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

Recall: CTL*

CTL* state formulas

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

CTL* path formulas

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

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**

Recall: CTL*

CTL* state formulas

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

CTL* path formulas

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

derived operators:

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

Recall: CTL* and CTL

CTL* state formulas

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

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***

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 $\forall$

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

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

CTL: sublogic of **CTL***

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

Recall: CTL* and CTL

CTL* state formulas

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

CTL* path formulas

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

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

CTL equivalence

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

CTL equivalence

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

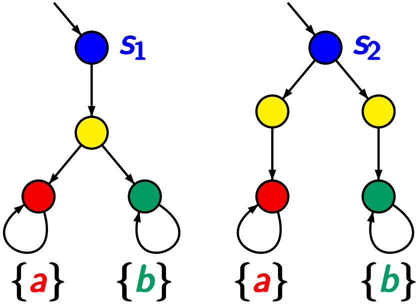
s_1, s_2 are **CTL** equivalent if for all **CTL** formulas Φ:

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

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

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

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

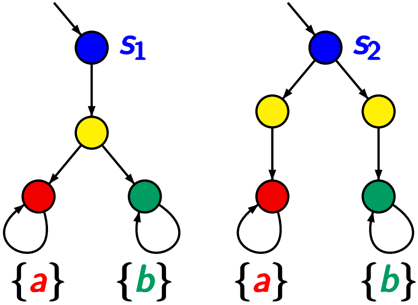


CTL equivalence

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

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

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



s_1, s_2 are not **CTL** equivalent

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

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

CTL, CTL* and LTL equivalence

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

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

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

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

CTL, CTL* and LTL equivalence

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

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

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

---

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

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

---

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

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

CTL/CTL* and bisimulation

CTL/CTL* and bisimulation

bisimulation equivalence

= **CTL** equivalence

= **CTL*** equivalence

CTL/CTL* and bisimulation

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

⟵ for finite TS

CTL/CTL* and bisimulation

	bisimulation equivalence	
=	**CTL** equivalence	⟵ for finite TS
=	**CTL*** equivalence	

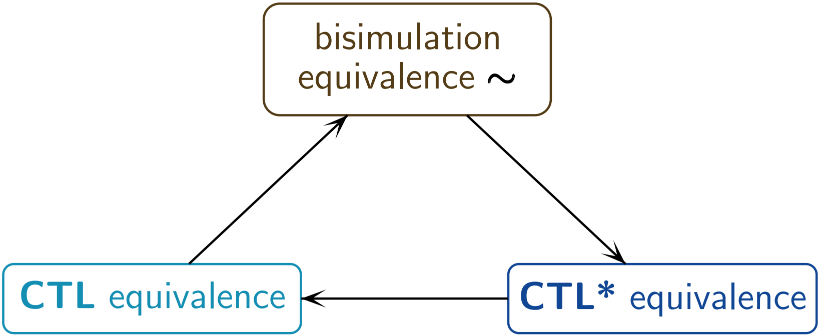
Let $\mathcal{T}$ be a finite TS without terminal states, and s_1, s_2 states in $\mathcal{T}$. Then:

