Modeling and Analysis of Hybrid Systems

Propositional and temporal logics

Prof. Dr. Erika Ábrahám

Informatik 2 - Theory of Hybrid Systems
RWTH Aachen University

SS 2015

Temporal logics

Assume

- a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$,
- a set of atomic propositions *AP*, and
- a labeling function $L : \Sigma \to 2^{AP}$.

Temporal logics

Assume

- a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$,
- a set of atomic propositions *AP*, and
- a labeling function $L : \Sigma \to 2^{AP}$.

- How can we describe properties of this system?

Temporal logics

Assume

- a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$,
- a set of atomic propositions *AP*, and
- a labeling function $L : \Sigma \to 2^{AP}$.

- How can we describe properties of this system?
- We need a well-suited logic.

Propositional logic

- Abstract syntax:

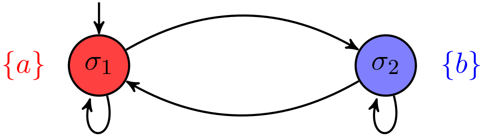
$$\varphi \;::=\; a \mid (\varphi \wedge \varphi) \mid (\neg \varphi)$$

  with $a \in AP$.

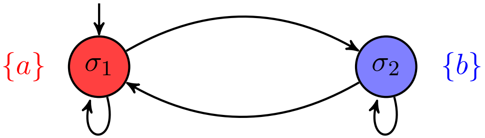
- Syntactic sugar: *true*, *false*, $\vee$, $\rightarrow$, $\leftrightarrow$, $\ldots$
- Omit parentheses when no confusion
- Semantics:

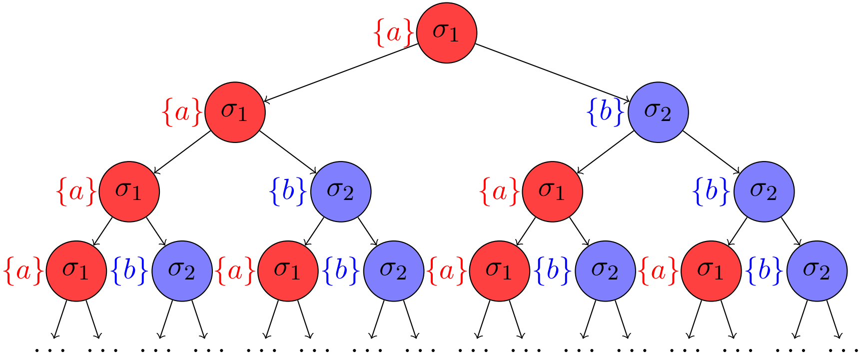
$$\begin{array}{lll} L, \sigma \models a & \textit{iff} & a \in L(\sigma), \\ L, \sigma \models (\varphi_1 \wedge \varphi_2) & \textit{iff} & \sigma \models \varphi_1 \textit{ and } \sigma \models \varphi_2, \\ L, \sigma \models (\neg \varphi) & \textit{iff} & \sigma \not\models \varphi. \end{array}$$

Computation tree



Computation tree

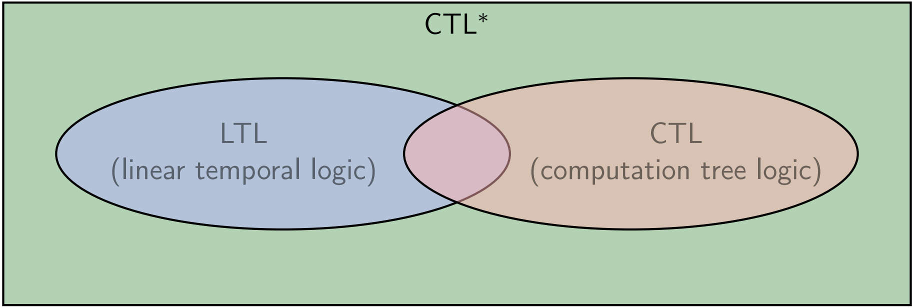




Temporal logics

In the computation tree we can describe

- a given path starting in a state (path formulas, "linear" properties) and
- quantified (universal/existential) properties over all paths starting in a given state (state formulas, "branching" properties).



