

Introduction

Modelling parallel systems

Linear Time Properties

Regular Properties

Linear Temporal Logic (LTL)

Computation Tree Logic

 syntax and semantics of CTL

 expressiveness of CTL and LTL

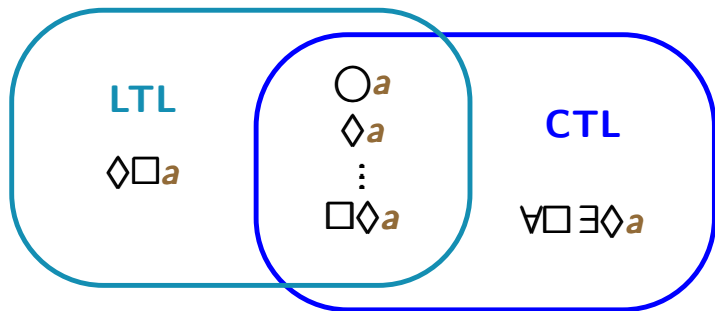
 CTL model checking

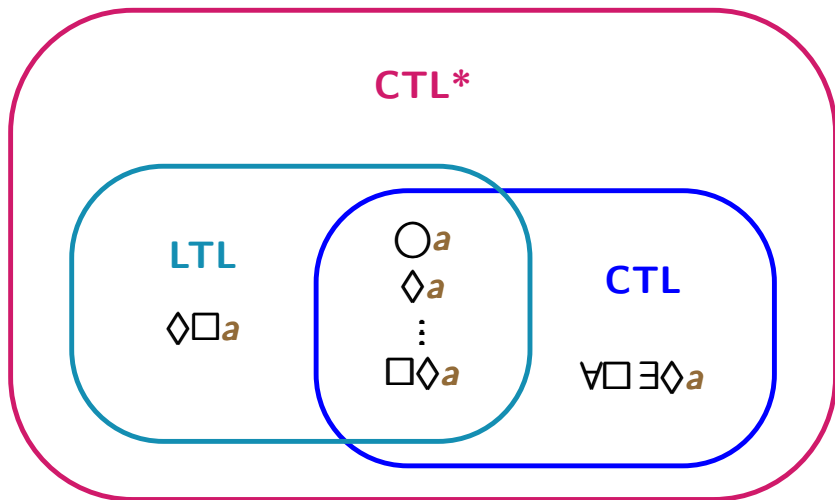
 fairness, counterexamples/witnesses

 CTL⁺ and CTL*



Equivalences and Abstraction





state formulas:

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

path formulas:

$$\varphi ::= \dots$$

state formulas:

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

path formulas:

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

state formulas:

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

path formulas:

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

derived operators:

- \forall, \rightarrow , etc.
- eventually, always

state formulas:

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

path formulas:

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

derived operators:

- \forall, \rightarrow , etc.
- eventually, always as in **LTL**:

$$\diamond\varphi = \text{true} \mathbf{U} \varphi, \quad \square\varphi = \neg\diamond\neg\varphi$$

state formulas:

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

path formulas:

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

derived operators:

- \forall, \rightarrow , etc.
- eventually, always as in **LTL**:

$$\diamond\varphi = \text{true} \mathbf{U} \varphi, \quad \square\varphi = \neg\diamond\neg\varphi$$

- universal quantification:

state formulas:

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

path formulas:

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

derived operators:

- \forall, \rightarrow , etc.
- eventually, always as in **LTL**:

$$\diamond\varphi = \text{true} \mathbf{U} \varphi, \quad \square\varphi = \neg\diamond\neg\varphi$$

- universal quantification: $\forall\varphi = \neg\exists\neg\varphi$

Let $\mathcal{T} = (\mathcal{S}, Act, \rightarrow, \mathcal{S}_0, AP, L)$ be a transition system without terminal states.

Let $\mathcal{T} = (\mathcal{S}, Act, \rightarrow, s_0, AP, L)$ be a transition system without terminal states.

define by structural induction:

- a satisfaction relation \models for states $s \in \mathcal{S}$ and **CTL*** state formulas
- a satisfaction relation \models for infinite path fragments π in \mathcal{T} and **CTL*** path formulas

$s \models \text{true}$

$s \models a$ iff $a \in L(s)$

$s \models \neg\phi$ iff $s \not\models \phi$

$s \models \phi_1 \wedge \phi_2$ iff $s \models \phi_1$ and $s \models \phi_2$

$s \models \exists\psi$ iff there exists a path $\pi \in \text{Paths}(s)$
such that $\pi \models \psi$

$s \models \text{true}$ $s \models a$ iff $a \in L(s)$ $s \models \neg\phi$ iff $s \not\models \phi$ $s \models \phi_1 \wedge \phi_2$ iff $s \models \phi_1$ and $s \models \phi_2$ $s \models \exists\psi$ iff there exists a path $\pi \in \text{Paths}(s)$
such that $\pi \models \psi$

↑
satisfaction relation \models
for CTL* path formulas

let $\pi = s_0 s_1 s_2 \dots$ be an infinite path fragment in \mathcal{T}

let $\pi = s_0 s_1 s_2 \dots$ be an infinite path fragment in \mathcal{T}

$\pi \models \Phi$ iff ...

$\pi \models \neg\varphi$ iff $\pi \not\models \varphi$

$\pi \models \varphi_1 \wedge \varphi_2$ iff $\pi \models \varphi_1$ and $\pi \models \varphi_2$

$\pi \models \bigcirc\varphi$ iff $\text{suffix}(\pi, 1) \models \varphi$

$\pi \models \varphi_1 \mathbf{U} \varphi_2$ iff there exists $j \geq 0$ such that
 $\text{suffix}(\pi, j) \models \varphi_2$
 $\text{suffix}(\pi, i) \models \varphi_1$ for $0 \leq i < j$

let $\pi = s_0 s_1 s_2 \dots$ be an infinite path fragment in \mathcal{T}

$\pi \models \Phi$ iff ...

$\pi \models \neg\varphi$ iff $\pi \not\models \varphi$

$\pi \models \varphi_1 \wedge \varphi_2$ iff $\pi \models \varphi_1$ and $\pi \models \varphi_2$

$\pi \models \bigcirc\varphi$ iff $\text{suffix}(\pi, 1) \models \varphi$

$\pi \models \varphi_1 \mathbf{U} \varphi_2$ iff there exists $j \geq 0$ such that