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

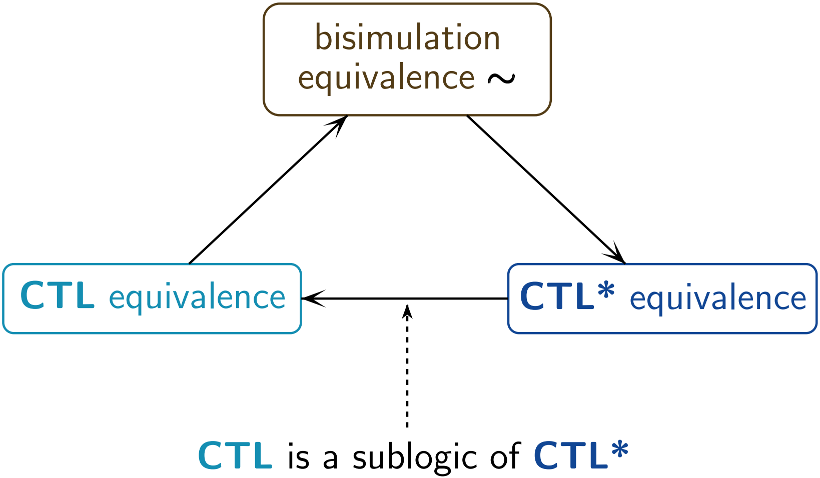
iff s_1 and s_2 are **CTL** equivalent

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

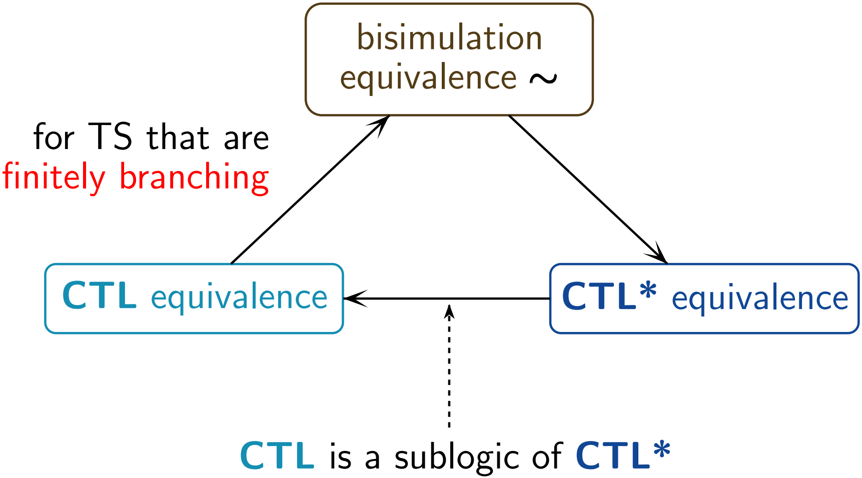
CTL/CTL* and bisimulation



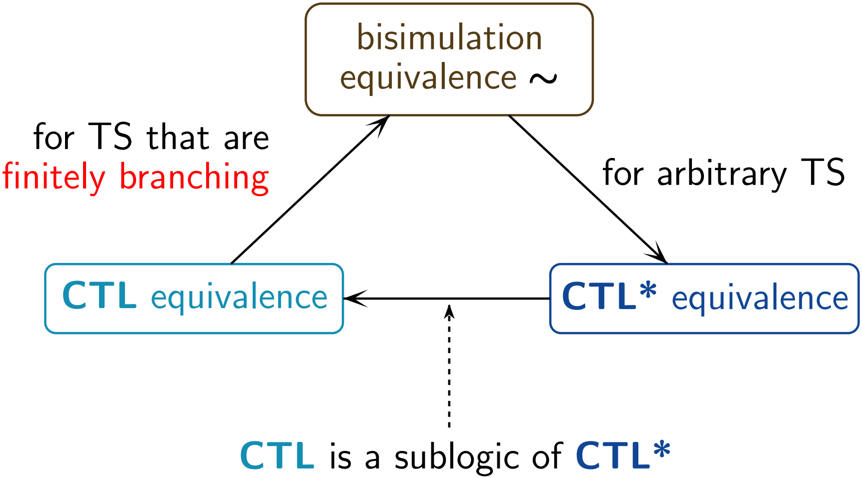
CTL/CTL* and bisimulation



CTL/CTL* and bisimulation



CTL/CTL* and bisimulation



Bisimulation equivalence $\Rightarrow$ 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 Φ:

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

Bisimulation equivalence $\Rightarrow$ 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 Φ:

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

(b) if π_1, π_2 are paths with $\pi_1 \sim_{\mathcal{T}} \pi_2$ then for all **CTL*** path formulas φ:

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

Bisimulation equivalence $\Rightarrow$ 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 Φ:

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

(b) if π_1, π_2 are paths with $\pi_1 \sim_{\mathcal{T}} \pi_2$ then for all **CTL*** path formulas φ:

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

$\pi_1 \sim_{\mathcal{T}} \pi_2 \quad \overset{\text{def}}{\Longleftrightarrow} \quad \pi_1$ and π_2 are statewise bisimulation equivalent

Bisimulation equivalence ⇒ 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 Φ and path formulas φ:

(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 Φ and path formulas φ:

(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 ⇒ CTL* equivalence

For all CTL* state formulas Φ and path formulas φ:

(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 Φ
s.t. statement (a) holds for Φ

For all CTL* state formulas Φ and path formulas φ:

(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 Φ_1, Φ_2, Ψ

(b) holds for φ

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

(a) holds for $\varphi_1, \varphi_2, \varphi'$

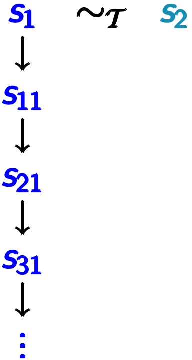
Path lifting for $\sim_{\mathcal{T}}$

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

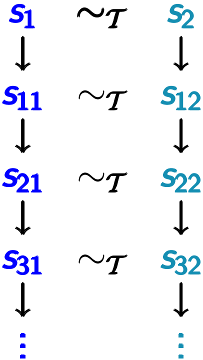
can be completed to

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

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

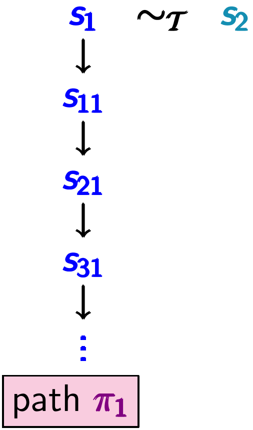


can be completed to

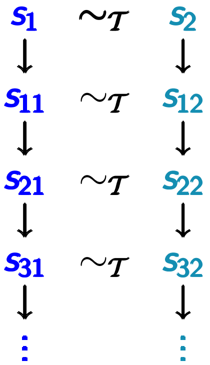


If $s_1 \sim_{\mathcal{T}} s_2$ then for all $\pi_1 \in Paths(s_1)$ there exists $\pi_2 \in Paths(s_2)$ with $\pi_1 \sim_{\mathcal{T}} \pi_2$

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

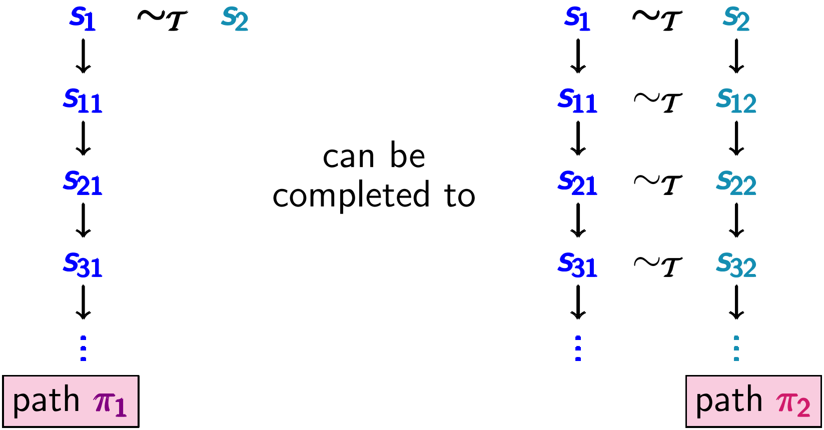


can be completed to



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

Correct or wrong?