Examples for path formulas

proposition a :

next $\mathcal{X} b$:

$\bigcirc b$

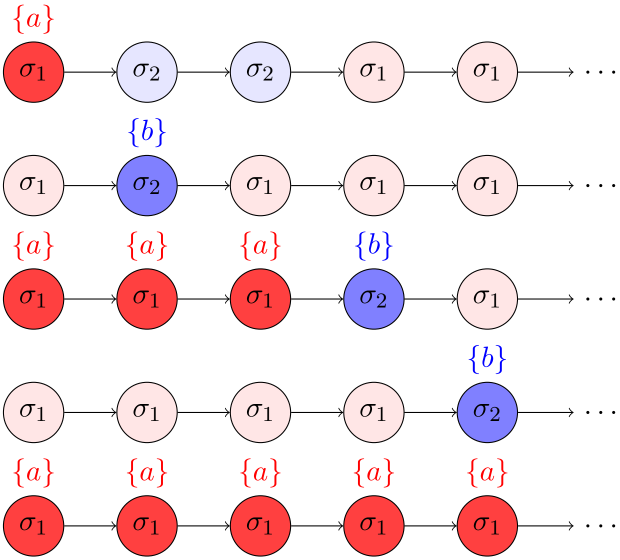
until $a \, \mathcal{U} \, b$:

finally $\mathcal{F} b$:

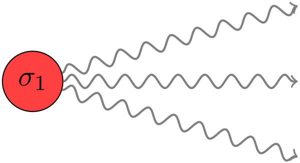
$\Diamond b$

globally $\mathcal{G} a$:

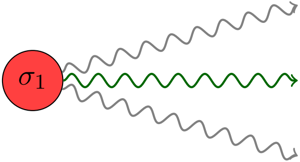
$\Box a$



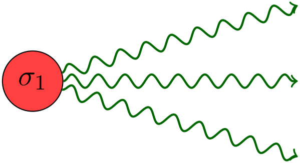
proposition a :



exists $\mathbf{E}\ \varphi^p$:



for all $\mathbf{A}\ \varphi^p$:



CTL* syntax

CTL* state formulae:

$$\varphi^s \quad ::= \quad a \mid (\varphi^s \wedge \varphi^s) \mid (\neg \varphi^s) \mid (\mathbf{E}\varphi^p)$$

with $a \in AP$ and φ^p are CTL* path formulae.

CTL* path formulae:

$$\varphi^p \quad ::= \quad \varphi^s \mid (\varphi^p \wedge \varphi^p) \mid (\neg \varphi^p) \mid (\mathcal{X}\varphi^p) \mid (\varphi^p \, \mathcal{U} \, \varphi^p)$$

where φ^s are CTL* state formulae.

$(a \, \mathcal{U} \, b) \, \mathcal{U} \, (\mathcal{X} c)$

$(a \, \mathcal{U} (A (c \, \mathcal{U} \, d)))$

CTL* syntax

CTL* state formulae:

$$\varphi^s \quad ::= \quad a \mid (\varphi^s \wedge \varphi^s) \mid (\neg \varphi^s) \mid (\mathbf{E}\varphi^p)$$

with $a \in AP$ and φ^p are CTL* path formulae.

CTL* path formulae:

$$\varphi^p \quad ::= \quad \varphi^s \mid (\varphi^p \wedge \varphi^p) \mid (\neg \varphi^p) \mid (\mathcal{X}\varphi^p) \mid (\varphi^p \, \mathcal{U} \, \varphi^p)$$

where φ^s are CTL* state formulae.

CTL* formulae are CTL* state formulae.

CTL* syntax

CTL* state formulae:

$$\varphi^s \quad ::= \quad a \mid (\varphi^s \wedge \varphi^s) \mid (\neg\varphi^s) \mid (\mathbf{E}\varphi^p)$$

with $a \in AP$ and φ^p are CTL* path formulae.

CTL* path formulae:

$$\varphi^p \quad ::= \quad \varphi^s \mid (\varphi^p \wedge \varphi^p) \mid (\neg\varphi^p) \mid (\mathcal{X}\varphi^p) \mid (\varphi^p \, \mathcal{U} \, \varphi^p)$$

where φ^s are CTL* state formulae.

