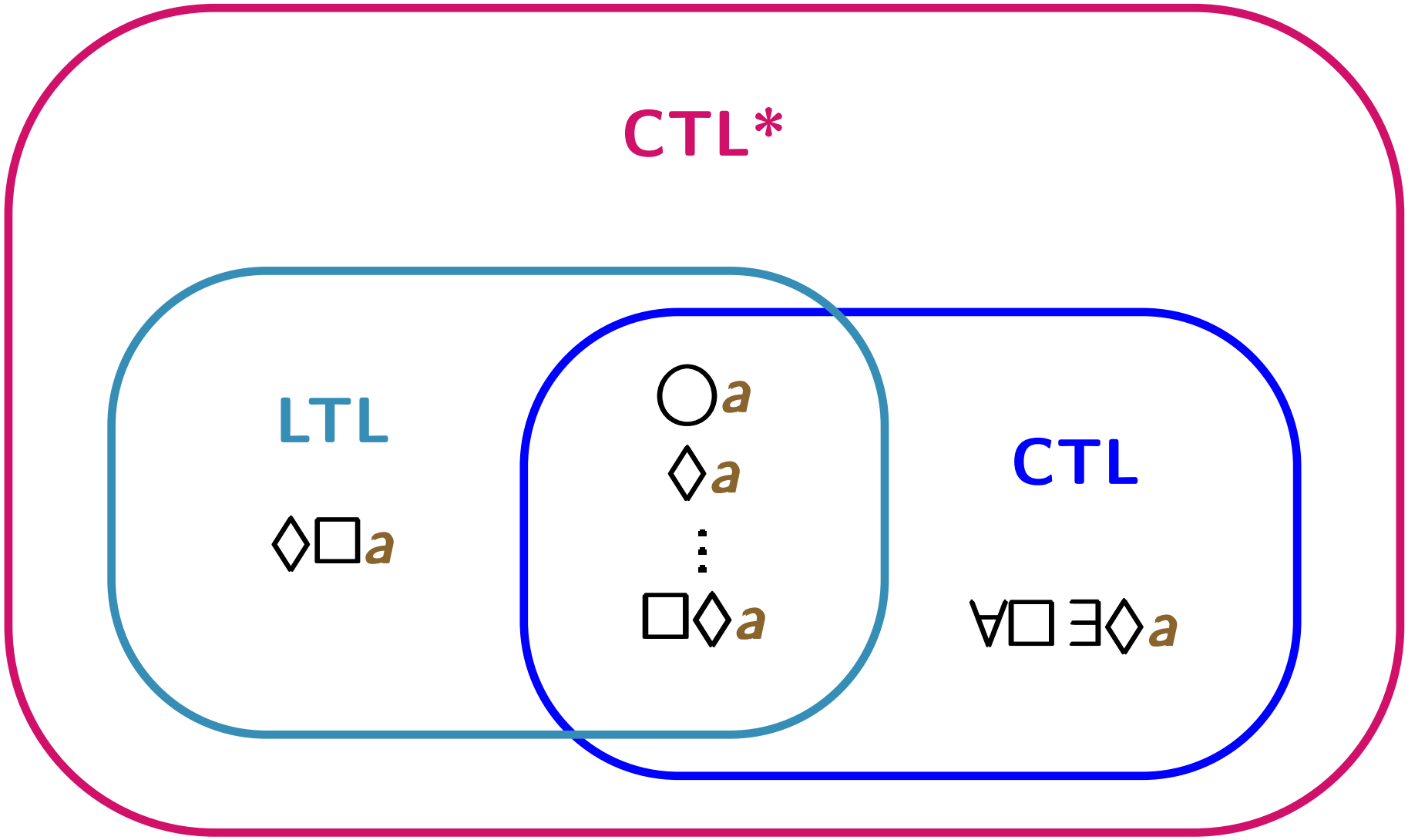
Overview

Combining LTL and CTL $\rightsquigarrow$ CTL*



Syntax of CTL*

state formulas:

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

path formulas:

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

derived operators:

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

  $$\Diamond\varphi = \mathit{true}\,\mathsf{U}\,\varphi, \quad \Box\varphi = \neg\Diamond\neg\varphi$$

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

Semantics of CTL*

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a transition system without terminal states.

