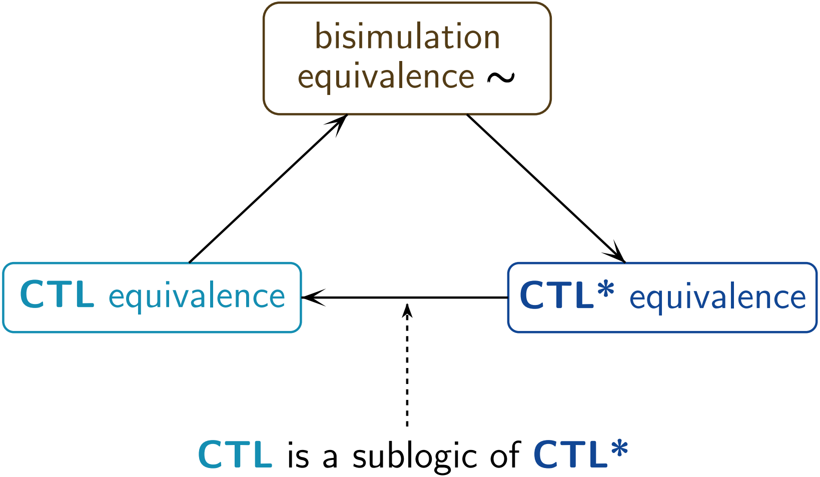
# Summary: CTL/CTL* and bisimulation





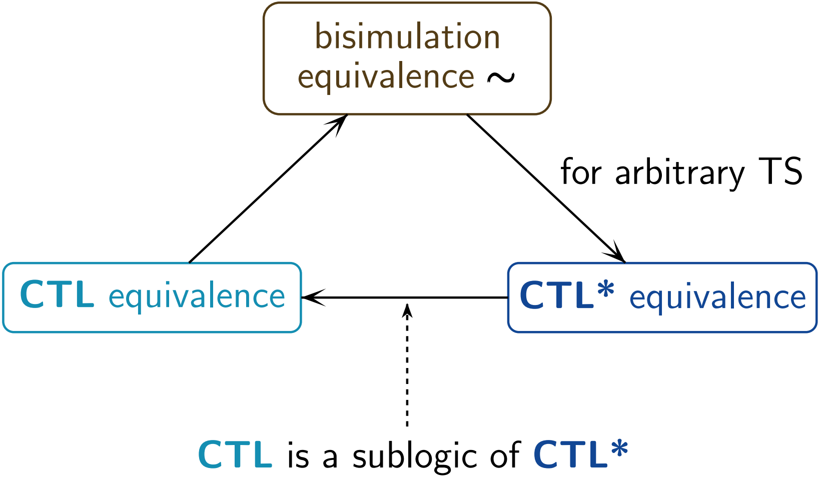
# Summary: CTL/CTL* and bisimulation



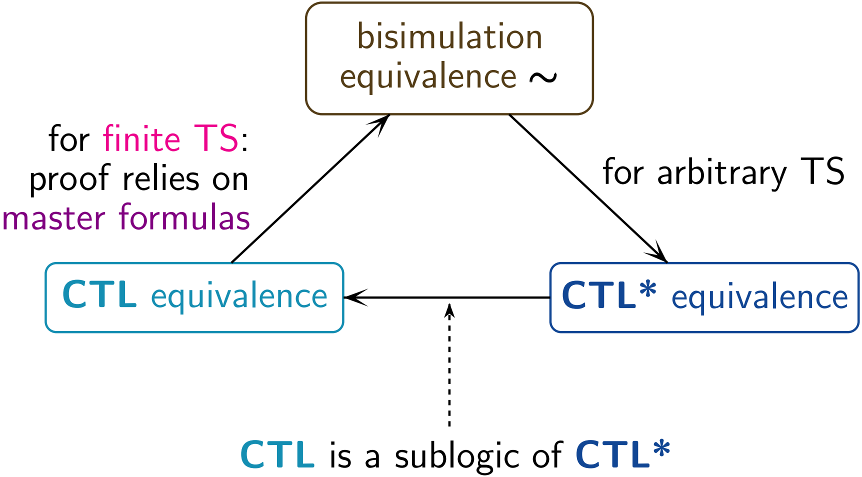


# Summary: CTL/CTL* and bisimulation

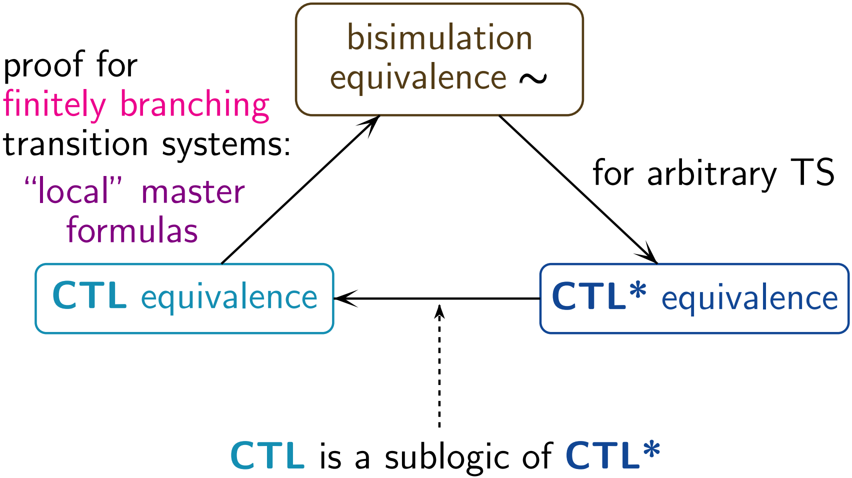




# Summary: CTL/CTL* and bisimulation



# Summary: CTL/CTL* and bisimulation



# CTL/CTL* and bisimulation for TS

# CTL/CTL* and bisimulation for TS



*so far:* we considered

- **CTL**/**CTL*** equivalence
- bisimulation equivalence $\sim_{\mathcal{T}}$

for the states of a single transition system $\mathcal{T}$

# 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?



Does the following statements hold for finite TS without terminal states **?**

# Correct or wrong?



**CTL** equivalence is finer than **LTL** equivalence

# Correct or wrong?



**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***

# Correct or wrong?

CTLEQ5.2-9