CTL* formulae are CTL* state formulae.

We sometimes omit parentheses, based on the order $\mathbf{E} > \mathcal{U} > \mathcal{X} > \wedge > \neg$ from strongest to weakest binding.

CTL* syntax

CTL* state formulae:

$$\varphi^s \quad ::= \quad a \mid (\varphi^s \wedge \varphi^s) \mid (\neg\varphi^s) \mid (\mathbf{E}\varphi^p)$$

with $a \in AP$ and φ^p are CTL* path formulae.

CTL* path formulae:

$$\varphi^p \quad ::= \quad \varphi^s \mid (\varphi^p \wedge \varphi^p) \mid (\neg\varphi^p) \mid (\mathcal{X}\varphi^p) \mid (\varphi^p \,\mathcal{U}\, \varphi^p)$$

where φ^s are CTL* state formulae.

CTL* formulae are CTL* state formulae.

We sometimes omit parentheses, based on the order $\mathbf{E} > \mathcal{U} > \mathcal{X} > \wedge > \neg$ from strongest to weakest binding.

Syntactic sugar:
$\mathbf{A}\varphi^p := \neg\mathbf{E}\neg\varphi^p$ ("for all"), $\mathcal{F}\varphi^p := \mathit{true}\,\mathcal{U}\varphi^p$ ("finally" or "eventually"),
$\mathcal{G}\varphi^p := \neg\mathcal{F}\neg\varphi^p$ ("globally" or "always"), $\mathcal{R}$ ("releases"), ...

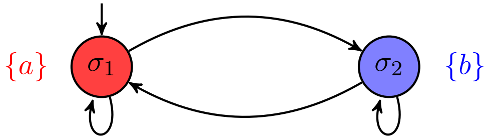
CTL* semantics

Assume $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ to be a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ along with a labeling function $L : \Sigma \to 2^{AP}$, where AP is a finite set of atomic propositions.
For a path $\pi = \sigma_0 \to \sigma_1 \to \ldots$ of $\mathcal{LSTS}$, let $\pi(i)$ denote σ_i, and let π^i denote $\sigma_i \to \sigma_{i+1} \to \ldots$.

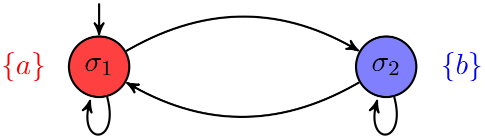
$\mathcal{L}, \sigma \models a$	*iff*	$a \in L(\sigma)$
$\mathcal{L}, \sigma \models \varphi_1^s \land \varphi_2^s$	*iff*	$\mathcal{L}, \sigma \models \varphi_1^s$ *and* $\mathcal{L}, \sigma \models \varphi_2^s$
$\mathcal{L}, \sigma \models \neg\varphi^s$	*iff*	$\mathcal{L}, \sigma \not\models \varphi^s$
$\mathcal{L}, \sigma \models \mathbf{E}\varphi^p$	*iff*	$\mathcal{L}, \pi \models \varphi^p$ *for some path* $\pi = \sigma \to \ldots$ *of* $\mathcal{LSTS}$
$\mathcal{L}, \pi \models \varphi^s$	*iff*	$\mathcal{L}, \pi(0) \models \varphi^s$
$\mathcal{L}, \pi \models \varphi_1^p \land \varphi_2^p$	*iff*	$\mathcal{L}, \pi \models \varphi_1^p$ *and* $\mathcal{L}, \pi \models \varphi_2^p$
$\mathcal{L}, \pi \models \neg\varphi^p$	*iff*	$\mathcal{L}, \pi \not\models \varphi^p$
$\mathcal{L}, \pi \models \mathcal{X}\varphi^p$	*iff*	$\mathcal{L}, \pi^1 \models \varphi^p$
$\mathcal{L}, \pi \models \varphi_1^p \,\mathcal{U}\, \varphi_2^p$	*iff*	*exists* $0 \leq j$ *with* $\mathcal{L}, \pi^j \models \varphi_2^p$ *and* $\mathcal{L}, \pi^i \models \varphi_1^p$ *for all* $0 \leq i < j$.