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

correct.

> If s_1, s_2 are <u>not</u> **CTL** equivalent then there exists a **CTL** formula Φ 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 Φ with

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

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

If s_1, s_2 are not **CTL** equivalent then there exists a **CTL** formula Φ 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 Φ with

$$\begin{array}{ll} & s_1 \models \Phi \wedge s_2 \not\models \Phi \\ \text{or} & s_1 \not\models \Phi \wedge s_2 \models \Phi \implies s_1 \models \neg\Phi \wedge s_2 \not\models \neg\Phi \end{array}$$

Correct or wrong?

> If s_1, s_2 are not **CTL** equivalent then there exists a **CTL** formula Φ 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 φ 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 Φ 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 φ with $s_1 \models \varphi$ and $s_2 \not\models \varphi$

wrong.

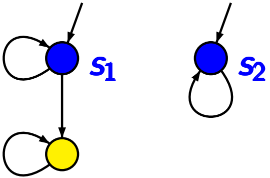
Correct or wrong?

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

correct.

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

wrong.



Correct or wrong?

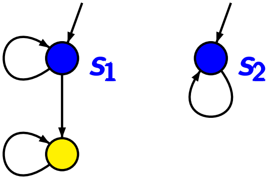
If s_1, s_2 are <u>not</u> **CTL** equivalent then there exists a **CTL** formula Φ with $s_1 \models \Phi$ and $s_2 \not\models \Phi$

correct.

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

wrong.

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



Correct or wrong?

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

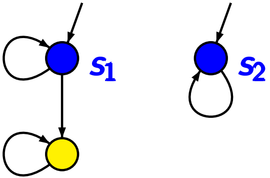
correct.

If s_1, s_2 are <u>not</u> **LTL** equivalent then there exists a **LTL** formula φ 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

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

is a bisimulation

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

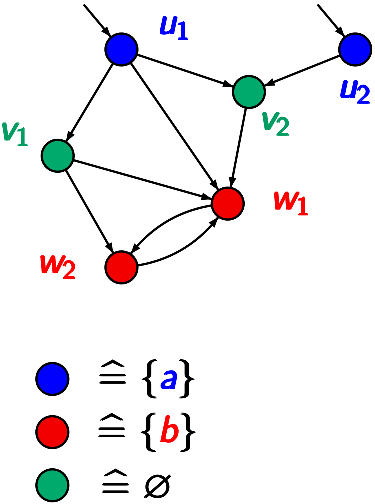
$$\mathcal{R} \stackrel{\text{def}}{=} \{ (s_1, s_2) : s_1, s_2 \text{ satisfy the same } \textbf{CTL} \text{ formulas} \}$$

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

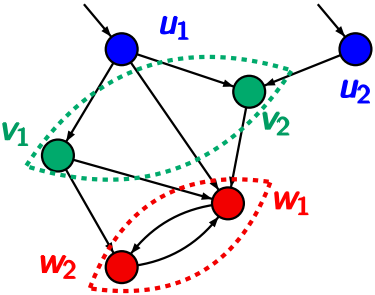
(1) $L(s_1) = L(s_2)$

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

Example: CTL master formulas



Example: CTL master formulas



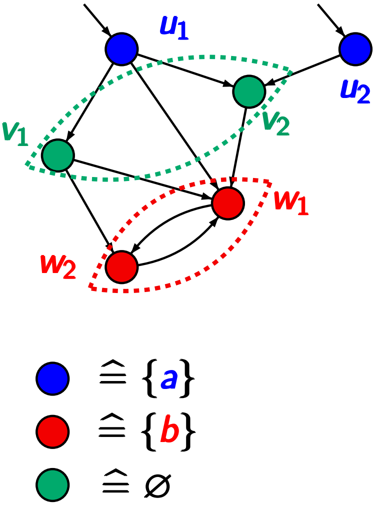
bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

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

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

 $\widehat{=} \varnothing$

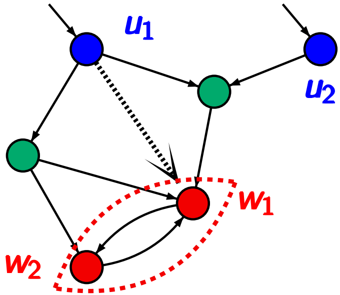
Example: CTL master formulas



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

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

Example: CTL master formulas

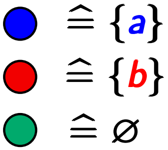


bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

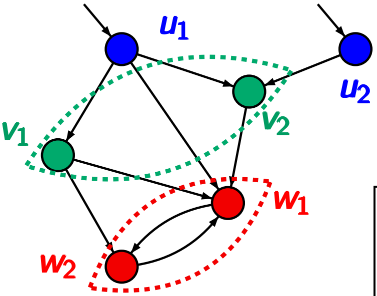
but $u_1 \not\sim_{\mathcal{T}} u_2$

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

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



Example: CTL master formulas



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

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

CTL master formulas:

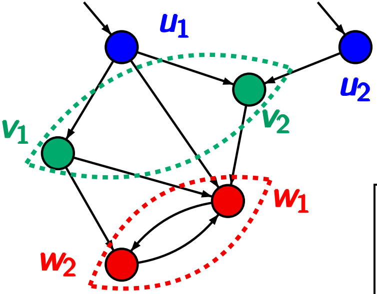
$w_1, w_2 \models$?

$v_1, v_2 \models$?

$u_1 \models$?

$u_2 \models$?