$\text{suffix}(\pi, j) \models \varphi_2$

$\text{suffix}(\pi, i) \models \varphi_1$ for $0 \leq i < j$

$\text{suffix}(\pi, k) = s_k s_{k+1} s_{k+2} \dots$

let $\pi = s_0 s_1 s_2 \dots$ be an infinite path fragment in \mathcal{T}

$$\pi \models \Phi \quad \text{iff} \quad s_0 \models \Phi$$

$$\pi \models \neg\varphi \quad \text{iff} \quad \pi \not\models \varphi$$

$$\pi \models \varphi_1 \wedge \varphi_2 \quad \text{iff} \quad \pi \models \varphi_1 \text{ and } \pi \models \varphi_2$$

$$\pi \models \bigcirc\varphi \quad \text{iff} \quad \text{suffix}(\pi, 1) \models \varphi$$

$$\pi \models \varphi_1 \mathbf{U} \varphi_2 \quad \text{iff} \quad \text{there exists } j \geq 0 \text{ such that}$$

$$\text{suffix}(\pi, j) \models \varphi_2$$

$$\text{suffix}(\pi, i) \models \varphi_1 \quad \text{for } 0 \leq i < j$$

$$\text{suffix}(\pi, k) = s_k s_{k+1} s_{k+2} \dots$$

let $\pi = s_0 s_1 s_2 \dots$ be an infinite path fragment in \mathcal{T}

$\pi \models \Phi$	iff	$s_0 \models \Phi$	←	satisfaction relation for CTL* state formulas
$\pi \models \neg\varphi$	iff	$\pi \not\models \varphi$		
$\pi \models \varphi_1 \wedge \varphi_2$	iff	$\pi \models \varphi_1$ and $\pi \models \varphi_2$		
$\pi \models \bigcirc\varphi$	iff	$\text{suffix}(\pi, 1) \models \varphi$		
$\pi \models \varphi_1 \mathbf{U} \varphi_2$	iff	there exists $j \geq 0$ such that		
		$\text{suffix}(\pi, j) \models \varphi_2$		
		$\text{suffix}(\pi, i) \models \varphi_1$	for $0 \leq i < j$	

$$\text{suffix}(\pi, k) = s_k s_{k+1} s_{k+2} \dots$$

mutual exclusion:

safety $\forall \square (\neg \text{crit}_1 \vee \neg \text{crit}_2)$

mutual exclusion:

safety $\forall \square (\neg \text{crit}_1 \vee \neg \text{crit}_2)$

liveness $\forall \square \diamond \text{crit}_1 \wedge \forall \square \diamond \text{crit}_2$

mutual exclusion:

safety $\forall \square (\neg \textit{crit}_1 \vee \neg \textit{crit}_2)$

liveness $\forall \square \diamond \textit{crit}_1 \wedge \forall \square \diamond \textit{crit}_2$

progress property, e.g., $\forall \square (\textit{request} \rightarrow \diamond \textit{response})$

mutual exclusion:

safety $\forall \square (\neg \textit{crit}_1 \vee \neg \textit{crit}_2)$

liveness $\forall \square \diamond \textit{crit}_1 \wedge \forall \square \diamond \textit{crit}_2$

progress property, e.g., $\forall \square (\textit{request} \rightarrow \diamond \textit{response})$

persistence property, e.g., $\forall \diamond \square a$

mutual exclusion:

safety $\forall \square (\neg \text{crit}_1 \vee \neg \text{crit}_2)$

liveness $\forall \square \diamond \text{crit}_1 \wedge \forall \square \diamond \text{crit}_2$

progress property, e.g., $\forall \square (\text{request} \rightarrow \diamond \text{response})$

persistence property, e.g., $\forall \diamond \square a$

CTL* formulas with existential quantification, e.g.,
Hamilton path problem (for fixed initial state)

$$\exists \left(\bigwedge_{v \in V} (\diamond v \wedge \square (v \rightarrow \bigcirc \square \neg v)) \right)$$

- CTL is a sublogic of CTL*

- **CTL** is a sublogic of **CTL***

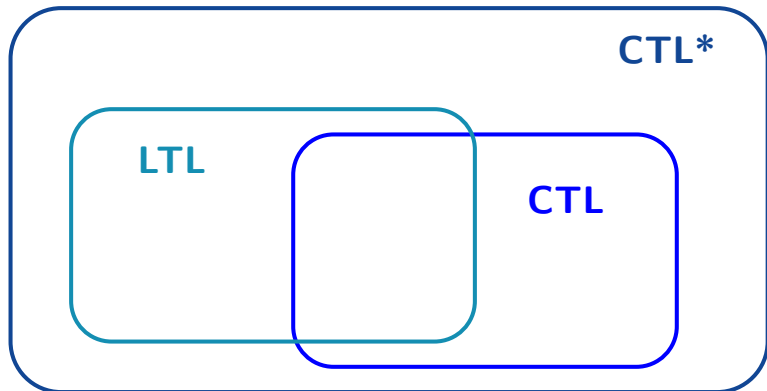


The diagram consists of two nested rounded rectangles. The outer rectangle is larger and contains the text 'CTL*' in its top right corner. The inner rectangle is smaller and is centered within the outer one, containing the text 'CTL' in its center. This visualizes that CTL is a subset of CTL*.

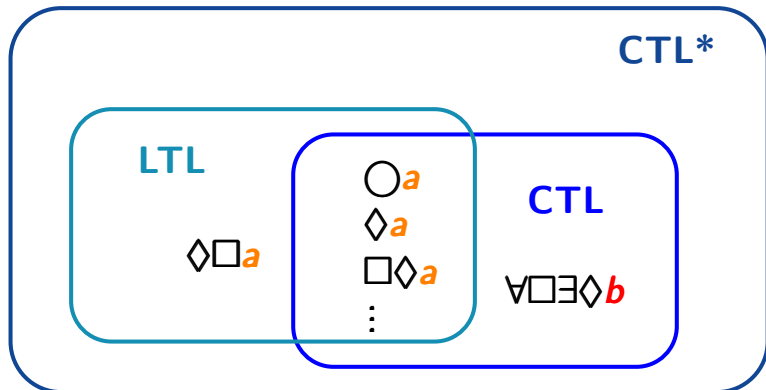
CTL*

CTL

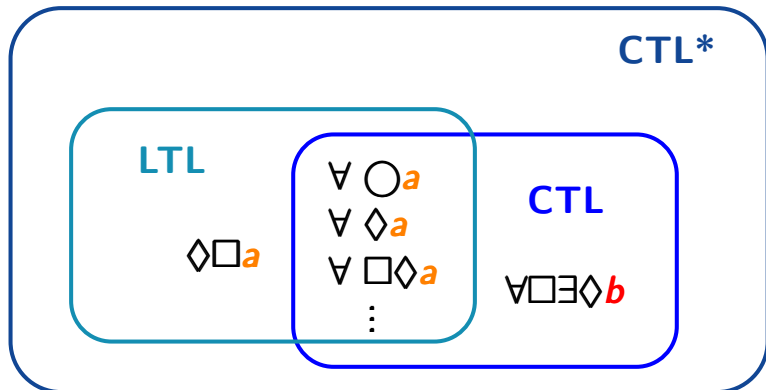
- **CTL** is a sublogic of **CTL***
- **LTL** is a sublogic of **CTL***



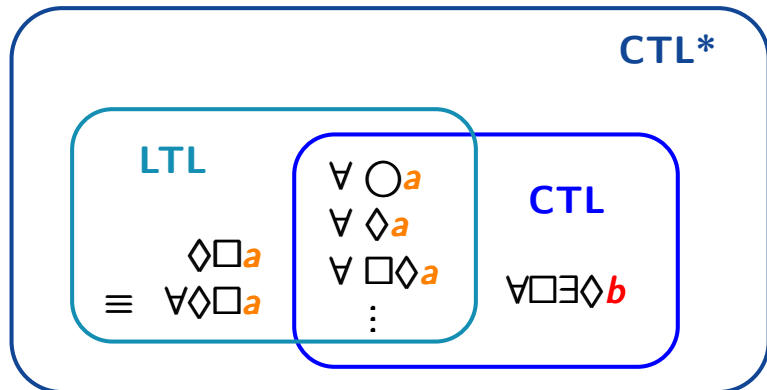
- **CTL** is a sublogic of **CTL***
- **LTL** is a sublogic of **CTL***