$\mathcal{L} \models \varphi^s$ iff $\mathcal{L}, \sigma_0 \models \varphi^s$ for all initial states σ_0 of $\mathcal{LSTS}$.

Computation tree



Computation tree



a ✓

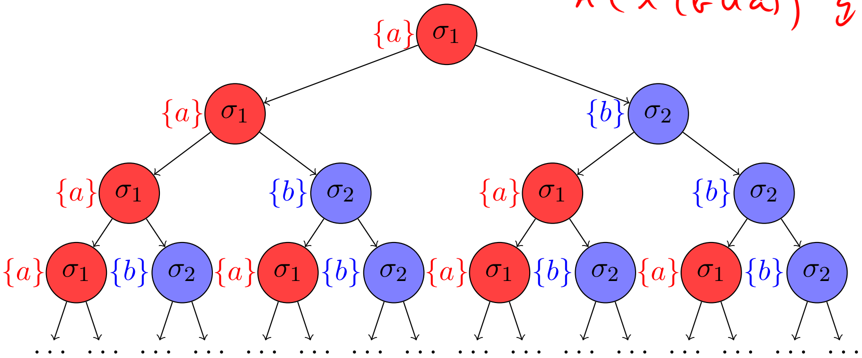
$\neg b$ ✓

$A(a \, U \, b)$ ↯

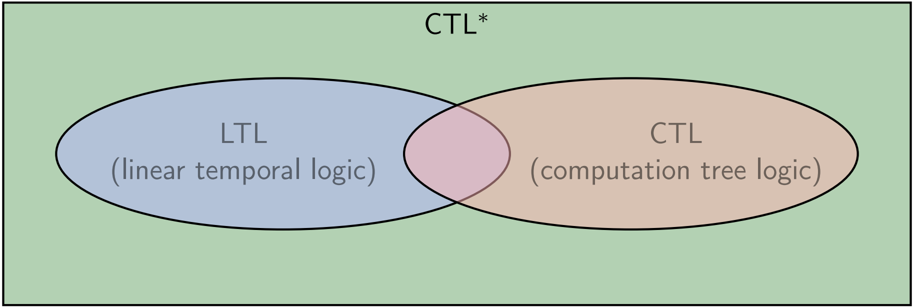
$E(G a)$ ✓

$E(X(b \, U \, a))$ ✓

$A(X(b \, U \, a))$ ↯



The relation of LTL, CTL, and CTL*



LTL syntax

Linear Temporal Logic (LTL) is suited to argue about single (linear) paths in the computation tree.

- Abstract syntax:

$$\varphi^p \quad ::= \quad a \mid (\varphi^p \land \varphi^p) \mid (\neg \varphi^p) \mid (\mathcal{X} \varphi^p) \mid (\varphi^p \; \mathcal{U} \; \varphi^p)$$

  where $a \in AP$.
- Syntactic sugar: $\mathcal{F}$ ("finally" or "eventually"), $\mathcal{G}$ ("globally"), etc.
- Again, we sometimes omit parentheses using the same binding order as for CTL*.

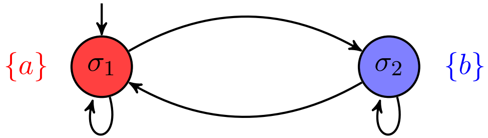
LTL semantics

Assume $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ to be a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ along with a labeling function $L : \Sigma \to 2^{AP}$, where AP is a finite set of atomic propositions.
For a path $\pi = \sigma_0 \to \sigma_1 \to \ldots$ of $\mathcal{LSTS}$, let $\pi(i)$ denote σ_i, and let π^i denote $\sigma_i \to \sigma_{i+1} \to \ldots$.