Example: CTL master formulas



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

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

$\widehat{=} \varnothing$

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

CTL master formulas:

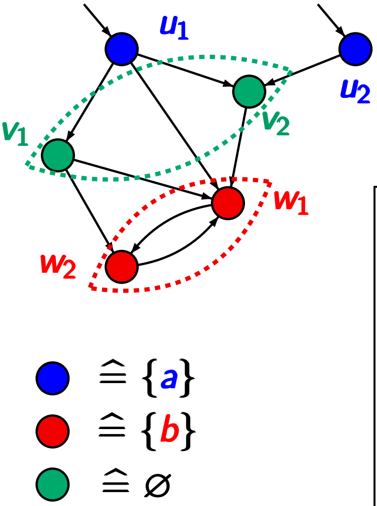
$w_1, w_2 \models b$

$v_1, v_2 \models ?$

$u_1 \models ?$

$u_2 \models ?$

Example: CTL master formulas



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

CTL master formulas:

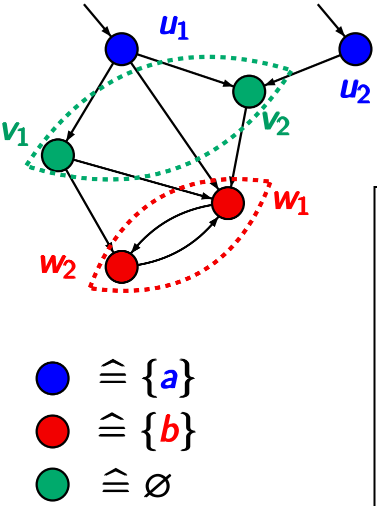
$w_1, w_2 \models b$

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

$u_1 \models ?$

$u_2 \models ?$

Example: CTL master formulas

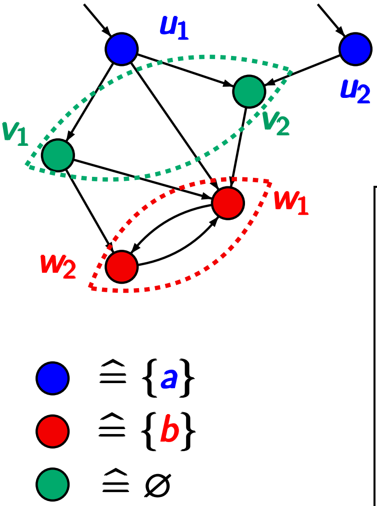


bisimulation equivalence $\sim_{\mathcal{T}}$
$= \{(v_1, v_2), (w_1, w_2), \ldots\}$

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

CTL master formulas:

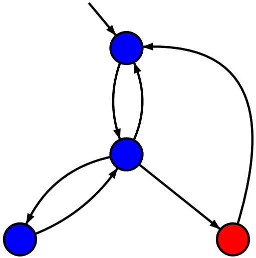
$$w_1, w_2 \models b$$

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

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

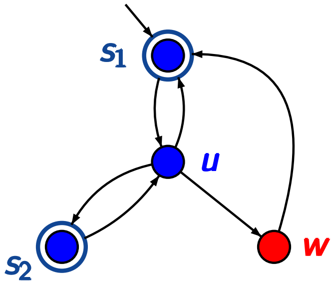
$$u_2 \models (\neg \exists \bigcirc b) \wedge a$$

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



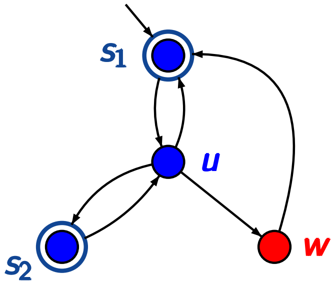
$AP = \{blue, red\}$

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



$AP = \{blue, red\}$

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



$AP = \{blue, red\}$

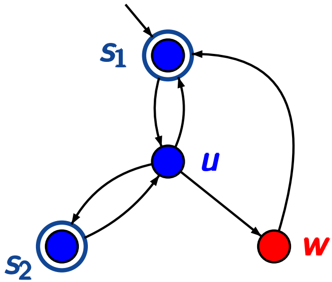
$s_1 \sim_{\mathcal{T}} s_2 \not\sim_{\mathcal{T}} u$

$\Phi_w \;=\; ?$

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

$\Phi_u \;=\; ?$

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



$AP = \{blue, red\}$

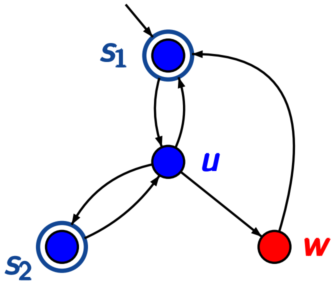
$s_1 \sim_{\mathcal{T}} s_2 \not\sim_{\mathcal{T}} u$

$$\Phi_w = red$$

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

$$\Phi_u = ?$$

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



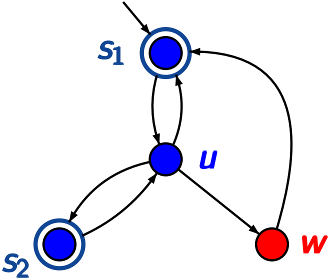
$AP = \{blue, red\}$

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

$$\Phi_w = red$$

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

$$\Phi_u = ?$$



$AP = \{blue, red\}$

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

$$\Phi_w = red$$

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

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

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

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

↑

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 $s \in S$

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

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.