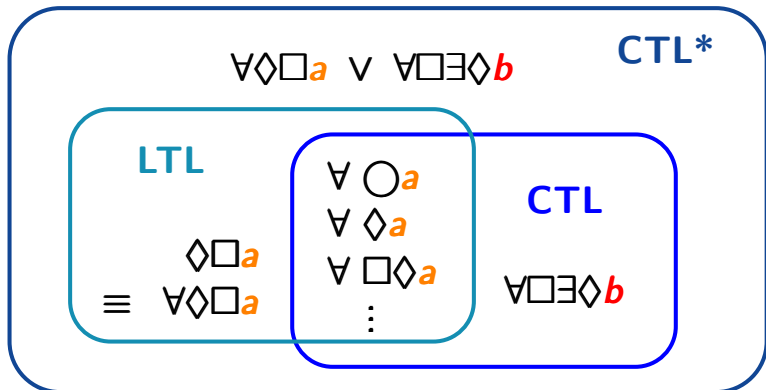
- **CTL** is a sublogic of **CTL***
- **LTL** is a sublogic of **CTL***



- **CTL** is a sublogic of **CTL***
- **LTL** is a sublogic of **CTL***



- **CTL** is a sublogic of **CTL***
- **LTL** is a sublogic of **CTL***
- **CTL*** is more expressive than **LTL** and **CTL**



$\Phi_1 \equiv \Phi_2$ iff for all transition systems \mathcal{T} :

$$\mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

$$\Phi_1 \equiv \Phi_2 \text{ iff for all transition systems } \mathcal{T}: \\ \mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

Examples:

$$\neg \exists \square \diamond a \equiv \forall \diamond \square \neg a$$

$$\forall \square \diamond a \equiv \forall \square \forall \diamond a$$

⋮

$$\Phi_1 \equiv \Phi_2 \text{ iff for all transition systems } \mathcal{T}: \\ \mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

Examples:

$$\neg \exists \square \diamond a \equiv \forall \diamond \square \neg a$$

$$\forall \square \diamond a \equiv \forall \square \forall \diamond a$$

⋮

$$\forall \forall \psi \equiv \forall \psi$$

$$\exists \exists \psi \equiv \exists \psi$$

$$\Phi_1 \equiv \Phi_2 \text{ iff for all transition systems } \mathcal{T}: \\ \mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

Examples:

$$\neg \exists \square \diamond a \equiv \forall \diamond \square \neg a$$

$$\forall \square \diamond a \equiv \forall \square \forall \diamond a$$

⋮

$$\forall \forall \varphi \equiv \forall \varphi$$

$$\exists \exists \varphi \equiv \exists \varphi$$

$$\forall \exists \varphi \equiv ?$$

$$\Phi_1 \equiv \Phi_2 \text{ iff for all transition systems } \mathcal{T}: \\ \mathcal{T} \models \Phi_1 \iff \mathcal{T} \models \Phi_2$$

Examples:

$$\neg \exists \square \diamond a \equiv \forall \diamond \square \neg a$$

$$\forall \square \diamond a \equiv \forall \square \forall \diamond a$$

⋮

$$\forall \forall \varphi \equiv \forall \varphi$$

$$\exists \exists \varphi \equiv \exists \varphi$$

$$\forall \exists \varphi \equiv \exists \varphi$$

Correct or wrong?

CTLST4.6-13

$$\forall(\varphi_1 \vee \varphi_2) \equiv \forall\varphi_1 \vee \forall\varphi_2$$

Correct or wrong?

CTLST4.6-13

$$\forall(\varphi_1 \vee \varphi_2) \equiv \forall\varphi_1 \vee \forall\varphi_2$$

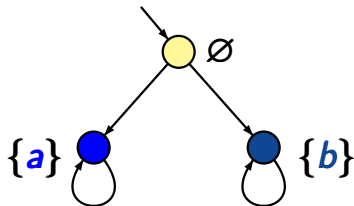
wrong, e.g., $\forall(\diamond a \vee \diamond b) \not\equiv \forall\diamond a \vee \forall\diamond b$

Correct or wrong?

CTLST4.6-13

$$\forall(\varphi_1 \vee \varphi_2) \equiv \forall\varphi_1 \vee \forall\varphi_2$$

wrong, e.g., $\forall(\diamond a \vee \diamond b) \not\equiv \forall\diamond a \vee \forall\diamond b$



$$\models \forall(\diamond a \vee \diamond b)$$

$$\not\models \forall\diamond a$$

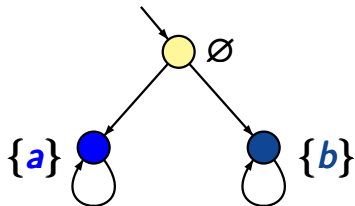
$$\not\models \forall\diamond b$$

Correct or wrong?

CTLST4.6-13

$$\forall(\varphi_1 \vee \varphi_2) \equiv \forall\varphi_1 \vee \forall\varphi_2$$

wrong, e.g., $\forall(\diamond a \vee \diamond b) \not\equiv \forall\diamond a \vee \forall\diamond b$



$$\models \forall(\diamond a \vee \diamond b)$$

$$\not\models \forall\diamond a$$

$$\not\models \forall\diamond b$$

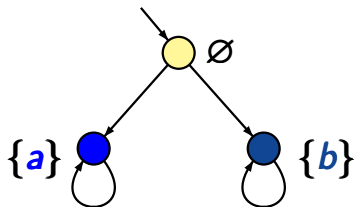
$$\exists(\varphi_1 \vee \varphi_2) \equiv \exists\varphi_1 \vee \exists\varphi_2$$

Correct or wrong?

CTLST4.6-13

$$\forall(\varphi_1 \vee \varphi_2) \equiv \forall\varphi_1 \vee \forall\varphi_2$$

wrong, e.g., $\forall(\diamond a \vee \diamond b) \not\equiv \forall\diamond a \vee \forall\diamond b$



$$\models \forall(\diamond a \vee \diamond b)$$

$$\not\models \forall\diamond a$$

$$\not\models \forall\diamond b$$

$$\exists(\varphi_1 \vee \varphi_2) \equiv \exists\varphi_1 \vee \exists\varphi_2$$

correct

Correct or wrong?

CTLST4.6-14

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

Correct or wrong?

CTLST4.6-14

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

correct.

Correct or wrong?

CTLST4.6-14

$$\exists \Diamond \exists \Box a \equiv \exists \Diamond \Box a$$

correct. $\exists \Diamond \exists \Box a \equiv \neg \forall \Box \forall \Diamond \neg a$

Correct or wrong?

CTLST4.6-14

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

correct. $\exists \diamond \exists \square a \equiv \neg \forall \square \forall \diamond \neg a$
 $\equiv \neg \forall \square \diamond \neg a$

Correct or wrong?

CTLST4.6-14

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

correct. $\exists \diamond \exists \square a \equiv \neg \forall \square \forall \diamond \neg a$
 $\equiv \neg \forall \square \diamond \neg a$
 $\equiv \exists \diamond \square a$

Correct or wrong?