Semantics of CTL*

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a transition system without terminal states.

define by structural induction:

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

Semantics of CTL* state formulas

CTLST4.6-2A

Semantics of CTL* state formulas

$s \models true$

$s \models a$ 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 $s \models \Phi_2$

$s \models \exists\varphi$ iff there exists a path $\pi \in Paths(s)$ such that $\pi \models \varphi$

Semantics of CTL* state formulas

$s \models true$

$s \models a$ 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 $s \models \Phi_2$

$s \models \exists\varphi$ iff there exists a path $\pi \in Paths(s)$ such that $\pi \models \varphi$

satisfaction relation $\models$ for **CTL*** path formulas

Semantics of CTL* path formulas

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

Semantics of CTL* path formulas

let $\pi = s_0\, s_1\, s_2 \ldots$ 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 $\mathit{suffix}(\pi, 1) \models \varphi$

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

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

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

Semantics of CTL* path formulas

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

$$
\begin{array}{lll}
\pi \models \Phi & \text{iff} & \ldots \\
\pi \models \neg\varphi & \text{iff} & \pi \not\models \varphi \\
\pi \models \varphi_1 \wedge \varphi_2 & \text{iff} & \pi \models \varphi_1 \text{ and } \pi \models \varphi_2 \\
\pi \models \bigcirc\varphi & \text{iff} & \mathit{suffix}(\pi, 1) \models \varphi \\
\pi \models \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} & \text{there exists } j \geq 0 \text{ such that} \\
 & & \quad \mathit{suffix}(\pi, j) \models \varphi_2 \\
 & & \quad \mathit{suffix}(\pi, i) \models \varphi_1 \quad \text{for } 0 \leq i < j
\end{array}
$$

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

Semantics of CTL* path formulas

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

$$
\begin{array}{lll}
\pi \models \Phi & \text{iff} & s_0 \models \Phi \\
\pi \models \neg\varphi & \text{iff} & \pi \not\models \varphi \\
\pi \models \varphi_1 \wedge \varphi_2 & \text{iff} & \pi \models \varphi_1 \text{ and } \pi \models \varphi_2 \\
\pi \models \bigcirc\varphi & \text{iff} & \mathit{suffix}(\pi, 1) \models \varphi \\
\pi \models \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} & \text{there exists } j \geq 0 \text{ such that} \\
 & & \quad \mathit{suffix}(\pi, j) \models \varphi_2 \\
 & & \quad \mathit{suffix}(\pi, i) \models \varphi_1 \quad \text{for } 0 \leq i < j
\end{array}
$$

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

Semantics of CTL* path formulas

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

satisfaction relation for **CTL*** state formulas

$\pi \models \Phi$ iff $s_0 \models \Phi$

$\pi \models \neg\varphi$ 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 $\mathit{suffix}(\pi, 1) \models \varphi$

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

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

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

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

Examples of CTL*-formulas

mutual exclusion:

safety $\forall\Box(\neg \mathit{crit}_1 \vee \neg \mathit{crit}_2)$

liveness $\forall\Box\Diamond \mathit{crit}_1 \wedge \forall\Box\Diamond \mathit{crit}_2$

progress property, e.g., $\forall\Box(\mathit{request} \rightarrow \Diamond \mathit{response})$

persistence property, e.g., $\forall\Diamond\Box a$

Examples of CTL*-formulas

mutual exclusion:

safety $\forall\Box(\neg crit_1 \vee \neg crit_2)$

liveness $\forall\Box\Diamond crit_1 \wedge \forall\Box\Diamond crit_2$

progress property, e.g., $\forall\Box(request \rightarrow \Diamond response)$

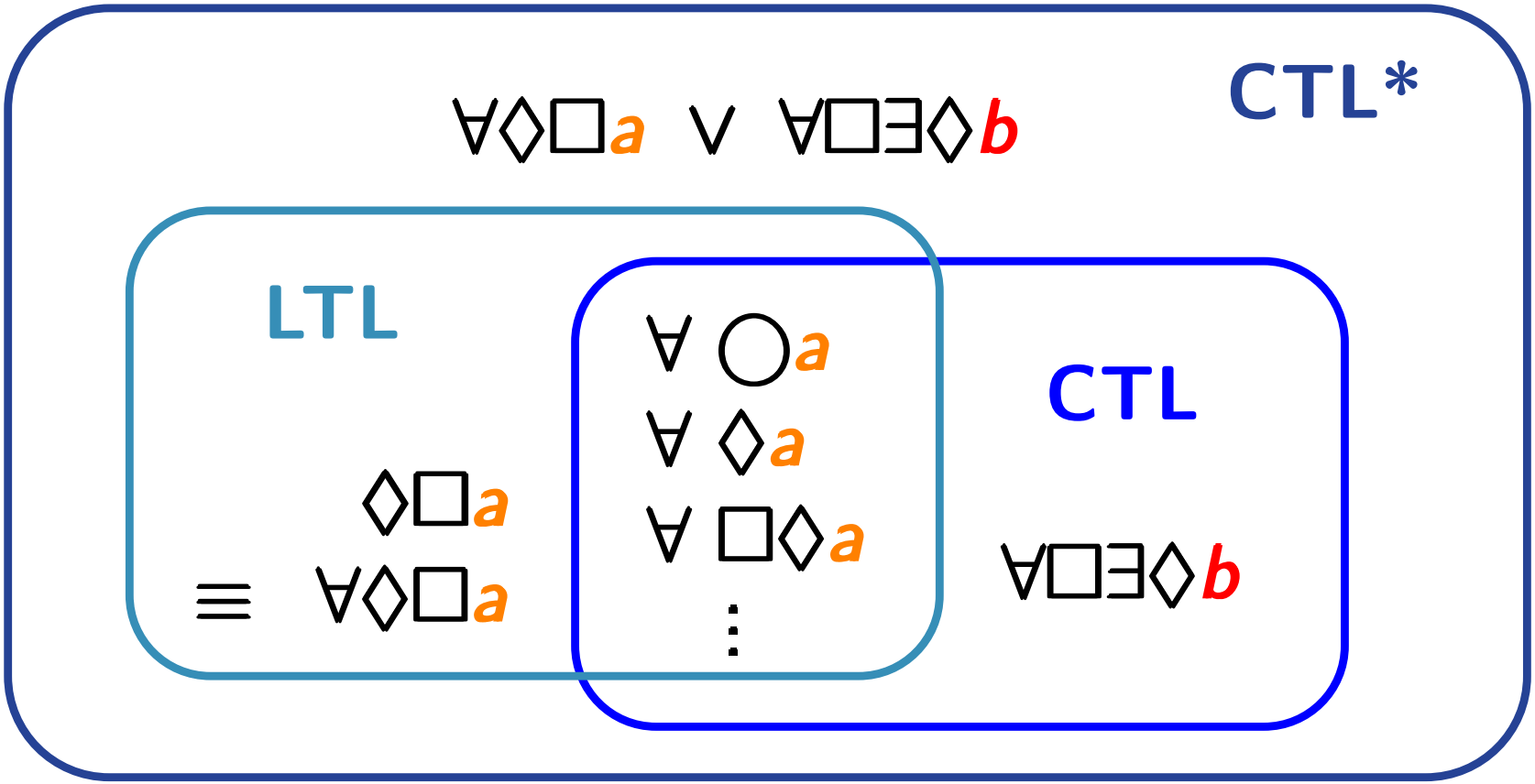
persistence property, e.g., $\forall\Diamond\Box a$