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

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

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

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

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

Summary: CTL/CTL* and bisimulation

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:

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

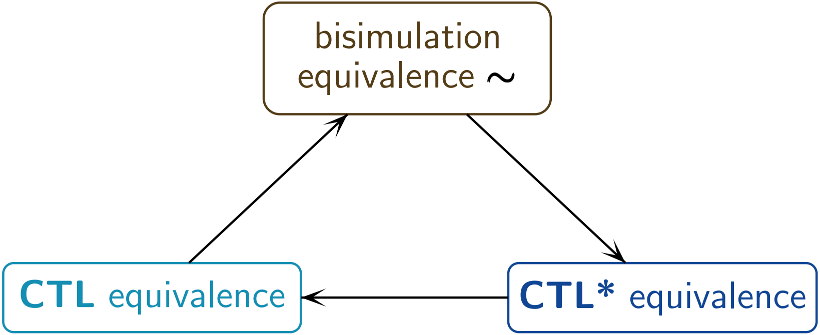
iff s_1 and s_2 are **CTL** equivalent

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

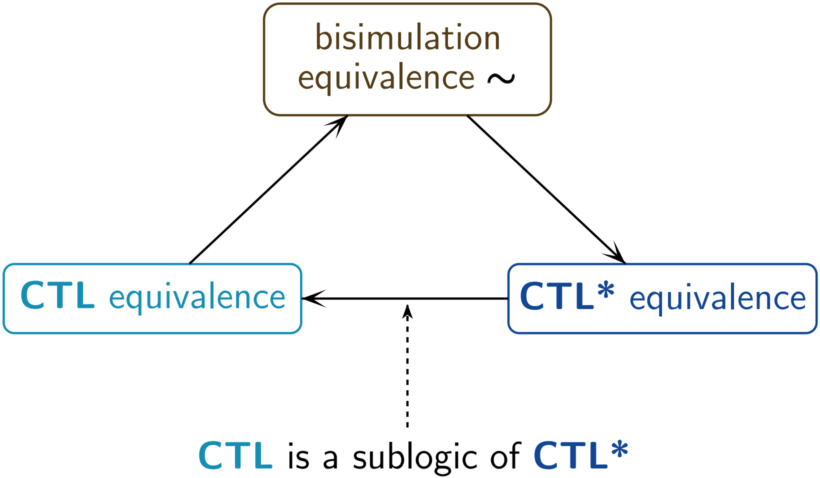
Let $\mathcal{T}$ be a finitely branching TS without terminal states, and s_1, s_2 states in $\mathcal{T}$. Then:

$$
\begin{aligned}
& s_1 \sim_{\mathcal{T}} s_2 \\
\text{iff} \quad & s_1 \text{ and } s_2 \text{ are } \textbf{CTL} \text{ equivalent} \\
\text{iff} \quad & s_1 \text{ and } s_2 \text{ are } \textbf{CTL*} \text{ equivalent}
\end{aligned}
$$