CTLST4.6-14

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

correct. $\exists \diamond \exists \square a \equiv \neg \forall \square \forall \diamond \neg a$
 $\equiv \neg \forall \square \diamond \neg a$
 $\equiv \exists \diamond \square a$

$$\exists \circ \exists \diamond a \equiv \exists \circ \diamond a$$

Correct or wrong?

CTLST4.6-14

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

correct. $\exists \diamond \exists \square a \equiv \neg \forall \square \forall \diamond \neg a$
 $\equiv \neg \forall \square \diamond \neg a$
 $\equiv \exists \diamond \square a$

$$\exists \circ \exists \diamond a \equiv \exists \circ \diamond a$$

correct.

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

correct. $\exists \diamond \exists \square a \equiv \neg \forall \square \forall \diamond \neg a$
 $\equiv \neg \forall \square \diamond \neg a$
 $\equiv \exists \diamond \square a$

$$\exists \circ \exists \diamond a \equiv \exists \circ \diamond a$$

correct. Both formulas assert that an **a**-state is reachable from the current state within one or more steps.

we already saw:

$$\forall \square \forall \diamond a \equiv \forall \square \diamond a$$

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

we already saw:

$$\forall \square \forall \diamond a \equiv \forall \square \diamond a$$

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

does $\exists \square \exists \diamond a \equiv \exists \square \diamond a$ hold ?

we already saw:

$$\forall \square \forall \diamond a \equiv \forall \square \diamond a$$

$$\exists \diamond \exists \square a \equiv \exists \diamond \square a$$

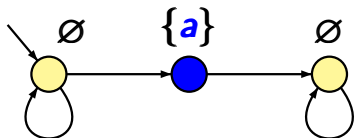
does $\exists \square \exists \diamond a \equiv \exists \square \diamond a$ hold ?

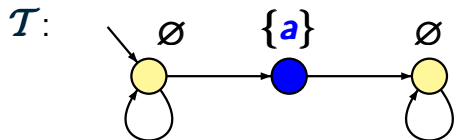
answer: **no**

$\exists x \exists y \diamond a$ and $\exists x \diamond a$ are not equivalent

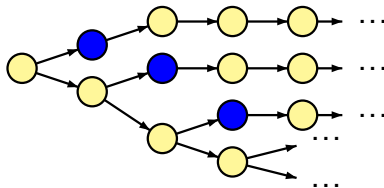
CTLST4.6-16

\mathcal{T} :



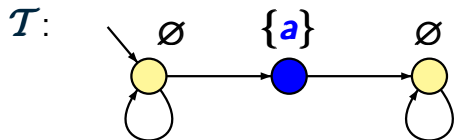


computation tree:



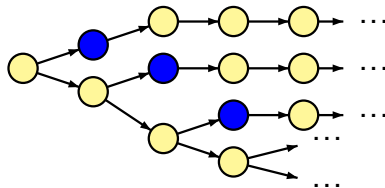
$\exists \square \exists \diamond a$ and $\exists \square \diamond a$ are not equivalent

CTLST4.6-16



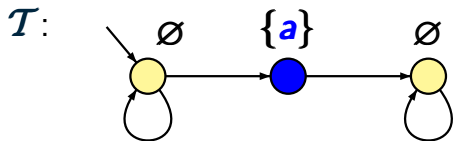
$\mathcal{T} \not\models \exists \square \diamond a$

computation tree:



$\exists \square \exists \diamond a$ and $\exists \square \diamond a$ are not equivalent

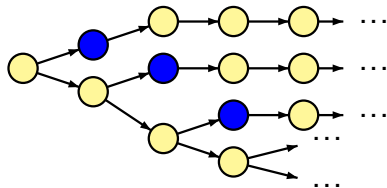
CTLST4.6-16

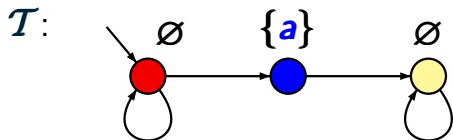


$$\mathcal{T} \not\models \exists \square \diamond a$$

$$\mathcal{T} \models \exists \square \exists \diamond a$$

computation tree:

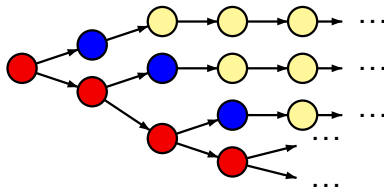


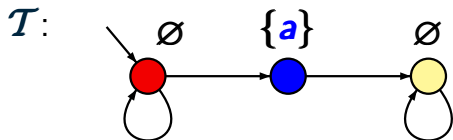


$$\mathcal{T} \not\models \exists\Box\Diamond a$$

$$\mathcal{T} \models \exists\Box\exists\Diamond a \quad \text{note: } \text{Sat}(\exists\Diamond a) = \{ \text{red}, \text{blue} \}$$

computation tree:



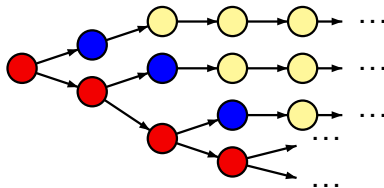


$\mathcal{T} \not\models \exists \square \diamond a$

$\mathcal{T} \models \exists \square \exists \diamond a$ note: $Sat(\exists \diamond a) = \{ \text{red}, \text{blue} \}$

hence: $\text{red red red} \dots \models \square \exists \diamond a$

computation tree:



$$\neg \exists \varphi \equiv \forall \neg \varphi$$

$$\neg \forall \varphi \equiv \exists \neg \varphi$$

$$\neg\exists\varphi \equiv \forall\neg\varphi$$

e.g., $\neg\exists\Box\Diamond a \equiv \forall\Diamond\Box\neg a$

$$\neg\forall\varphi \equiv \exists\neg\varphi$$

e.g., $\neg\forall\Box\Diamond a \equiv \exists\Diamond\Box\neg a$

$$\neg\exists\varphi \equiv \forall\neg\varphi \quad \text{e.g., } \neg\exists\Box\Diamond a \equiv \forall\Diamond\Box\neg a$$

$$\neg\forall\varphi \equiv \exists\neg\varphi \quad \text{e.g., } \neg\forall\Box\Diamond a \equiv \exists\Diamond\Box\neg a$$

$$\forall(\varphi_1 \wedge \varphi_2) \equiv \forall\varphi_1 \wedge \forall\varphi_2$$

$$\exists(\varphi_1 \vee \varphi_2) \equiv \exists\varphi_1 \vee \exists\varphi_2$$

$$\neg\exists\varphi \equiv \forall\neg\varphi \quad \text{e.g., } \neg\exists\Box\Diamond a \equiv \forall\Diamond\Box\neg a$$

$$\neg\forall\varphi \equiv \exists\neg\varphi \quad \text{e.g., } \neg\forall\Box\Diamond a \equiv \exists\Diamond\Box\neg a$$

$$\forall(\varphi_1 \wedge \varphi_2) \equiv \forall\varphi_1 \wedge \forall\varphi_2$$

$$\exists(\varphi_1 \vee \varphi_2) \equiv \exists\varphi_1 \vee \exists\varphi_2$$

$$\text{but: } \forall(\varphi_1 \vee \varphi_2) \not\equiv \forall\varphi_1 \vee \forall\varphi_2$$

