Modeling and Analysis of Hybrid Systems

1. Preliminaries

Prof. Dr. Erika Ábrahám

Informatik 2 - LuFG Theory of Hybrid Systems
RWTH Aachen University

Szeged, Hungary, 27 September - 06 October 2017

Organizational

- email: abraham@cs.rwth-aachen.de
- 7 x 90 minutes lectures, 7 x 90 minutes exercises
- agree on time and place
- learning materials
- written exam mid of November 2017

Contents

1 Hybrid systems

2 Labeled state transition systems

3 Labeled transition systems

4 Temporal logics

5 CTL model checking

“Hybrid”

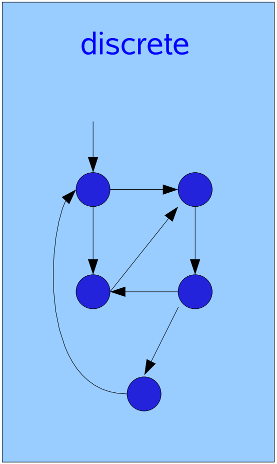
Wikipedia:

> “A hybrid is the combination of two or more different things, aimed at achieving a particular objective or goal.”

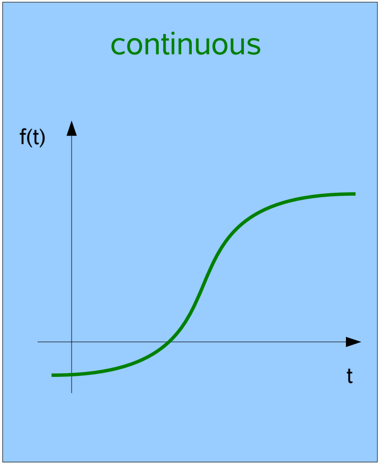
A hybrid rose

A hybrid car

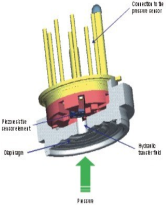
Hybrid in computer science

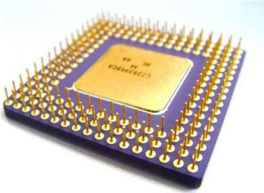


+



The discrete part


Connection to the pressure sensor
Piezoresistive sensorelement
Diaphragm
Hydraulic transfer fluid
Pressure



Combined with the continuous part

Example: Bouncing ball

Ball falls from a given height, bounces at the ground, raises, falls again...

- vertical position of the ball x_1
- velocity x_2

Example: Bouncing ball

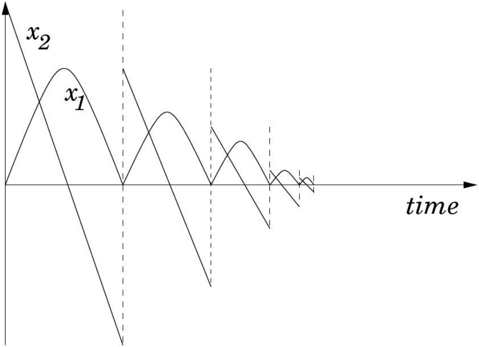
Ball falls from a given height, bounces at the ground, raises, falls again...

- vertical position of the ball x_1
- velocity x_2
- continuous changes of position between bounces
- discrete changes at bounce time

Example: Bouncing ball

Ball falls from a given height, bounces at the ground, raises, falls again...

- vertical position of the ball x_1
- velocity x_2
- continuous changes of position between bounces
- discrete changes at bounce time

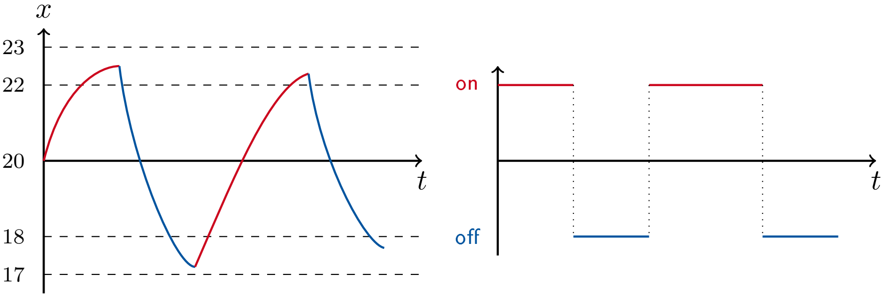


Example: Thermostat

- Temperature x is controlled by switching a heater on and off
- x is regulated by a thermostat:
  - $17^\circ \leq x \leq 18^\circ \rightsquigarrow$ "heater on"
  - $22^\circ \leq x \leq 23^\circ \rightsquigarrow$ "heater off"