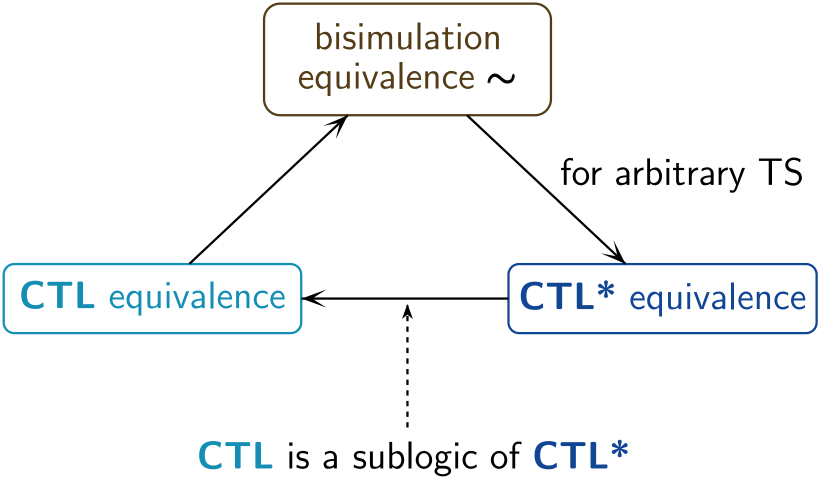
Summary: CTL/CTL* and bisimulation



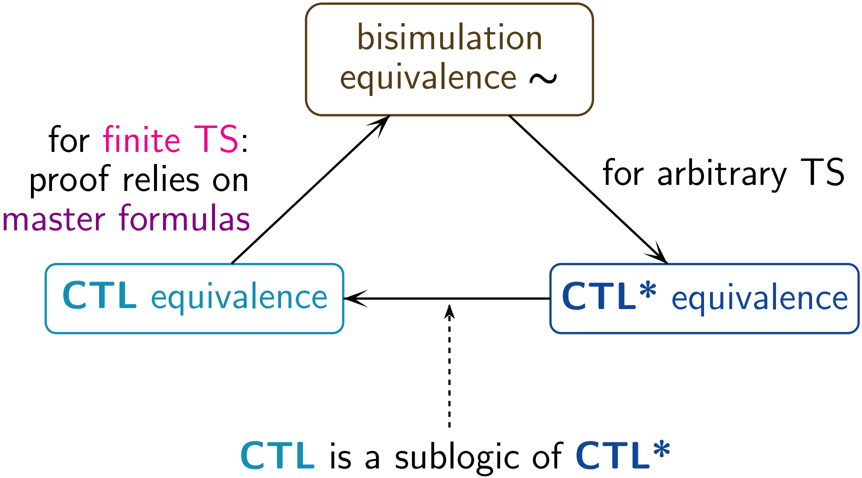
Summary: CTL/CTL* and bisimulation



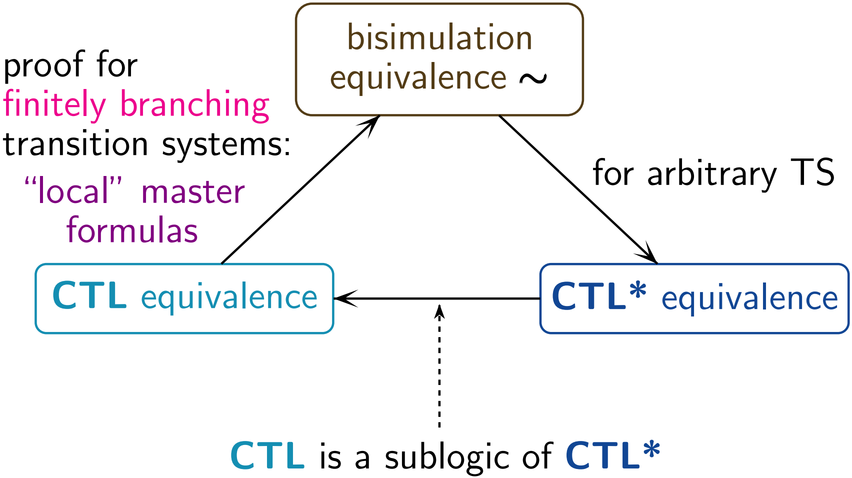
Summary: CTL/CTL* and bisimulation



Summary: CTL/CTL* and bisimulation



Summary: CTL/CTL* and bisimulation



CTL/CTL* and bisimulation for TS

CTLEQ5.2-2-FOR-2-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.

Correct or wrong?

CTL equivalence is finer than **LTL** equivalence

correct.

↑

CTL equivalence = **CTL*** equivalence

LTL is sublogic of **CTL***

Correct or wrong?

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

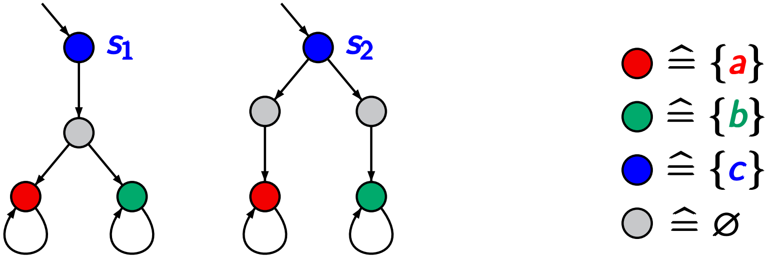
wrong.

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.



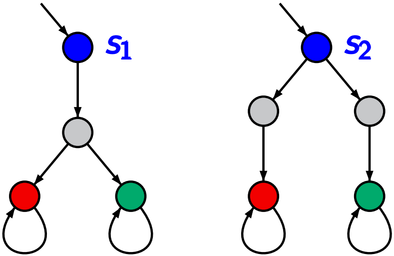
Correct or wrong?

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.



s_1, s_2 are trace equivalent

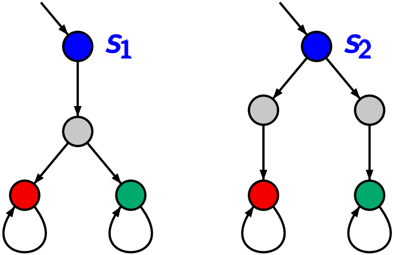
Correct or wrong?

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

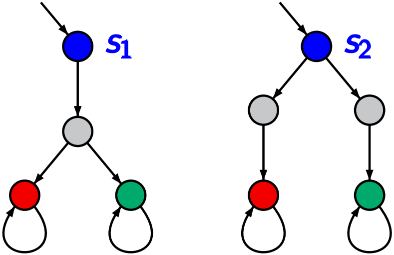
Correct or wrong?

CTL equivalence is finer than **LTL** equivalence

correct.

LTL equivalence is finer than **CTL** equivalence

wrong.