**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 |

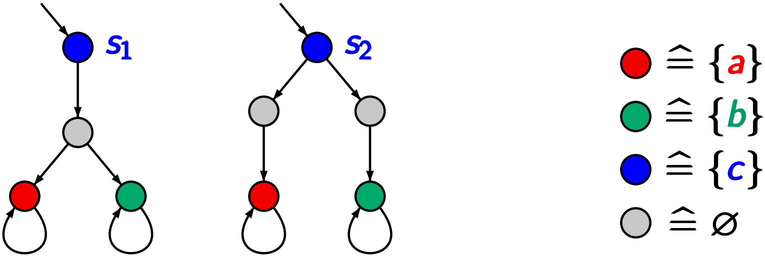
**wrong.**

**CTL** equivalence is finer than **LTL** equivalence

**correct.**

**LTL** equivalence is finer than **CTL** equivalence

**wrong.**

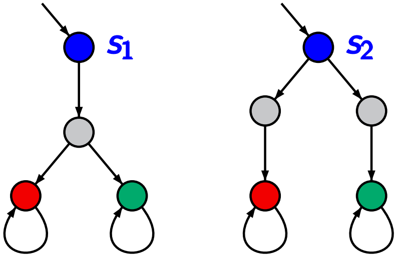


**CTL** equivalence is finer than **LTL** equivalence

**correct.**

**LTL** equivalence is finer than **CTL** equivalence

**wrong.**



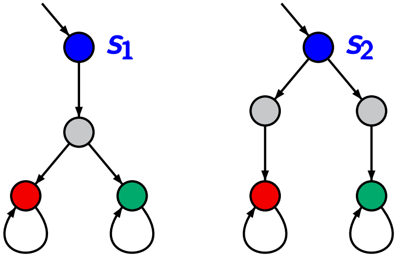

$s_1$, $s_2$ are trace equivalent

| |
|---|
| **CTL** equivalence is finer than **LTL** equivalence |

**correct.**

| |
|---|
| **LTL** equivalence is finer than **CTL** equivalence |

**wrong.**



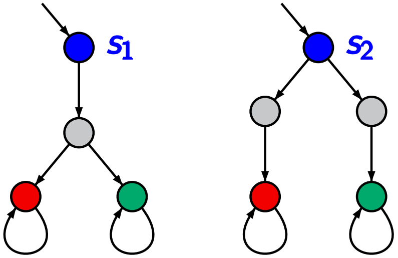

$s_1$, $s_2$ are trace equivalent and **LTL** equivalent

**CTL** equivalence is finer than **LTL** equivalence

**correct.**

**LTL** equivalence is finer than **CTL** equivalence

**wrong.**



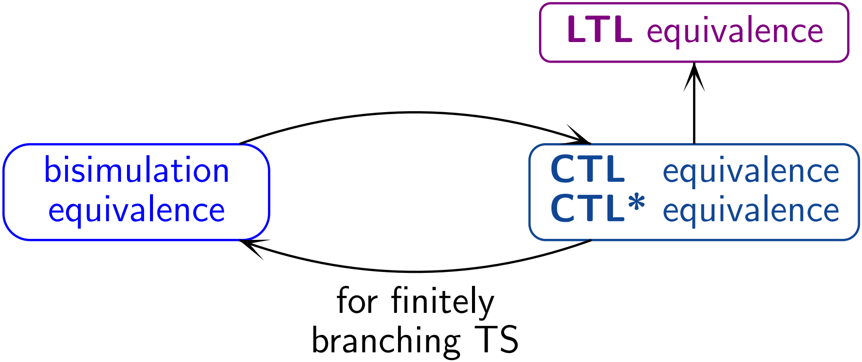