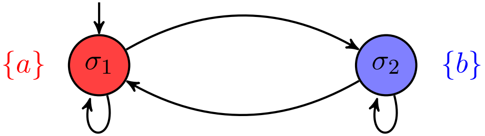
$$
\begin{array}{lll}
\mathcal{L}, \pi \models a & \textit{iff} & a \in L(\pi(0)), \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} & \mathcal{L}, \pi \models \varphi_1^p \textit{ and } \mathcal{L}, \pi \models \varphi_2^p, \\
\mathcal{L}, \pi \models \neg \varphi^p & \textit{iff} & \mathcal{L}, \pi \not\models \varphi^p, \\
\mathcal{L}, \pi \models \mathcal{X} \varphi^p & \textit{iff} & \pi^1 \models \varphi^p, \\
\mathcal{L}, \pi \models \varphi_1^p \,\mathcal{U}\, \varphi_2^p & \textit{iff} & \exists j \geq 0. \pi^j \models \varphi_2^p \wedge \forall 0 \leq i < j. \pi^i \models \varphi_1^p.
\end{array}
$$

$\mathcal{LSTS} \models \varphi^p$ iff $\pi \models \varphi^p$ for all paths π of $\mathcal{LSTS}$ starting in an initial state.

Computation tree



Computation tree

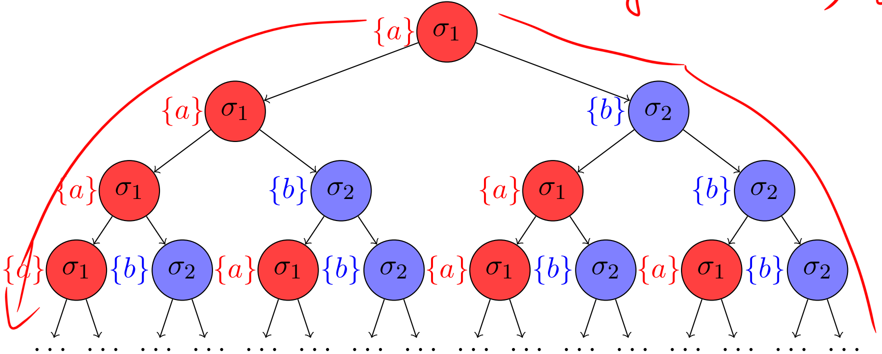


X (b U a) ↯

G (a ∨ b) ✓

a ⇒ a U (b U a) ✓

G (a → F b) ↯



CTL syntax

CTL state formulae:

$$\varphi^s \quad ::= \quad a \mid (\varphi^s \wedge \varphi^s) \mid (\neg \varphi^s) \mid (\mathbf{E}\varphi^p) \mid (\mathbf{A}\varphi^p)$$

$E\,X\,\varphi^s \mid E\,\varphi^s\,U\,\varphi^s \mid A\,X\,\varphi^s \mid A\,\varphi^s\,U\,\varphi^s$

with $a \in AP$ and φ^p are CTL path formulae.

CTL path formulae:

$$\varphi^p \quad ::= \quad \mathcal{X}\varphi^s \mid \varphi^s\,\mathcal{U}\,\varphi^s$$

where φ^s are CTL state formulae.

CTL formulae are CTL state formulae.

As before, we sometimes omit parentheses.

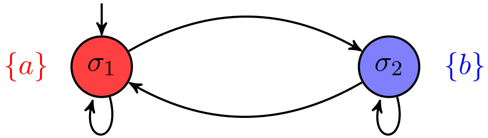
CTL semantics

Assume $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ to be a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ along with a labeling function $L : \Sigma \rightarrow 2^{AP}$, where AP is a finite set of atomic propositions.
For a path $\pi = \sigma_0 \rightarrow \sigma_1 \rightarrow \ldots$ of $\mathcal{LSTS}$, let $\pi(i)$ denote σ_i, and let π^i denote $\sigma_i \rightarrow \sigma_{i+1} \rightarrow \ldots$.