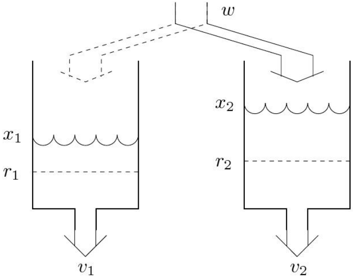
Example: Thermostat

- Temperature x is controlled by switching a heater on and off
- x is regulated by a thermostat:
  - $17° \leq x \leq 18°$ $\rightsquigarrow$ “heater on”
  - $22° \leq x \leq 23°$ $\rightsquigarrow$ “heater off”



Example: Water tank system

- two constantly leaking tanks v_1 and v_2
- hose w refills exactly one tank at one point in time
- w can switch between tanks instantaneously



There are much more complex examples of hybrid systems, like e.g.

- automobiles, trains, etc.
- automated highway systems
- collision-avoidance and free flight for aircrafts
- digitally controlled chemical plants
- biological cell growth and division ...

In this course we learn how to model and analyse hybrid systems, considering a sequence of modeling languages with increasing expressive power.

- labeled state transition systems
- labeled transition systems
- timed automata
- initialized rectangular automata
- linear hybrid automata I
- linear hybrid automata II

Contents

1 Hybrid systems

2 Labeled state transition systems

3 Labeled transition systems

4 Temporal logics

5 CTL model checking

Labeled state transition systems

Definition

Labeled state transition systems

Definition

A labeled state transition system (LSTS) is a tuple $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ with

- a (possibly infinite) state set Σ,
- a label set Lab (for synchronisation, we do not use it in this course),
- a transition relation $Edge \subseteq \Sigma \times Lab \times \Sigma$ and
- a non-empty set of initial states $Init \subseteq \Sigma$.

Labeled state transition systems

Definition

A labeled state transition system (LSTS) is a tuple $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ with

- a (possibly infinite) state set Σ,
- a label set Lab (for synchronisation, we do not use it in this course),
- a transition relation $Edge \subseteq \Sigma \times Lab \times \Sigma$ and
- a non-empty set of initial states $Init \subseteq \Sigma$.

Operational semantics:

Labeled state transition systems

Definition

A labeled state transition system (LSTS) is a tuple $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ with

- a (possibly infinite) state set Σ,
- a label set Lab (for synchronisation, we do not use it in this course),
- a transition relation $Edge \subseteq \Sigma \times Lab \times \Sigma$ and
- a non-empty set of initial states $Init \subseteq \Sigma$.

Operational semantics:

$$\frac{(\sigma, a, \sigma') \in Edge}{\sigma \xrightarrow{a} \sigma'}$$

- Path:

Labeled state transition systems

Definition

A labeled state transition system (LSTS) is a tuple $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ with

- a (possibly infinite) state set Σ,
- a label set Lab (for synchronisation, we do not use it in this course),
- a transition relation $Edge \subseteq \Sigma \times Lab \times \Sigma$ and
- a non-empty set of initial states $Init \subseteq \Sigma$.

Operational semantics:

$$\frac{(\sigma, a, \sigma') \in Edge}{\sigma \overset{a}{\rightarrow} \sigma'}$$

- Path: $\sigma_0 \overset{a_0}{\rightarrow} \sigma_1 \overset{a_1}{\rightarrow} \sigma_2 \ldots$.

Labeled state transition systems

Definition

A labeled state transition system (LSTS) is a tuple $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ with

- a (possibly infinite) state set Σ,
- a label set Lab (for synchronisation, we do not use it in this course),
- a transition relation $Edge \subseteq \Sigma \times Lab \times \Sigma$ and
- a non-empty set of initial states $Init \subseteq \Sigma$.

Operational semantics:

$$\frac{(\sigma, a, \sigma') \in Edge}{\sigma \overset{a}{\rightarrow} \sigma'}$$

- Path: $\sigma_0 \overset{a_0}{\rightarrow} \sigma_1 \overset{a_1}{\rightarrow} \sigma_2 \ldots$.
- Initial path:

Labeled state transition systems

Definition

A labeled state transition system (LSTS) is a tuple $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ with

- a (possibly infinite) state set Σ,
- a label set Lab (for synchronisation, we do not use it in this course),
- a transition relation $Edge \subseteq \Sigma \times Lab \times \Sigma$ and
- a non-empty set of initial states $Init \subseteq \Sigma$.