$s_1$, $s_2$ are trace equivalent

and **LTL** equivalent

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

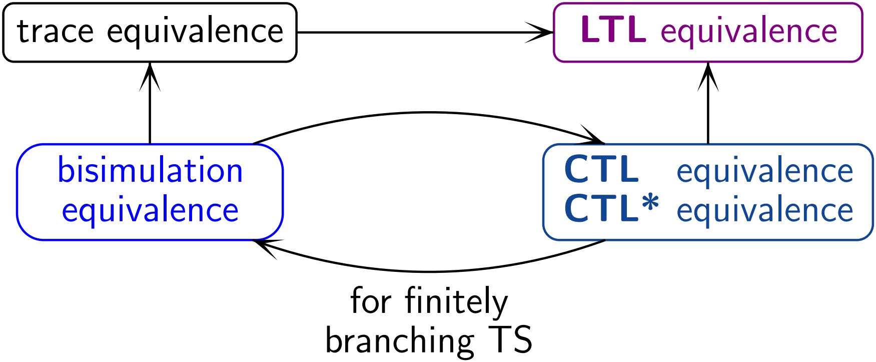
# Summary: equivalences





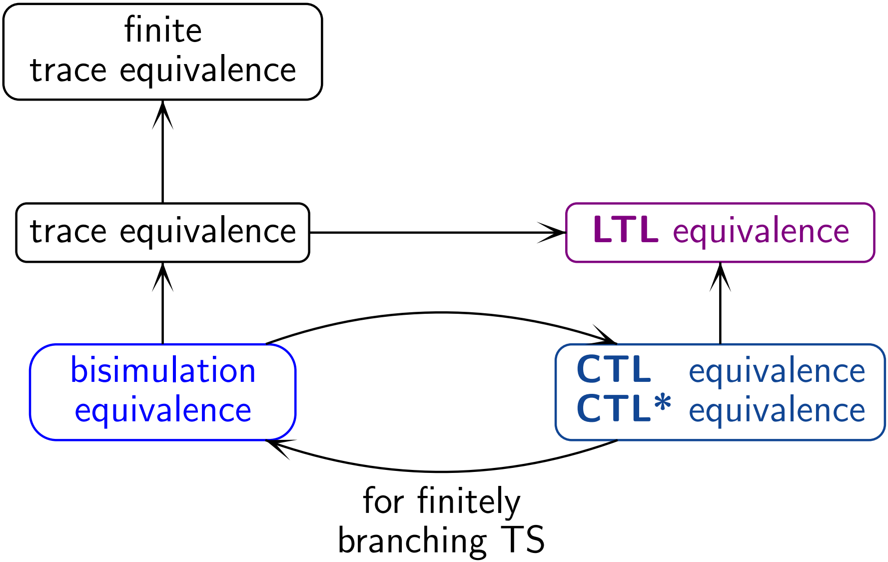
# Summary: equivalences





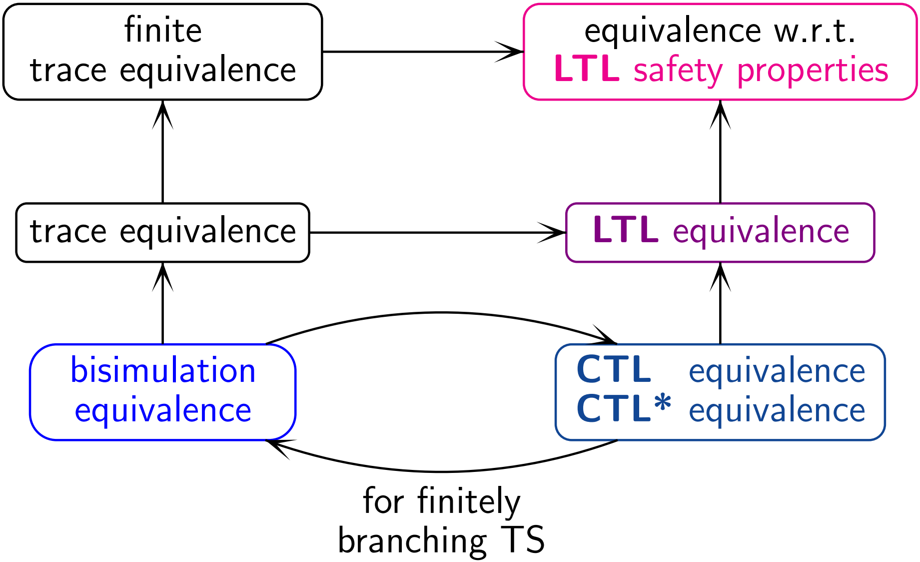
# Summary: equivalences





# Summary: equivalences





# Correct or wrong?



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

> If $s_1$, $s_2$ satisfy the same $\mathrm{CTL}_{\setminus \mathsf{U}}$ formulas then
>
> $s_1 \sim_{\mathcal{T}} s_2$.

# Correct or wrong?



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

> If $s_1$, $s_2$ satisfy the same $\mathrm{CTL}_{\setminus \mathbf{U}}$ formulas then
> $s_1 \sim_{\mathcal{T}} s_2$.

where $\mathrm{CTL}_{\setminus \mathbf{U}} \mathrel{\widehat{=}} \mathrm{CTL}$ without until operator $\mathbf{U}$

# Correct or wrong?



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

| |
|---|
| If $s_1$, $s_2$ satisfy the same $\text{CTL}_{\setminus U}$ formulas then $s_1 \sim_{\mathcal{T}} s_2$. |

where $\text{CTL}_{\setminus U} \mathrel{\widehat{=}} \text{CTL}$ without until operator **U**

**correct.**

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

> If $s_1$, $s_2$ satisfy the same $\mathrm{CTL}_{\setminus \mathrm{U}}$ formulas then
> $s_1 \sim_{\mathcal{T}} s_2$.

where $\mathrm{CTL}_{\setminus \mathrm{U}} \mathrel{\widehat{=}}$ CTL without until operator **U**

**correct.** see the proof

"**CTL** equivalence $\Longrightarrow$ bisimulation equivalence"

# $CTL_{\setminus U}$-equivalence $\Rightarrow$ bisimulation equivalence 

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

> If $s_1$, $s_2$ satisfy the same $CTL_{\setminus U}$ formulas then
> $s_1 \sim_{\mathcal{T}} s_2$.