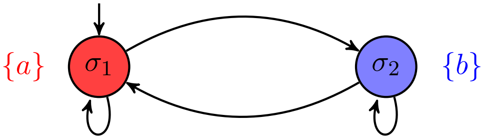
$\mathcal{L}, \sigma \models a$	*iff*	$a \in L(\sigma)$
$\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s$	*iff*	$\mathcal{L}, \sigma \models \varphi_1^s$ *and* $\mathcal{L}, \sigma \models \varphi_2^s$
$\mathcal{L}, \sigma \models \neg\varphi^s$	*iff*	$\mathcal{L}, \sigma \not\models \varphi^s$
$\mathcal{L}, \sigma \models \mathbf{E}\varphi^p$	*iff*	$\mathcal{L}, \pi \models \varphi^p$ *for some path* $\pi = \sigma \rightarrow \ldots$ *of* $\mathcal{LSTS}$
$\mathcal{L}, \sigma \models \mathbf{A}\varphi^p$	*iff*	$\mathcal{L}, \pi \models \varphi^p$ *for all* $\pi = \sigma_0 \rightarrow \sigma_1 \rightarrow \ldots$ *with* $\sigma_0 = \sigma$
$\mathcal{L}, \pi \models \mathcal{X}\varphi^s$	*iff*	$\mathcal{L}, \pi(1) \models \varphi^s$
$\mathcal{L}, \pi \models \varphi_1^s \,\mathcal{U}\, \varphi_2^s$	*iff*	*exists* $0 \leq j$ *with* $\mathcal{L}, \pi(j) \models \varphi_2^s$ *and* $\mathcal{L}, \pi(i) \models \varphi_1^s$ *for all* $0 \leq i < j$.

$\mathcal{L} \models \varphi^s$ iff $\mathcal{L}, \sigma_0 \models \varphi^s$ for all initial states σ_0 of $\mathcal{LSTS}$.

Computation tree

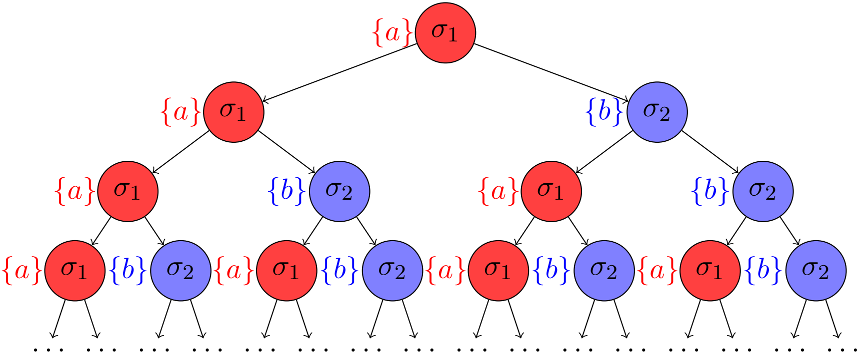


Computation tree

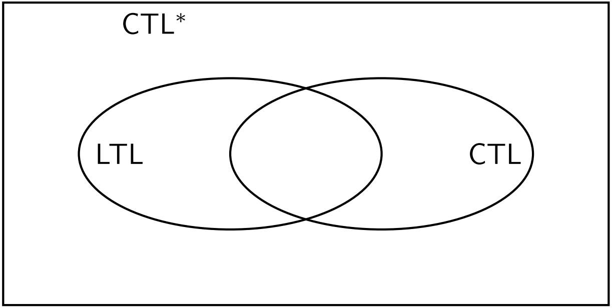


$AG\, EF\, b$

$EG\, (a \wedge EX\, b)$



The relation of LTL, CTL, and CTL*



- The LTL formula $\mathcal{F}\mathcal{G}a$ is not expressible in CTL.
- The CTL formula $\mathbf{A}\mathcal{F}\mathbf{A}\mathcal{G}a$ is not expressible in LTL.

CTL (explicit) model checking

Given a state transition system and a CTL formula ψ^s, CTL model checking labels the states recursively with the sub-formulae of ψ^s inside-out.

CTL (explicit) model checking

Given a state transition system and a CTL formula ψ^s, CTL model checking labels the states recursively with the sub-formulae of ψ^s inside-out.

- The labeling with atomic propositions $a \in AP$ is given by

CTL (explicit) model checking

Given a state transition system and a CTL formula ψ^s, CTL model checking labels the states recursively with the sub-formulae of ψ^s inside-out.

- The labeling with atomic propositions $a \in AP$ is given by a labeling function.

CTL (explicit) model checking

Given a state transition system and a CTL formula ψ^s, CTL model checking labels the states recursively with the sub-formulae of ψ^s inside-out.