$$\exists(\varphi_1 \wedge \varphi_2) \not\equiv \exists\varphi_1 \wedge \exists\varphi_2$$

$$\neg\exists\varphi \equiv \forall\neg\varphi \quad \text{e.g., } \neg\exists\Box\Diamond a \equiv \forall\Diamond\Box\neg a$$

$$\neg\forall\varphi \equiv \exists\neg\varphi \quad \text{e.g., } \neg\forall\Box\Diamond a \equiv \exists\Diamond\Box\neg a$$

$$\forall(\varphi_1 \wedge \varphi_2) \equiv \forall\varphi_1 \wedge \forall\varphi_2$$

$$\exists(\varphi_1 \vee \varphi_2) \equiv \exists\varphi_1 \vee \exists\varphi_2$$

$$\text{but: } \forall(\varphi_1 \vee \varphi_2) \not\equiv \forall\varphi_1 \vee \forall\varphi_2$$

$$\exists(\varphi_1 \wedge \varphi_2) \not\equiv \exists\varphi_1 \wedge \exists\varphi_2$$

$$\forall\Box\Diamond\varphi \equiv \forall\Box\forall\Diamond\varphi$$

$$\exists\Diamond\Box\varphi \equiv \exists\Diamond\exists\Box\varphi$$

Equivalence of CTL*-formulas

CTLST4.6-15

$$\neg\exists\varphi \equiv \forall\neg\varphi \quad \text{e.g., } \neg\exists\Box\Diamond a \equiv \forall\Diamond\Box\neg a$$

$$\neg\forall\varphi \equiv \exists\neg\varphi \quad \text{e.g., } \neg\forall\Box\Diamond a \equiv \exists\Diamond\Box\neg a$$

$$\forall(\varphi_1 \wedge \varphi_2) \equiv \forall\varphi_1 \wedge \forall\varphi_2$$

$$\exists(\varphi_1 \vee \varphi_2) \equiv \exists\varphi_1 \vee \exists\varphi_2$$

$$\text{but: } \forall(\varphi_1 \vee \varphi_2) \not\equiv \forall\varphi_1 \vee \forall\varphi_2$$

$$\exists(\varphi_1 \wedge \varphi_2) \not\equiv \exists\varphi_1 \wedge \exists\varphi_2$$

$$\forall\Box\Diamond\varphi \equiv \forall\Box\forall\Diamond\varphi \quad \text{but: } \forall\Diamond\Box\varphi \not\equiv \forall\Diamond\forall\Box\varphi$$

$$\exists\Diamond\Box\varphi \equiv \exists\Diamond\exists\Box\varphi \quad \exists\Box\Diamond\varphi \not\equiv \exists\Box\exists\Diamond\varphi$$

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

CTL* formula ϕ

question: does $\mathcal{T} \models \phi$ hold ?

given: finite TS $\mathcal{T} = (\mathcal{S}, Act, \rightarrow, \mathcal{S}_0, AP, L)$

CTL* formula ϕ

question: does $\mathcal{T} \models \phi$ hold ?

main procedure as for **CTL**:

FOR ALL subformulas ψ of ϕ DO

 compute $Sat(\psi) = \{s \in \mathcal{S} : s \models \psi\}$

OD

given: finite TS $\mathcal{T} = (\mathcal{S}, \text{Act}, \rightarrow, \mathcal{S}_0, \text{AP}, L)$

CTL* formula ϕ

question: does $\mathcal{T} \models \phi$ hold ?

main procedure as for **CTL**:

FOR ALL subformulas ψ of ϕ DO

 compute $\text{Sat}(\psi) = \{s \in \mathcal{S} : s \models \psi\}$

OD

IF $\mathcal{S}_0 \subseteq \text{Sat}(\phi)$

 THEN return “yes”

 ELSE return “no”

FI

$$\text{Sat}(\text{true}) = S$$

$$\text{Sat}(a) = \{s \in S : a \in L(s)\}$$

$$\text{Sat}(\Phi_1 \wedge \Phi_2) = \text{Sat}(\Phi_1) \cap \text{Sat}(\Phi_2)$$

$$\text{Sat}(\neg\Phi) = S \setminus \text{Sat}(\Phi)$$

$$\left. \begin{aligned} \text{Sat}(\text{true}) &= S \\ \text{Sat}(a) &= \{s \in S : a \in L(s)\} \\ \text{Sat}(\Phi_1 \wedge \Phi_2) &= \text{Sat}(\Phi_1) \cap \text{Sat}(\Phi_2) \\ \text{Sat}(\neg \Phi) &= S \setminus \text{Sat}(\Phi) \end{aligned} \right\} \text{as for CTL}$$

$$\left. \begin{aligned} \text{Sat}(\text{true}) &= S \\ \text{Sat}(a) &= \{s \in S : a \in L(s)\} \\ \text{Sat}(\Phi_1 \wedge \Phi_2) &= \text{Sat}(\Phi_1) \cap \text{Sat}(\Phi_2) \\ \text{Sat}(\neg \Phi) &= S \setminus \text{Sat}(\Phi) \end{aligned} \right\} \text{as for CTL}$$

$$\left. \begin{aligned} \text{Sat}(\forall \varphi) &= \text{Sat}_{\text{LTL}}(\varphi) \end{aligned} \right\} \text{using an LTL model checker}$$

$$\begin{array}{l}
 \text{Sat}(\text{true}) = S \\
 \text{Sat}(a) = \{s \in S : a \in L(s)\} \\
 \text{Sat}(\phi_1 \wedge \phi_2) = \text{Sat}(\phi_1) \cap \text{Sat}(\phi_2) \\
 \text{Sat}(\neg\phi) = S \setminus \text{Sat}(\phi)
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{Sat}(\text{true}) \\ \text{Sat}(a) \\ \text{Sat}(\phi_1 \wedge \phi_2) \\ \text{Sat}(\neg\phi) \end{array}} \right\} \text{as for CTL}$$

$$\begin{array}{l}
 \text{Sat}(\forall\varphi) = \text{Sat}_{LTL}(\varphi) \\
 \text{Sat}(\exists\varphi) = S \setminus \text{Sat}_{LTL}(\neg\varphi)
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{Sat}(\forall\varphi) \\ \text{Sat}(\exists\varphi) \end{array}} \right\} \text{using an LTL model checker}$$

$$\phi = \exists \diamond \square a \wedge \exists \square (\bigcirc b \wedge \diamond \neg \exists (a \text{ U } b))$$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \diamond \underbrace{\neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \diamond \underbrace{\neg \exists (a \cup b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists \square (\bigcirc b \wedge \diamond a_2)$$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \underbrace{\exists \square (\bigcirc b \wedge \diamond a_2)}_{\text{LTL formula } \varphi}$$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \underbrace{\exists \square (\bigcirc b \wedge \diamond a_2)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists \varphi$$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \underbrace{\exists \square (\bigcirc b \wedge \diamond a_2)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists \varphi$$

3. use an **LTL** model checker to compute $Sat(\exists \varphi)$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \underbrace{\exists \square (\bigcirc b \wedge \diamond a_2)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists \varphi$$