*Proof.* Show that $CTL_{\setminus U}$ equivalence is a bisimulation

# $\text{CTL}_{\setminus U}$-equivalence $\Rightarrow$ bisimulation equivalence



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

> If $s_1$, $s_2$ satisfy the same $\text{CTL}_{\setminus U}$ formulas then
> $$s_1 \sim_{\mathcal{T}} s_2.$$

*Proof.* Show that $\text{CTL}_{\setminus U}$ equivalence is a bisimulation

- labeling condition only uses atomic propositions

# $\text{CTL}_{\setminus U}$-equivalence $\Rightarrow$ bisimulation equivalence 

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

> If $s_1$, $s_2$ satisfy the same $\text{CTL}_{\setminus U}$ formulas then
> $$s_1 \sim_{\mathcal{T}} s_2.$$

*Proof.* Show that $\text{CTL}_{\setminus U}$ equivalence is a bisimulation

- labeling condition only uses atomic propositions
- simulation condition can be established by $\text{CTL}_{\setminus U}$ master formulas of the form:

# $\mathrm{CTL}_{\setminus \mathrm{U}}$-equivalence ⇒ bisimulation equivalence 

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

> If $s_1$, $s_2$ satisfy the same $\mathrm{CTL}_{\setminus \mathrm{U}}$ formulas then
> $s_1 \sim_{\mathcal{T}} s_2$.

*Proof.* Show that $\mathrm{CTL}_{\setminus \mathrm{U}}$ equivalence is a bisimulation

- labeling condition only uses atomic propositions
- simulation condition can be established by $\mathrm{CTL}_{\setminus \mathrm{U}}$ master formulas of the form:

$$\exists \bigcirc \Phi_C \quad \text{where} \quad \Phi_C = \bigwedge_D \Phi_{C,D}$$

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

> If $s_1$, $s_2$ satisfy the same $\mathrm{CTL}_{\setminus \mathrm{U}}$ formulas then
> $s_1 \sim_{\mathcal{T}} s_2$.

*Proof.* Show that $\mathrm{CTL}_{\setminus \mathrm{U}}$ equivalence is a bisimulation

- labeling condition only uses atomic propositions
- simulation condition can be established by $\mathrm{CTL}_{\setminus \mathrm{U}}$ master formulas of the form:

$$\exists \bigcirc \Phi_C \quad \text{where} \quad \Phi_C = \bigwedge_D \Phi_{C,D}$$

$$\text{and} \quad Sat(\Phi_{C,D}) \subseteq C \setminus D$$

# Correct or wrong?



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

$\mathcal{T}$ and its bisimulation quotient $\mathcal{T}/\sim$ satisfy the same **CTL*** formulas.

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

$\mathcal{T}$ and its bisimulation quotient $\mathcal{T}/\sim$ satisfy the same **CTL*** formulas.

**correct.**

# Correct or wrong?



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

$\mathcal{T}$ and its bisimulation quotient $\mathcal{T}/\!\sim$ satisfy the same **CTL*** formulas.

**correct.** Recall that $\mathcal{T} \sim \mathcal{T}/\!\sim$

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

> $\mathcal{T}$ and its bisimulation quotient $\mathcal{T}/\!\sim$ satisfy the same **CTL*** formulas.

**correct.** Recall that $\mathcal{T} \sim \mathcal{T}/\!\sim$ as

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

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

here: $[s] = \sim_{\mathcal{T}}$-equivalence class of state $s$