- The labeling with atomic propositions $a \in AP$ is given by a labeling function.
- Given the labelings for ψ_1^s and ψ_2^s, we label a state with $\psi_1^s \wedge \psi_2^s$ iff

CTL (explicit) model checking

Given a state transition system and a CTL formula ψ^s, CTL model checking labels the states recursively with the sub-formulae of ψ^s inside-out.

- The labeling with atomic propositions $a \in \mathit{AP}$ is given by a labeling function.
- Given the labelings for ψ_1^s and ψ_2^s, we label a state with $\psi_1^s \wedge \psi_2^s$ iff the state is labeled with both ψ_1^s and ψ_2^s.

CTL (explicit) model checking

Given a state transition system and a CTL formula ψ^s, CTL model checking labels the states recursively with the sub-formulae of ψ^s inside-out.

- The labeling with atomic propositions $a \in AP$ is given by a labeling function.
- Given the labelings for ψ_1^s and ψ_2^s, we label a state with $\psi_1^s \wedge \psi_2^s$ iff the state is labeled with both ψ_1^s and ψ_2^s.
- Given the labeling for ψ^s, we label a state with $\neg\psi^s$ iff

CTL (explicit) model checking

Given a state transition system and a CTL formula ψ^s, CTL model checking labels the states recursively with the sub-formulae of ψ^s inside-out.

- The labeling with atomic propositions $a \in AP$ is given by a labeling function.
- Given the labelings for ψ_1^s and ψ_2^s, we label a state with $\psi_1^s \wedge \psi_2^s$ iff the state is labeled with both ψ_1^s and ψ_2^s.
- Given the labeling for ψ^s, we label a state with $\neg\psi^s$ iff the state is not labeled with ψ^s.

CTL (explicit) model checking

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{E}\psi_1^s \ \mathcal{U} \ \psi_2^s$, and

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{E}\psi_1^s\ \mathcal{U}\ \psi_2^s$, and
  - label all states that have the label ψ_1^s and have a successor state with the label $\mathbf{E}\psi_1^s\ \mathcal{U}\ \psi_2^s$ also with $\mathbf{E}\psi_1^s\ \mathcal{U}\ \psi_1^s$ iteratively until a fixed point is reached.

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$, and
  - label all states that have the label ψ_1^s and have a successor state with the label $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$ also with $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_1^s$ iteratively until a fixed point is reached.
- Given the labeling for ψ^s, we label a state with $\mathbf{A}\mathcal{X}\psi^s$ iff

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$, and
  - label all states that have the label ψ_1^s and have a successor state with the label $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$ also with $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_1^s$ iteratively until a fixed point is reached.
- Given the labeling for ψ^s, we label a state with $\mathbf{A}\mathcal{X}\psi^s$ iff all successor states are labeled with ψ^s.

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$, and
  - label all states that have the label ψ_1^s and have a successor state with the label $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$ also with $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_1^s$ iteratively until a fixed point is reached.
- Given the labeling for ψ^s, we label a state with $\mathbf{A}\mathcal{X}\psi^s$ iff all successor states are labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$, and
  - label all states that have the label ψ_1^s and have a successor state with the label $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$ also with $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_1^s$ iteratively until a fixed point is reached.
- Given the labeling for ψ^s, we label a state with $\mathbf{A}\mathcal{X}\psi^s$ iff all successor states are labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{A}\psi_1^s \,\mathcal{U}\, \psi_2^s$, and

CTL (explicit) model checking

- Given the labeling for ψ^s, we label a state with $\mathbf{E}\mathcal{X}\psi^s$ iff there is a successor state labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{E}\psi_1^s\,\mathcal{U}\,\psi_2^s$, and
  - label all states that have the label ψ_1^s and have a successor state with the label $\mathbf{E}\psi_1^s\,\mathcal{U}\,\psi_2^s$ also with $\mathbf{E}\psi_1^s\,\mathcal{U}\,\psi_1^s$ iteratively until a fixed point is reached.
- Given the labeling for ψ^s, we label a state with $\mathbf{A}\mathcal{X}\psi^s$ iff all successor states are labeled with ψ^s.
- Given the labeling for ψ_1^s and ψ_2^s, we
  - label all with ψ_2^s labeled states additionally with $\mathbf{A}\psi_1^s\,\mathcal{U}\,\psi_2^s$, and
  - label all states that have the label ψ_1^s and all of their successor states have the label $\mathbf{A}\psi_1^s\,\mathcal{U}\,\psi_2^s$ also with $\mathbf{A}\psi_1^s\,\mathcal{U}\,\psi_2^s$ iteratively until a fixed point is reached.