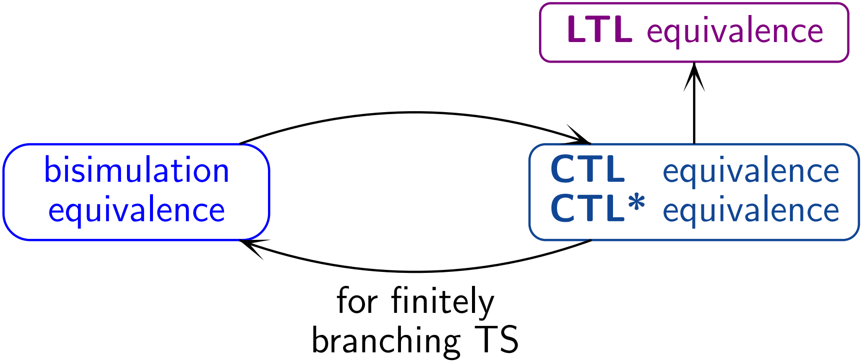


s_1, s_2 are trace equivalent

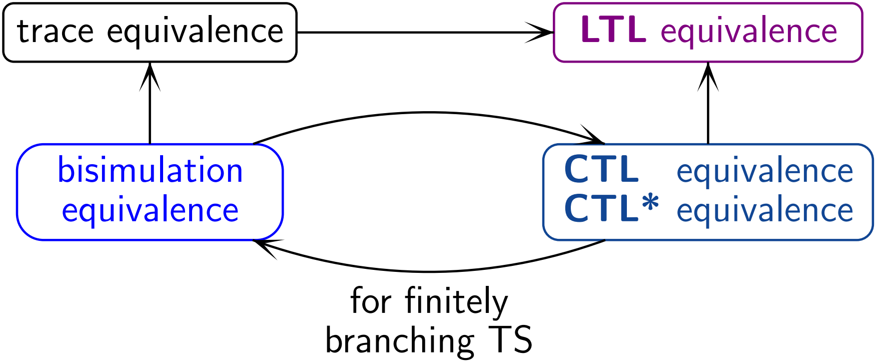
and **LTL** equivalent

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

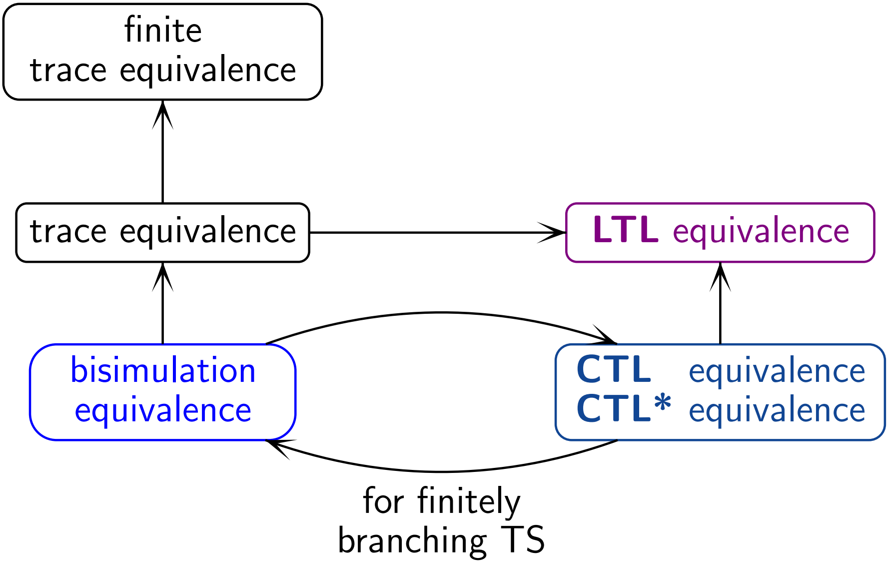
Summary: equivalences



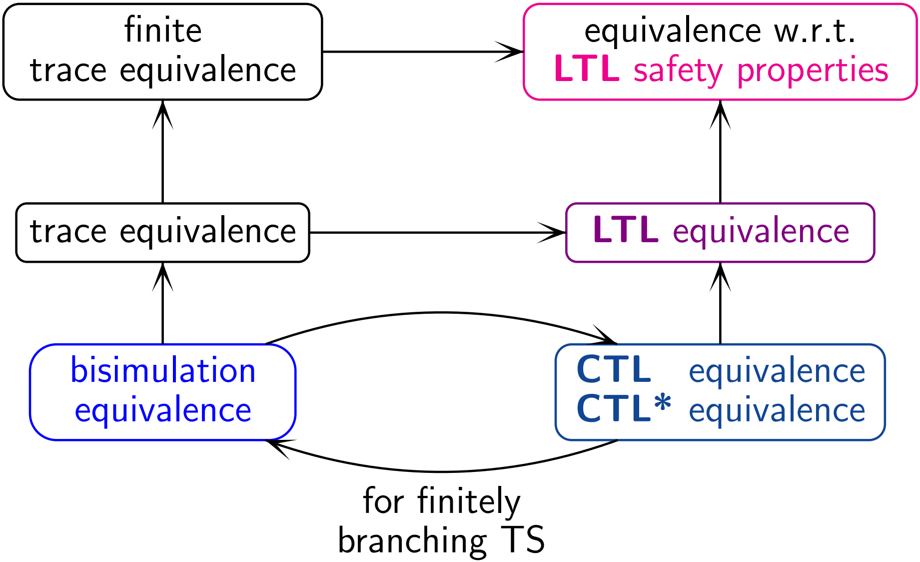
Summary: equivalences



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 $\mathrm{CTL}_{\setminus \mathrm{U}}$ formulas then $s_1 \sim_{\mathcal{T}} s_2$.

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

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

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

where $CTL_{\setminus U} \,\widehat{=}\, CTL$ without until operator $\mathbf{U}$

correct.

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

correct. see the proof

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

$\mathrm{CTL}_{\setminus \mathrm{U}}$-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 $\mathrm{CTL}_{\setminus \mathrm{U}}$ formulas then
> $s_1 \sim_{\mathcal{T}} s_2$.

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

$\mathbf{CTL}_{\setminus U}$-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 $\mathrm{CTL}_{\setminus U}$ formulas then
> $s_1 \sim_{\mathcal{T}} s_2$.

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

- labeling condition only uses atomic propositions

$\mathrm{CTL}_{\setminus \mathrm{U}}$-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 $\mathrm{CTL}_{\setminus \mathrm{U}}$ 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:

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:

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

$\mathsf{CTL}_{\setminus \mathsf{U}}$-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 $\mathsf{CTL}_{\setminus \mathsf{U}}$ formulas then $s_1 \sim_{\mathcal{T}} s_2$.

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

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

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

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

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 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.

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

Correct or wrong?

Let $\mathcal{T}$ be a finite TS without terminal states and let *fair* be a **CTL** fairness assumption.

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL** formulas Φ:

$$s_1 \models_{fair} \Phi \quad \text{iff} \quad s_2 \models_{fair} \Phi$$

Correct or wrong?

Let $\mathcal{T}$ be a finite TS without terminal states and let *fair* be a **CTL** fairness assumption.

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL** formulas Φ:

$$s_1 \models_{fair} \Phi \quad \text{iff} \quad s_2 \models_{fair} \Phi$$

correct

Correct or wrong?

Let $\mathcal{T}$ be a finite TS without terminal states and let *fair* be a **CTL** fairness assumption.

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL** formulas Φ:

$$s_1 \models_{fair} \Phi \quad \text{iff} \quad s_2 \models_{fair} \Phi$$

correct, as $\models_{fair}$ is "**CTL***-definable"

Let $\mathcal{T}$ be a finite TS without terminal states and let *fair* be a **CTL** fairness assumption.