3. use an **LTL** model checker to compute $Sat(\exists \varphi)$

more precisely: existential **LTL** model checker

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \underbrace{\exists \square (\bigcirc b \wedge \diamond a_2)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists \varphi$$

3. use an **LTL** model checker to compute $Sat(\exists \varphi)$

more precisely: existential **LTL** model checker

1. construct an **NBA** for φ
2. check via nested DFS whether $\mathcal{T} \otimes \mathcal{A} \models \exists \square \diamond F$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \underbrace{\exists \square (\bigcirc b \wedge \diamond a_2)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists \varphi$$

3. compute $Sat(\exists \varphi)$ via NBA \mathcal{A} for φ and nested DFS in $\mathcal{T} \otimes \mathcal{A}$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \underbrace{\exists \square (\bigcirc b \wedge \diamond a_2)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists \varphi$$

3. compute $Sat(\exists \varphi)$ via NBA \mathcal{A} for φ and nested DFS in $\mathcal{T} \otimes \mathcal{A}$
4. return $Sat(\Phi) = Sat(a_1 \wedge \exists \varphi)$

$$\Phi = \underbrace{\exists \diamond \square a}_{\Phi_1} \wedge \exists \square (\bigcirc b \wedge \underbrace{\diamond \neg \exists (a U b)}_{\Phi_2})$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i , $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \underbrace{\exists \square (\bigcirc b \wedge \diamond a_2)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists \varphi$$

3. compute $Sat(\exists \varphi)$ via NBA \mathcal{A} for φ and nested DFS in $\mathcal{T} \otimes \mathcal{A}$
4. return $Sat(\Phi) = Sat(a_1 \wedge \exists \varphi) = Sat(\Phi_1) \cap Sat(\exists \varphi)$

Correct or wrong?

CTLST4.6-22

Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

Correct or wrong?

CTLST4.6-22

Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

CTL with fairness

CTL* semantic

Correct or wrong?

CTLST4.6-22

Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

CTL* path formula

Correct or wrong?

CTLST4.6-22

Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

CTL* path formula

correct.

Correct or wrong?

CTLST4.6-22

Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

correct.

$$s \models_{fair} \forall \square a \quad \text{iff} \quad s \models \forall (fair \wedge \square a)$$

Correct or wrong?

CTLST4.6-22

Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

correct.

$$s \models_{fair} \forall \square a \quad \text{iff} \quad s \models \forall (fair \wedge \square a)$$

wrong.

Correct or wrong?

CTLST4.6-22

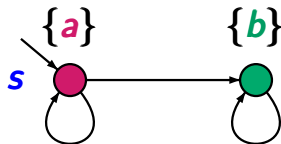
Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

correct.

$$s \models_{fair} \forall \square a \quad \text{iff} \quad s \models \forall (fair \wedge \square a)$$

wrong.



$$fair = \square \diamond \neg b$$

Correct or wrong?

CTLST4.6-22

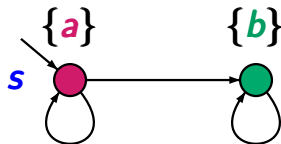
Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

correct.

$$s \models_{fair} \forall \square a \quad \text{iff} \quad s \models \forall (fair \wedge \square a)$$

wrong.



$$fair = \square \diamond \neg b$$

$$s \models_{fair} \forall \square a$$

Correct or wrong?

CTLST4.6-22

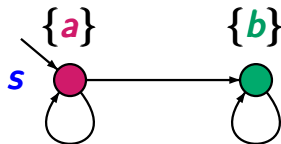
Let $fair = \bigwedge_{1 \leq i \leq k} \square \diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \square a \quad \text{iff} \quad s \models \exists (fair \wedge \square a)$$

correct.

$$s \models_{fair} \forall \square a \quad \text{iff} \quad s \models \forall (fair \wedge \square a)$$

wrong.



$$fair = \square \diamond \neg b$$

$$s \models_{fair} \forall \square a$$

$$s \not\models \forall (fair \wedge \square a)$$

Correct or wrong?

Let $fair = \bigwedge_{1 \leq i \leq k} \Box \Diamond c_i$ be an unconditional
LTL fairness assumption

$$s \models_{fair} \exists \Box a \quad \text{iff} \quad s \models \exists (fair \wedge \Box a)$$

correct.

$$s \models_{fair} \forall \Box a \quad \text{iff} \quad s \models \forall (fair \wedge \Box a)$$

wrong. But we have:

$$s \models_{fair} \forall \Box a \quad \text{iff} \quad s \models \forall (fair \rightarrow \Box a)$$

CTL* fairness assumptions are conjunctions of CTL* path formulas of the type

$\Box\Diamond\Phi$ unconditional fairness

$\Box\Diamond\Psi \rightarrow \Box\Diamond\Phi$ strong fairness

$\Diamond\Box\Psi \rightarrow \Box\Diamond\Phi$ weak fairness

CTL* fairness assumptions are **conjunctions** of **CTL*** path formulas of the type

$\Box\Diamond\Phi$ unconditional fairness

$\Box\Diamond\Psi \rightarrow \Box\Diamond\Phi$ strong fairness

$\Diamond\Box\Psi \rightarrow \Box\Diamond\Phi$ weak fairness

where Φ and Ψ are **CTL*** state formulas

CTL* fairness assumptions are **conjunctions** of **CTL*** path formulas of the type

$\Box\Diamond\Phi$ unconditional fairness

$\Box\Diamond\Psi \rightarrow \Box\Diamond\Phi$ strong fairness

$\Diamond\Box\Psi \rightarrow \Box\Diamond\Phi$ weak fairness

where Φ and Ψ are **CTL*** state formulas

obvious definition of the satisfaction relation \models_{fair}

$s \models_{fair} \exists \varphi$ iff there exists $\pi \in Paths(s)$
with $\pi \models_{fair}$ and $\pi \models_{fair} \varphi$

\models standard **CTL*** satisfaction relation

$s \models_{fair} \exists \varphi$ iff there exists $\pi \in Paths(s)$
with $\pi \models_{fair}$ and $\pi \models_{fair} \varphi$

$s \models_{fair} \forall \varphi$ iff for all $\pi \in Paths(s)$:
if $\pi \models_{fair}$ then $\pi \models_{fair} \varphi$

\models standard CTL* satisfaction relation

$s \models_{fair} \exists \varphi$ iff there exists $\pi \in Paths(s)$
with $\pi \models_{fair}$ and $\pi \models_{fair} \varphi$
iff $s \models \exists(fair \wedge \varphi)$

$s \models_{fair} \forall \varphi$ iff for all $\pi \in Paths(s)$:
if $\pi \models_{fair}$ then $\pi \models_{fair} \varphi$

\models standard CTL* satisfaction relation

$s \models_{fair} \exists \varphi$ iff there exists $\pi \in Paths(s)$
 with $\pi \models_{fair}$ and $\pi \models_{fair} \varphi$
 iff $s \models \exists (fair \wedge \varphi)$ \leftarrow if φ is quantifier-free

$s \models_{fair} \forall \varphi$ iff for all $\pi \in Paths(s)$:
 if $\pi \models_{fair}$ then $\pi \models_{fair} \varphi$