Discrete-time LTL

$\mathcal{X}^k \varphi^p =$

$$\begin{cases} \varphi^p & \textit{if } k = 0 \\ \mathcal{X}\mathcal{X}^{k-1}\varphi^p & \textit{else.} \end{cases}$$

$\varphi_1^p \, \mathcal{U}^{[k_1,k_2]} \, \varphi_2^p =$

$$\begin{cases} \varphi_1^p \, \mathcal{U} \, \varphi_2^p & \textit{for } [k_1,k_2] = [0,\infty] \\ \varphi_2^p & \textit{for } [k_1,k_2] = [0,0] \\ \varphi_1^p \wedge \mathcal{X}(\varphi_1^p \, \mathcal{U}^{[k_1-1,k_2-1]} \, \varphi_2^p) & \textit{for } k_1 > 0 \\ \varphi_2^p \vee (\varphi_1^p \wedge \mathcal{X}(\varphi_1^p \, \mathcal{U}^{[0,k_2-1]} \, \varphi_2^p)) & \textit{for } k_1 = 0, k_2 > 0 \end{cases}$$

Discrete-time CTL

$\mathbf{E}\mathcal{X}^k\psi^s =$

$$\begin{cases} \psi^s & \textit{if } k=0 \\ \mathbf{E}\mathcal{X}\mathbf{E}\mathcal{X}^{k-1}\psi^s & \textit{else.} \end{cases}$$

$\mathbf{E}\psi_1^s\ \mathcal{U}^{[k_1,k_2]}\ \psi_2^s =$

$$\begin{cases} \mathbf{E}\psi_1^s\ \mathcal{U}\ \psi_2^s & \textit{for } [k_1,k_2]=[0,\infty] \\ \psi_2^s & \textit{for } [k_1,k_2]=[0,0] \\ \psi_1^s \wedge \mathbf{E}\mathcal{X}\mathbf{E}(\psi_1^s\ \mathcal{U}^{[k_1-1,k_2-1]}\ \psi_2^s) & \textit{for } k_1>0 \\ \psi_2^s \vee (\psi_1^s \wedge \mathbf{E}\mathcal{X}\mathbf{E}(\psi_1^s\ \mathcal{U}^{[0,k_2-1]}\ \psi_2^s)) & \textit{for } k_1=0, k_2>0 \end{cases}$$

Syntactic sugar

We also write

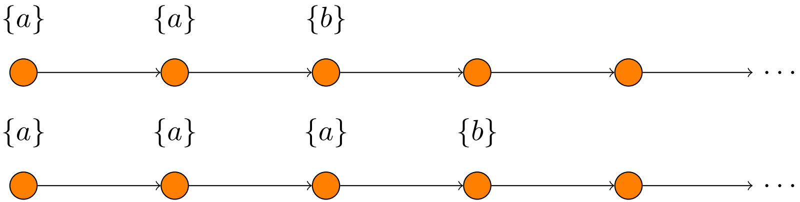
- $\mathcal{U}^{\leq k}$ instead of $\mathcal{U}^{[0,k]}$,
- $\mathcal{U}^{\geq k}$ for $\mathcal{U}^{[k,\infty]}$,
- $\mathcal{U}^{=k}$ for $\mathcal{U}^{[k,k]}$, and
- $\mathcal{U}$ for $\mathcal{U}^{[0,\infty]}$.

Example

The discrete-time LTL formula $a\,\mathcal{U}^{[2,3]}\,b$ is defined as

$$a \wedge \mathcal{X}(a \wedge \mathcal{X}(b \vee (a \wedge \mathcal{X} b))).$$

It is satisfied by paths of the following form:



Discrete-time model checking

As the discrete-time temporal operators are defined as syntactic sugar, LTL model checking can be applied to check the validity of discrete-time LTL formulae for state transition systems.