Operational semantics:

$$\frac{(\sigma, a, \sigma') \in Edge}{\sigma \xrightarrow{a} \sigma'}$$

- Path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$.
- Initial path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$ with $\sigma_0 \in Init$.

Labeled state transition systems

Definition

A labeled state transition system (LSTS) is a tuple $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ with

- a (possibly infinite) state set Σ,
- a label set Lab (for synchronisation, we do not use it in this course),
- a transition relation $Edge \subseteq \Sigma \times Lab \times \Sigma$ and
- a non-empty set of initial states $Init \subseteq \Sigma$.

Operational semantics:

$$\frac{(\sigma, a, \sigma') \in Edge}{\sigma \xrightarrow{a} \sigma'}$$

- Path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$.
- Initial path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$ with $\sigma_0 \in Init$.
- A state is called reachable iff

Labeled state transition systems

Definition

A labeled state transition system (LSTS) is a tuple $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ with

- a (possibly infinite) state set Σ,
- a label set Lab (for synchronisation, we do not use it in this course),
- a transition relation $Edge \subseteq \Sigma \times Lab \times \Sigma$ and
- a non-empty set of initial states $Init \subseteq \Sigma$.

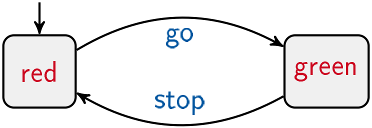
Operational semantics:

$$\frac{(\sigma, a, \sigma') \in Edge}{\sigma \xrightarrow{a} \sigma'}$$

- Path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$.
- Initial path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$ with $\sigma_0 \in Init$.
- A state is called reachable iff there is an initial path leading to it.

Pedestrian light

Pedestrian light



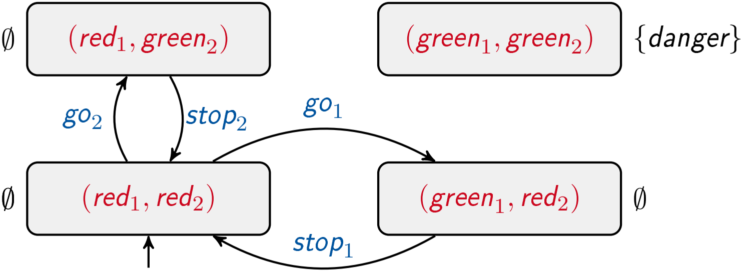
Labeling

To be able to formalize properties of LSTSs, it is common to define

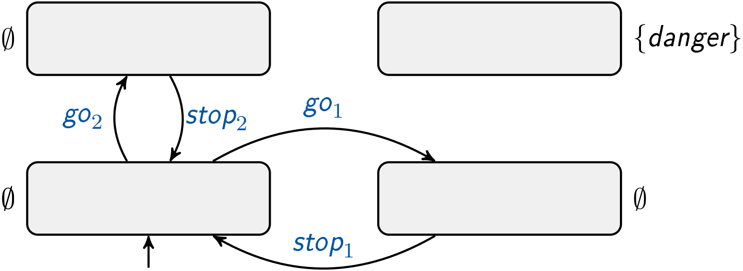
- a set of atomic propositions AP and
- a state labeling function $L : \Sigma \rightarrow 2^{AP}$ assigning a set of atomic propositions to each state.

The set $L(\sigma)$ consists of all propositions that are defined to hold in σ. These propositional labels on states should not be mixed up with the synchronization labels on edges.

Two traffic lights



Two traffic lights



Contents

1 Hybrid systems

2 Labeled state transition systems

3 Labeled transition systems

4 Temporal logics

5 CTL model checking

Labeled transition systems

Labeled transition systems

Definition

A labeled transition system (LTS) is a tuple $\mathcal{LTS} = (Loc, Var, Lab, Edge, Init)$ with

- finite set of locations Loc,
- finite set of (typed) variables Var,
- finite set of synchronization labels Lab, $\tau \in Lab$ (stutter label)
- finite set of edges $Edge \subseteq Loc \times Lab \times 2^{V^2} \times Loc$ (including stutter transitions (l, τ, μ_τ, l) for each location $l \in Loc$),
- initial states $Init \subseteq \Sigma$.

with

- valuations $\nu : Var \to Domain$, V is the set of valuations
- state $\sigma = (l, \nu) \in Loc \times V$, Σ is the set of states