\models standard CTL* satisfaction relation

$s \models_{fair} \exists \varphi$ iff there exists $\pi \in Paths(s)$
with $\pi \models_{fair}$ and $\pi \models_{fair} \varphi$

iff $s \models \exists(fair \wedge \varphi)$ \leftarrow if φ is quantifier-free

$s \models_{fair} \forall \varphi$ iff for all $\pi \in Paths(s)$:
if $\pi \models_{fair}$ then $\pi \models_{fair} \varphi$

iff $s \models \forall(fair \rightarrow \varphi)$ \leftarrow if φ is quantifier-free

\models standard CTL* satisfaction relation

	CTL	LTL	
		<i>PSPACE-</i> complete	
\models	$size(\mathcal{T}) \cdot \Phi $	$size(\mathcal{T}) \cdot \exp(\varphi)$	

	CTL	LTL	
	<i>P</i> TIME- complete	<i>P</i> SPACE- complete	
\models	$size(\mathcal{T}) \cdot \Phi $	$size(\mathcal{T}) \cdot \exp(\varphi)$	

	CTL	LTL
	<i>P</i> TIME-complete	<i>P</i> SPACE-complete
\models	$size(\mathcal{T}) \cdot \Phi $	$size(\mathcal{T}) \cdot \exp(\varphi)$
\models_{fair}	$size(\mathcal{T}) \cdot \Phi \cdot fair $	$size(\mathcal{T}) \cdot \exp(\varphi) \cdot fair $

Complexity of CTL/LTL/CTL* model checking CTLST4.6-26

	CTL	LTL	CTL*
	<i>P</i> TIME-complete	<i>P</i> SPACE-complete	?
\models	$size(\mathcal{T}) \cdot \Phi $	$size(\mathcal{T}) \cdot \exp(\varphi)$?
\models_{fair}	$size(\mathcal{T}) \cdot \Phi \cdot fair $	$size(\mathcal{T}) \cdot \exp(\varphi) \cdot fair $?

Complexity of CTL/LTL/CTL* model checking

CTLST4.6-26

	CTL	LTL and CTL*
	<i>P</i> TIME-complete	<i>P</i> SPACE-complete
\models	$\mathcal{O}(\text{size}(\mathcal{T}) \cdot \Phi)$	$\mathcal{O}(\text{size}(\mathcal{T}) \cdot \exp(\varphi))$
\models_{fair}	$\mathcal{O}(\text{size}(\mathcal{T}) \cdot \Phi \cdot \text{fair})$	$\mathcal{O}(\text{size}(\mathcal{T}) \cdot \exp(\varphi) \cdot \text{fair})$

Complexity of CTL/LTL/CTL* model checking

CTLST4.6-26

	CTL	LTL and CTL*
	<i>PTIME</i> -complete	<i>PSPACE</i> -complete
\models	$\mathcal{O}(\text{size}(\mathcal{T}) \cdot \Phi)$	$\mathcal{O}(\text{size}(\mathcal{T}) \cdot \exp(\varphi))$
\models_{fair}	$\mathcal{O}(\text{size}(\mathcal{T}) \cdot \Phi \cdot \text{fair})$	$\mathcal{O}(\text{size}(\mathcal{T}) \cdot \exp(\varphi) \cdot \text{fair})$

model complexity, i.e., for fixed formula:
 $\mathcal{O}(\text{size}(\mathcal{T}))$

correct or wrong?

CTLST4.6-17

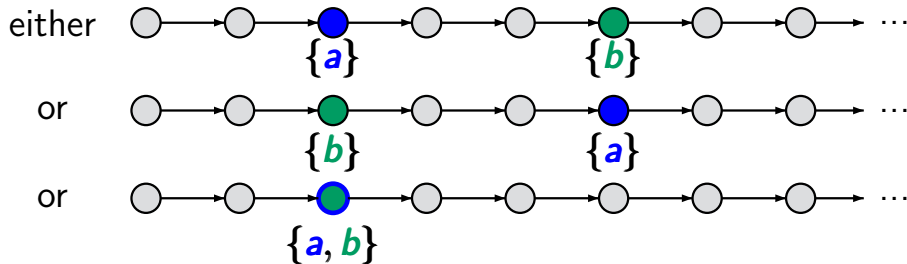
$$\exists(\diamond a \wedge \diamond b) \equiv \exists \diamond(a \wedge \exists \diamond b) \vee \exists \diamond(b \wedge \exists \diamond a)$$

$$\exists(\diamond a \wedge \diamond b) \equiv \exists\diamond(a \wedge \exists\diamond b) \vee \exists\diamond(b \wedge \exists\diamond a)$$

correct.

$$\exists(\diamond a \wedge \diamond b) \equiv \exists \diamond(a \wedge \exists \diamond b) \vee \exists \diamond(b \wedge \exists \diamond a)$$

correct.

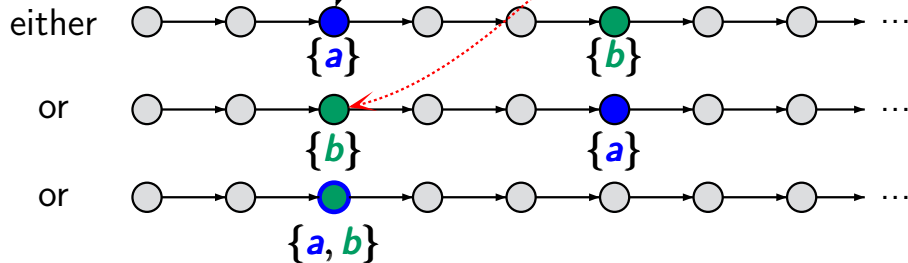


correct or wrong?

CTLST4.6-17

$$\exists(\diamond a \wedge \diamond b) \equiv \exists \diamond (a \wedge \exists \diamond b) \vee \exists \diamond (b \wedge \exists \diamond a)$$

correct.



- CTL with Boolean operators for path formulas

- CTL with Boolean operators for path formulas
- sublogic of CTL*

- CTL with Boolean operators for path formulas
- sublogic of CTL*

CTL⁺ state formulas

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

CTL⁺ path formulas

$$\psi ::= \dots$$

- CTL with Boolean operators for path formulas
- sublogic of CTL*

CTL⁺ state formulas

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

CTL⁺ path formulas

$$\psi ::= \dots$$

universal quantification can be derived: $\forall\psi \stackrel{\text{def}}{=} \neg\exists\neg\psi$

- CTL with Boolean operators for **path formulas**
- sublogic of CTL*

CTL⁺ state formulas

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

CTL⁺ path formulas

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \text{U} \Phi_2 \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi$$

- CTL with Boolean operators for path formulas
- sublogic of CTL*

CTL⁺ state formulas

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

CTL⁺ path formulas

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \text{U} \Phi_2 \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi$$

e.g., $\exists(\diamond b \wedge \square a)$

- CTL with Boolean operators for path formulas
- sublogic of CTL*

CTL⁺ state formulas

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

CTL⁺ path formulas

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \mathbf{U} \Phi_2 \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi$$

e.g., $\exists(\diamond b \wedge \square a)$ and $\exists(\bigcirc b \rightarrow (a \mathbf{U} c))$
are CTL⁺ formulas

CTL^+ is as expressive as CTL , i.e.,

For each CTL^+ -formula there exists an equivalent CTL formula.