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

$$\exists\Big(\bigwedge_{v\in V}\big(\Diamond v \wedge \Box(v \rightarrow \bigcirc\Box\neg v)\big)\Big)$$

Expressiveness of CTL, LTL and CTL*

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



Equivalence of CTL*-formulas

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

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

Equivalence of CTL*-formulas

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

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

Examples:

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

$$\forall\Box\Diamond a \equiv \forall\Box\,\forall\Diamond a$$

$$\vdots$$

Equivalence of CTL*-formulas

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

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

Examples:

$$\begin{aligned}
\neg\exists\Box\Diamond a &\equiv \forall\Diamond\Box\neg a \\
\forall\Box\Diamond a &\equiv \forall\Box\,\forall\Diamond a \\
&\vdots \\
\forall\forall\varphi &\equiv \forall\varphi \\
\exists\exists\varphi &\equiv \exists\varphi
\end{aligned}$$

Equivalence of CTL*-formulas

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

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

Examples:

$$\begin{aligned}
\neg\exists\Box\Diamond a &\equiv \forall\Diamond\Box\neg a \\
\forall\Box\Diamond a &\equiv \forall\Box\,\forall\Diamond a \\
&\vdots \\
\forall\forall\varphi &\equiv \forall\varphi \\
\exists\exists\varphi &\equiv \exists\varphi \\
\forall\exists\varphi &\equiv ?
\end{aligned}$$

Equivalence of CTL*-formulas

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

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

Examples:

$$\begin{aligned}
\neg\exists\Box\Diamond a &\equiv \forall\Diamond\Box\neg a \\
\forall\Box\Diamond a &\equiv \forall\Box\,\forall\Diamond a \\
&\vdots \\
\forall\forall\varphi &\equiv \forall\varphi \\
\exists\exists\varphi &\equiv \exists\varphi \\
\forall\exists\varphi &\equiv \exists\varphi
\end{aligned}$$

Correct or wrong?

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

Correct or wrong?

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

correct.

Correct or wrong?

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

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

Correct or wrong?

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

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

$\equiv \neg\forall\Box\Diamond\neg a$

Correct or wrong?

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

correct.

$$\begin{aligned}\exists\Diamond\exists\Box a &\equiv \neg\forall\Box\forall\Diamond\neg a\\ &\equiv \neg\forall\Box\Diamond\neg a\\ &\equiv \exists\Diamond\Box a\end{aligned}$$

Correct or wrong?

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

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

$\equiv \neg\forall\Box\Diamond\neg a$

$\equiv \exists\Diamond\Box a$

$$\exists\bigcirc\exists\Diamond a \equiv \exists\bigcirc\Diamond a$$

Correct or wrong?

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

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

$\equiv \neg\forall\Box\Diamond\neg a$

$\equiv \exists\Diamond\Box a$

$$\exists\bigcirc\exists\Diamond a \equiv \exists\bigcirc\Diamond a$$

correct.

Correct or wrong?

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

correct.

$$\begin{aligned}\exists\Diamond\exists\Box a &\equiv \neg\forall\Box\forall\Diamond\neg a\\ &\equiv \neg\forall\Box\Diamond\neg a\\ &\equiv \exists\Diamond\Box a\end{aligned}$$

$$\exists\bigcirc\exists\Diamond a \equiv \exists\bigcirc\Diamond a$$

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

Combinations of □ and ◊ in CTL*

Combinations of $\Box$ and $\Diamond$ in CTL*

we already saw:

$$\forall\Box\forall\Diamond a \equiv \forall\Box\Diamond a$$

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

Combinations of $\Box$ and $\Diamond$ in CTL*

we already saw:

$$\forall\Box\forall\Diamond a \equiv \forall\Box\Diamond a$$

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

does $\exists\Box\exists\Diamond a \equiv \exists\Box\Diamond a$ hold ?