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL** formulas Φ:

$$s_1 \models_{fair} \Phi \quad \text{iff} \quad s_2 \models_{fair} \Phi$$

correct, as $\models_{fair}$ is "**CTL***-definable"

↑

For each **CTL*** state formula Φ there exists a **CTL*** formula Ψ s.t. $s \models \Psi$ iff $s \models_{fair} \Phi$

Correct or wrong?

Let $\mathcal{T}$ be a finite TS without terminal states and let *fair* be a **CTL** fairness assumption.

> If $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL** formulas Φ:
>
> $$s_1 \models_{fair} \Phi \quad \text{iff} \quad s_2 \models_{fair} \Phi$$

correct, as $\models_{fair}$ is "**CTL***-definable"

> For each **CTL*** state formula Φ there exists a **CTL*** formula Ψ s.t. $s \models \Psi$ iff $s \models_{fair} \Phi$

Example: for $\Phi = \exists\Box(a \wedge \forall\Diamond b)$

Let $\mathcal{T}$ be a finite TS without terminal states and let *fair* be a **CTL** fairness assumption.

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **CTL** formulas Φ:

$$s_1 \models_{fair} \Phi \quad \text{iff} \quad s_2 \models_{fair} \Phi$$

correct, as $\models_{fair}$ is "**CTL***-definable"

For each **CTL*** state formula Φ there exists a **CTL*** formula Ψ s.t. $s \models \Psi$ iff $s \models_{fair} \Phi$

Example: for $\Phi = \exists\Box(a \wedge \forall\Diamond b)$

$$\Psi = \exists\big(fair \wedge \Box(a \wedge \forall(fair \rightarrow \Diamond b))\big)$$

Correct or wrong?

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

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

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

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

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

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

correct.

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

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

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

correct.

Note that:

$$(1) \quad s_1 \sim_{\mathcal{T}} s_2 \implies Traces(s_1) = Traces(s_2)$$

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

> If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:
> $$s_1 \models E \quad \text{iff} \quad s_2 \models E$$

correct.

Note that:

$$(1) \quad s_1 \sim_{\mathcal{T}} s_2 \implies Traces(s_1) = Traces(s_2)$$

$$(2) \quad s \models E \iff Traces(s) \subseteq E$$

Correct or wrong?

Let $\mathcal{F}$ be an action-based strong fairness assumption

e.g., strong fairness for a single action α

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

Let $\mathcal{F}$ be an action-based strong fairness assumption

e.g., strong fairness for a single action α

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

wrong.

Correct or wrong?

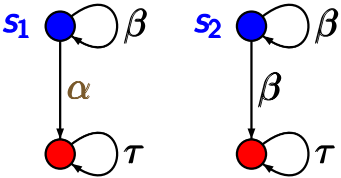
Let $\mathcal{F}$ be an action-based strong fairness assumption

e.g., strong fairness for a single action α

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

wrong.



$\mathcal{F} \mathrel{\widehat{=}}$ strong fairness assumption for action α

Let $\mathcal{F}$ be an action-based strong fairness assumption

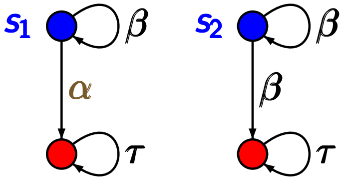
e.g., strong fairness for a single action α

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

wrong.

$E \mathrel{\widehat{=}} \Diamond red$



$\mathcal{F} \mathrel{\widehat{=}}$ strong fairness assumption for action α

Let $\mathcal{F}$ be an action-based strong fairness assumption

e.g., strong fairness for a single action α

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

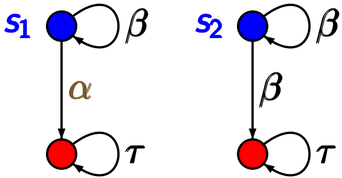
$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

wrong.

$E \mathrel{\widehat{=}} \Diamond red$

$s_1 \models_{\mathcal{F}} E$

$s_2 \not\models_{\mathcal{F}} E$



$\mathcal{F} \mathrel{\widehat{=}}$ strong fairness assumption for action α

Correct or wrong?

Let $\mathcal{F}$ be an action-based strong fairness assumption

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$: $s_1 \models_{\mathcal{F}} E$ iff $s_2 \models_{\mathcal{F}} E$

wrong.

If $s_1 \sim_{\mathcal{T}} s_2$ then for all safety properties $E \subseteq (2^{AP})^\omega$: $s_1 \models_{\mathcal{F}} E$ iff $s_2 \models_{\mathcal{F}} E$

Correct or wrong?

Let $\mathcal{F}$ be an action-based strong fairness assumption

> If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:
> $$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

wrong.

> If $s_1 \sim_{\mathcal{T}} s_2$ then for all safety properties $E \subseteq (2^{AP})^\omega$:
> $$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

correct.

Let $\mathcal{F}$ be an action-based strong fairness assumption

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

wrong.

If $s_1 \sim_{\mathcal{T}} s_2$ then for all safety properties $E \subseteq (2^{AP})^\omega$:

$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

correct.

- realizable fairness irrelevant for safety properties

Correct or wrong?

Let $\mathcal{F}$ be an action-based strong fairness assumption

If $s_1 \sim_{\mathcal{T}} s_2$ then for all **LT** properties $E \subseteq (2^{AP})^\omega$:

$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

wrong.

If $s_1 \sim_{\mathcal{T}} s_2$ then for all safety properties $E \subseteq (2^{AP})^\omega$:

$$s_1 \models_{\mathcal{F}} E \quad \text{iff} \quad s_2 \models_{\mathcal{F}} E$$

correct.

- realizable fairness irrelevant for safety properties
- strong action-based fairness assumptions are realizable