CTL^+ is as expressive as CTL , i.e.,

For each CTL^+ -formula there exists an equivalent CTL formula.

proof relies on a series of equivalence rules, e.g.:

CTL^+ is as expressive as CTL , i.e.,

For each CTL^+ -formula there exists an equivalent CTL formula.

proof relies on a series of equivalence rules, e.g.:

$$\exists(\neg\bigcirc\phi) \rightsquigarrow \exists\bigcirc\neg\phi$$

CTL⁺ is as expressive as CTL, i.e.,

For each CTL⁺-formula there exists an equivalent CTL formula.

proof relies on a series of equivalence rules, e.g.:

$$\exists(\neg \bigcirc \Phi) \rightsquigarrow \exists \bigcirc \neg \Phi$$

$$\exists(\neg(\Phi_1 \cup \Phi_2)) \rightsquigarrow \exists((\Phi_1 \wedge \Phi_2) \cup (\neg \Phi_1 \wedge \neg \Phi_2)) \vee \exists \square \neg \Phi_2$$

CTL⁺ is as expressive as CTL, i.e.,

For each CTL⁺-formula there exists an equivalent CTL formula.

proof relies on a series of equivalence rules, e.g.:

$$\exists(\neg \bigcirc \Phi) \rightsquigarrow \exists \bigcirc \neg \Phi$$

$$\exists(\neg(\Phi_1 \cup \Phi_2)) \rightsquigarrow \exists((\Phi_1 \wedge \Phi_2) \cup (\neg \Phi_1 \wedge \neg \Phi_2)) \vee \exists \square \neg \Phi_2$$

$$\exists((\Psi_1 \cup \Psi_2) \wedge (\Phi_1 \cup \Phi_2)) \rightsquigarrow \dots$$

$$\exists(\bigcirc \Psi \wedge (\Phi_1 \cup \Phi_2)) \rightsquigarrow \dots$$

$$\begin{aligned} \exists((a \cup b) \wedge (c \cup d)) &\equiv \exists((a \wedge c) \cup (b \wedge \exists(c \cup d))) \\ &\quad \vee \exists((c \wedge a) \cup (d \wedge \exists(a \cup b))) \end{aligned}$$

CTL⁺ formula

CTL formula

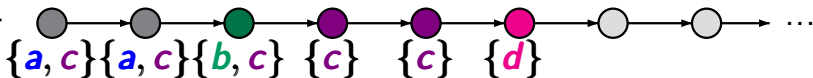
$$\exists((a \cup b) \wedge (c \cup d)) \equiv \exists((a \wedge c) \cup (b \wedge \exists(c \cup d)))$$

$$\vee \exists((c \wedge a) \cup (d \wedge \exists(a \cup b)))$$

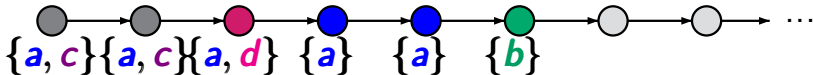
CTL⁺ formula

CTL formula

either



or



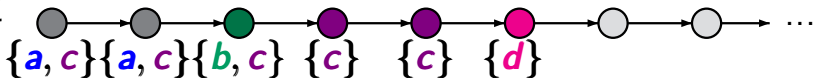
$$\exists((a \text{ U } b) \wedge (c \text{ U } d)) \equiv \exists((a \wedge c) \text{ U } (b \wedge \exists(c \text{ U } d)))$$

$$\vee \exists((c \wedge a) \text{ U } (d \wedge \exists(a \text{ U } b)))$$

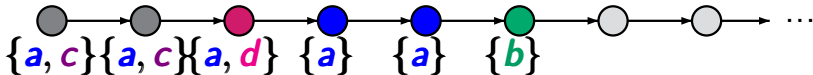
CTL⁺ formula

CTL formula

either



or



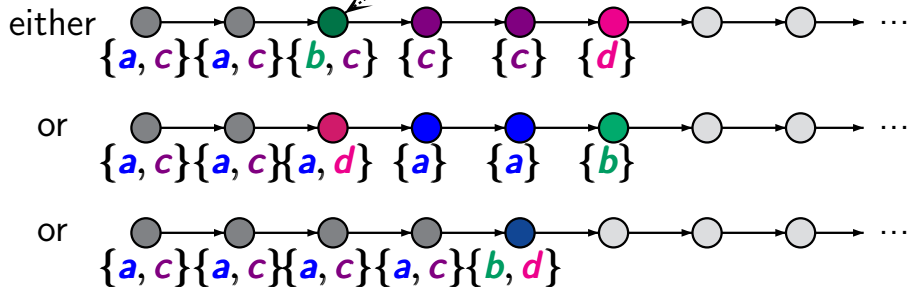
or



$$\exists((a \cup b) \wedge (c \cup d)) \equiv \exists((a \wedge c) \cup (b \wedge \exists(c \cup d))) \vee \exists((c \wedge a) \cup (d \wedge \exists(a \cup b)))$$

CTL⁺ formula

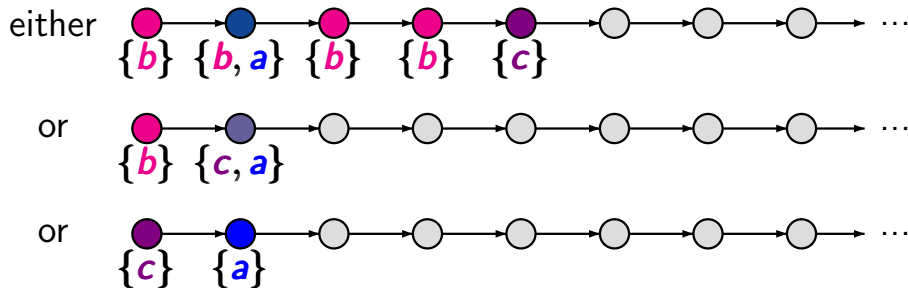
CTL formula



$$\exists(\bigcirc a \wedge (b \cup c))$$

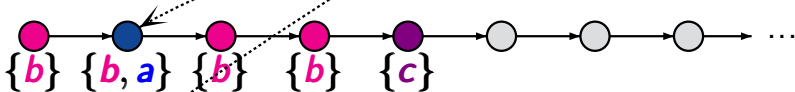
$$\begin{aligned} & \exists(\bigcirc a \wedge (b \mathbf{U} c)) \\ \equiv & (c \wedge \exists \bigcirc a) \vee (b \wedge \exists \bigcirc (a \wedge \exists (b \mathbf{U} c))) \end{aligned}$$

$$\begin{aligned} & \exists(\bigcirc a \wedge (b \cup c)) \\ \equiv & (c \wedge \exists \bigcirc a) \vee (b \wedge \exists \bigcirc (a \wedge \exists (b \cup c))) \end{aligned}$$

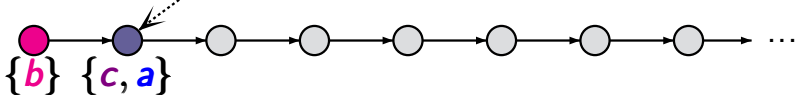


$$\begin{aligned} & \exists(\bigcirc a \wedge (b \cup c)) \\ \equiv & (c \wedge \exists \bigcirc a) \vee (b \wedge \exists \bigcirc (a \wedge \exists (b \cup c))) \end{aligned}$$

either



or



or