Combinations of $\Box$ and $\Diamond$ in CTL*

we already saw:

$$\forall\Box\forall\Diamond a \equiv \forall\Box\Diamond a$$

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

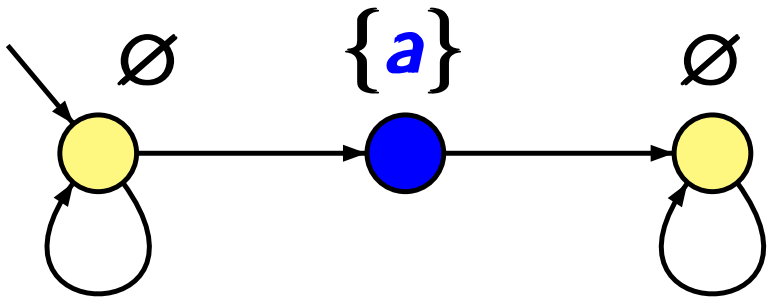
does $\exists\Box\exists\Diamond a \equiv \exists\Box\Diamond a$ hold ?

answer: **no**

$\exists\Box\exists\Diamond a$ and $\exists\Box\Diamond a$ are not equivalent

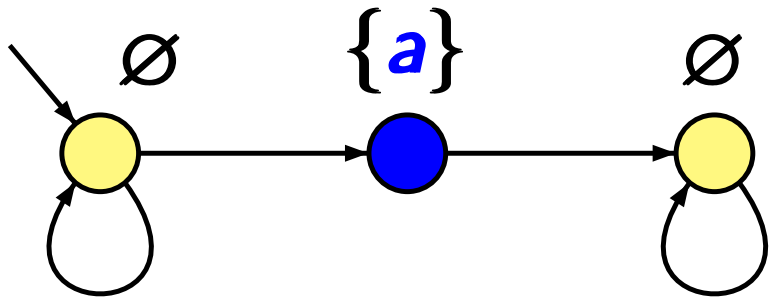
$\exists\Box\exists\Diamond a$ and $\exists\Box\Diamond a$ are not equivalent

$\mathcal{T}$:

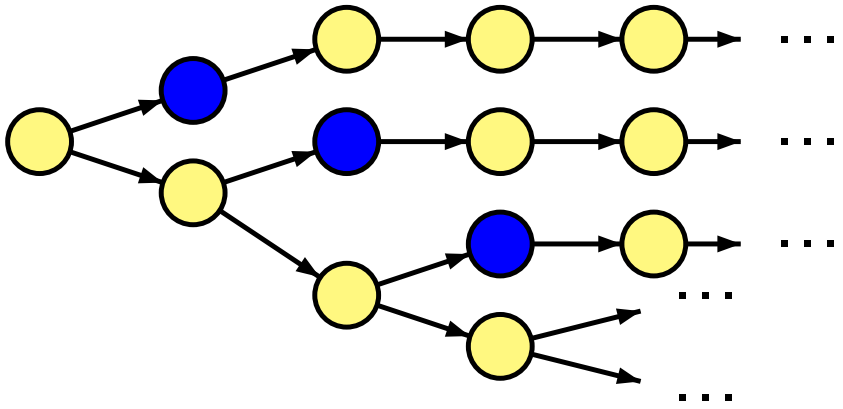


$\exists\Box\exists\Diamond a$ and $\exists\Box\Diamond a$ are not equivalent

$\mathcal{T}$:

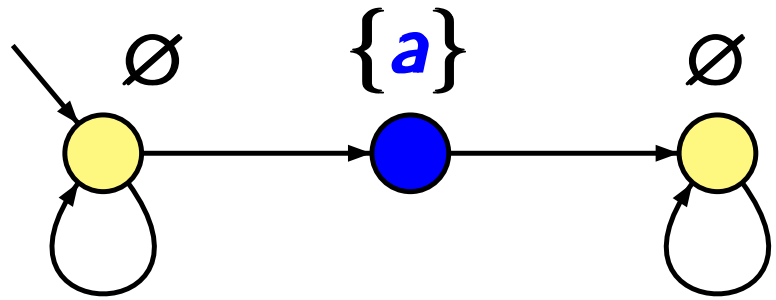


computation tree:



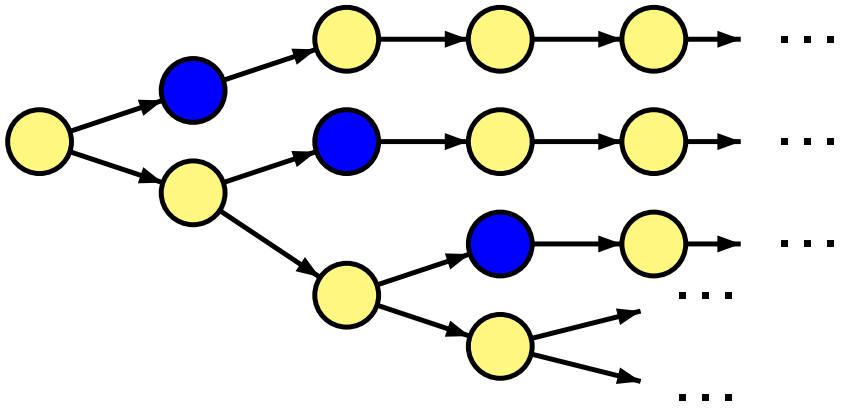
$\exists\Box\exists\Diamond a$ and $\exists\Box\Diamond a$ are not equivalent

$\mathcal{T}$:



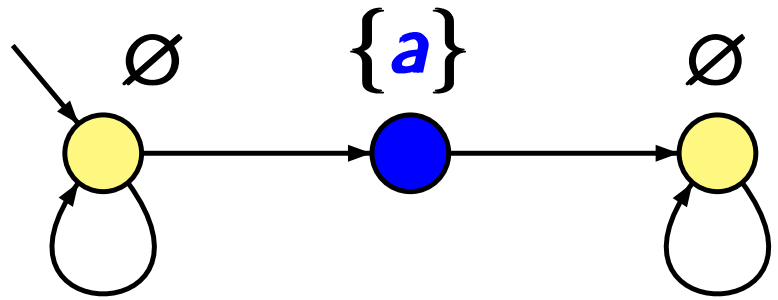
$\mathcal{T} \not\models \exists\Box\Diamond a$

computation tree:



$\exists\Box\exists\Diamond a$ and $\exists\Box\Diamond a$ are not equivalent

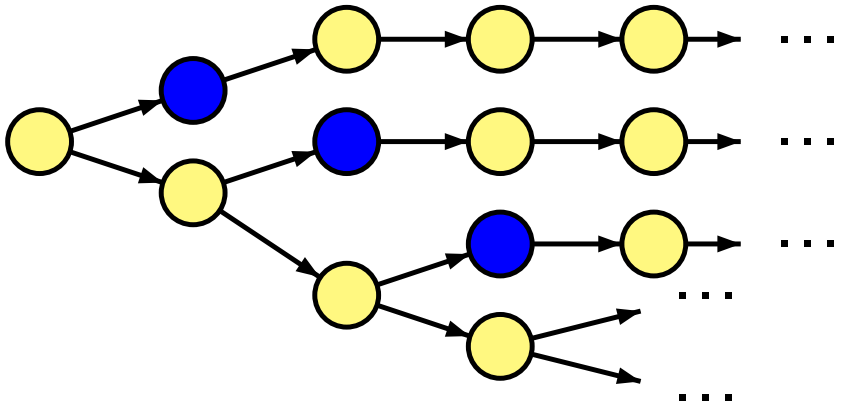
$\mathcal{T}$:



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

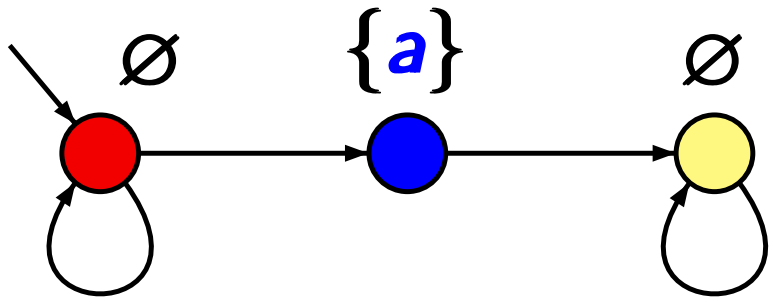
$\mathcal{T} \models \exists\Box\exists\Diamond a$

computation tree:



$\exists\Box\exists\Diamond a$ and $\exists\Box\Diamond a$ are not equivalent

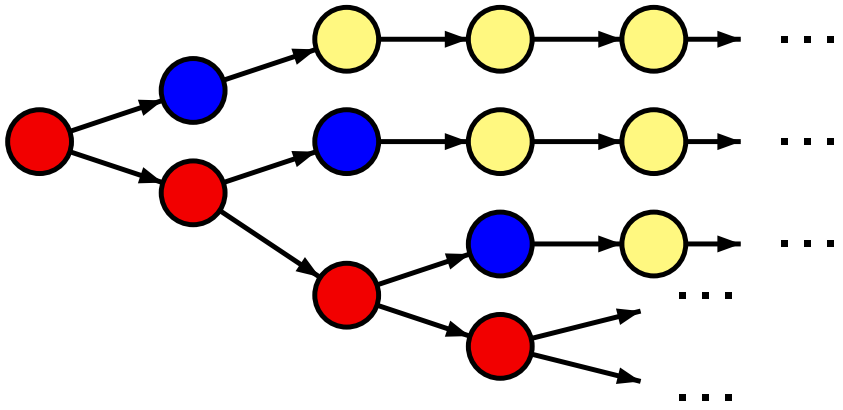
$\mathcal{T}$:



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

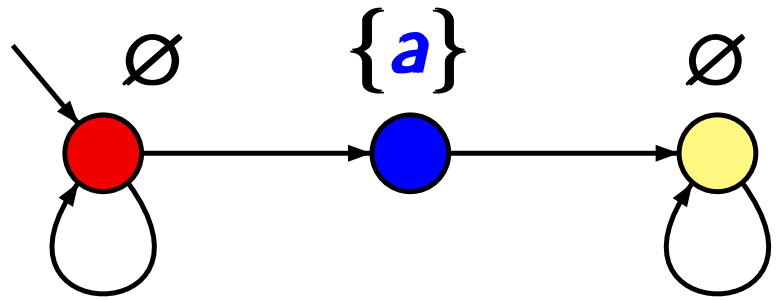
$\mathcal{T} \models \exists\Box\exists\Diamond a$ note: $Sat(\exists\Diamond a) = \{ \bullet, \bullet \}$

computation tree:



$\exists\Box\exists\Diamond a$ and $\exists\Box\Diamond a$ are not equivalent

$\mathcal{T}$:

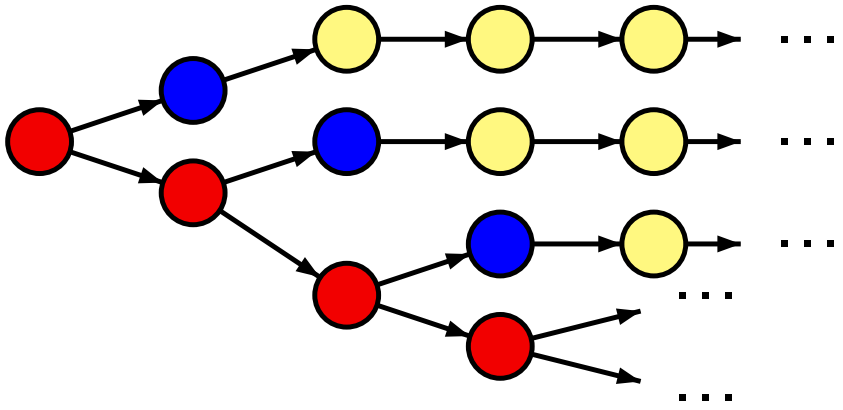


$\mathcal{T} \not\models \exists\Box\Diamond a$

$\mathcal{T} \models \exists\Box\exists\Diamond a$ note: $Sat(\exists\Diamond a) = \{ \bullet, \bullet \}$

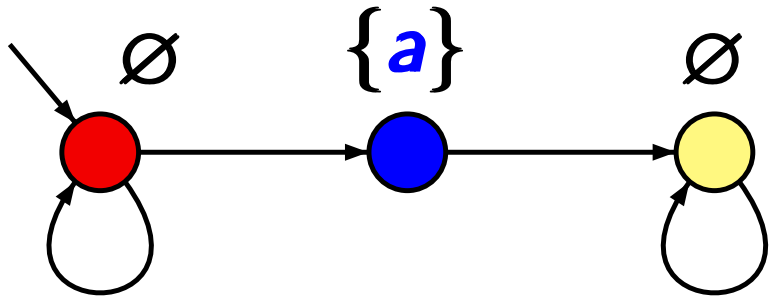
hence: $\bullet\bullet\bullet\ldots \models \Box\exists\Diamond a$

computation tree:



$\exists\Box\exists\Diamond a$ and $\exists\Box\Diamond a$ are not equivalent

$\mathcal{T}$:



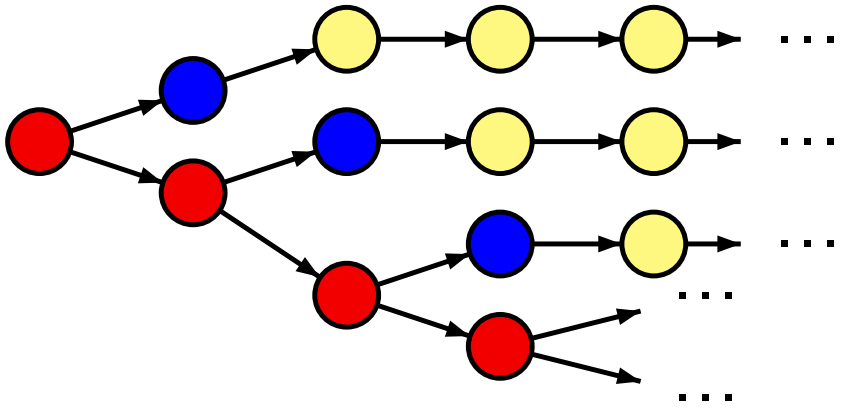
$$\exists\Box\exists\Diamond a \not\equiv \exists\Box\Diamond a$$

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

$\mathcal{T} \models \exists\Box\exists\Diamond a$ note: $Sat(\exists\Diamond a) = \{ \bullet, \bullet \}$

hence: $\bullet\bullet\bullet\ldots \models \Box\exists\Diamond a$

computation tree:



Equivalence of CTL*-formulas

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

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

Equivalence of CTL*-formulas

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

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

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

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

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

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

Equivalence of CTL*-formulas

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

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

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

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

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

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

$\forall\Box\Diamond\varphi \equiv \forall\Box\forall\Diamond\varphi$ but: $\forall\Diamond\Box\varphi \not\equiv \forall\Diamond\forall\Box\varphi$

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