Semantics of LTS

Operational semantics has a single rule:

Semantics of LTS

Operational semantics has a single rule:

$$\frac{(l, a, \mu, l') \in \mathit{Edge} \quad (\nu, \nu') \in \mu}{(l, \nu) \xrightarrow{a} (l', \nu')}$$

Semantics of LTS

Operational semantics has a single rule:

$$\frac{(l, a, \mu, l') \in Edge \quad (\nu, \nu') \in \mu}{(l, \nu) \xrightarrow{a} (l', \nu')}$$

- Path:

Semantics of LTS

Operational semantics has a single rule:

$$\frac{(l, a, \mu, l') \in Edge \quad (\nu, \nu') \in \mu}{(l, \nu) \xrightarrow{a} (l', \nu')}$$

- Path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$.

Semantics of LTS

Operational semantics has a single rule:

$$\frac{(l, a, \mu, l') \in \mathit{Edge} \quad (\nu, \nu') \in \mu}{(l, \nu) \xrightarrow{a} (l', \nu')}$$

- Path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$.
- Initial path:

Semantics of LTS

Operational semantics has a single rule:

$$\frac{(l, a, \mu, l') \in \mathit{Edge} \quad (\nu, \nu') \in \mu}{(l, \nu) \xrightarrow{a} (l', \nu')}$$

- Path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$.
- Initial path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$ with $\sigma_0 \in \mathit{Init}$.

Semantics of LTS

Operational semantics has a single rule:

$$\frac{(l, a, \mu, l') \in Edge \quad (\nu, \nu') \in \mu}{(l, \nu) \xrightarrow{a} (l', \nu')}$$

- Path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$.
- Initial path: $\sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \sigma_2 \ldots$ with $\sigma_0 \in Init$.
- A state is called reachable iff

Semantics of LTS

Operational semantics has a single rule:

$$\frac{(l,a,\mu,l') \in \mathit{Edge} \quad (\nu,\nu') \in \mu}{(l,\nu) \overset{a}{\to} (l',\nu')}$$

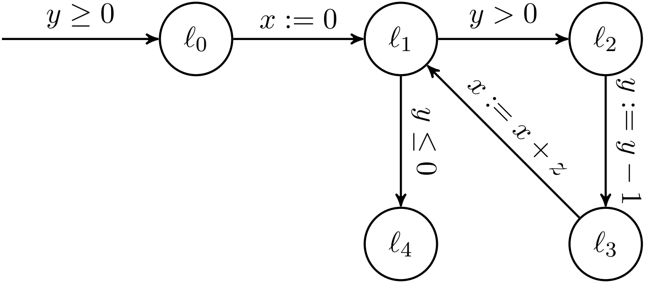
- Path: $\sigma_0 \overset{a_0}{\to} \sigma_1 \overset{a_1}{\to} \sigma_2 \ldots$.
- Initial path: $\sigma_0 \overset{a_0}{\to} \sigma_1 \overset{a_1}{\to} \sigma_2 \ldots$ with $\sigma_0 \in \mathit{Init}$.
- A state is called reachable iff there is an initial path leading to it.

LSTS semantics of LTS

Each LTS $\mathcal{LTS} = (Loc, Var, Lab, Edge, Init)$ induces a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge', Init)$ with

- $\Sigma = Loc \times V$ and
- $Edge' = \{(\sigma, a, \sigma') \mid \sigma \overset{a}{\rightarrow} \sigma'\}$.

Semantics of the simple while-program



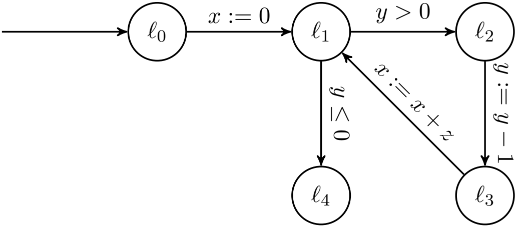
$$\frac{(l, a, \mu, l') \in Edge \quad (\nu, \nu') \in \mu}{(l, \nu) \xrightarrow{a} (l', \nu')}$$

Modeling a simple while-program

```
           method mult(int y, int z){
             int x;
$\ell_0$     x := 0;
$\ell_1$
             while( y > 0 ) {
$\ell_2$       y := y-1;
$\ell_3$       x := x+z;
             }
$\ell_4$   }
```

Modeling a simple while-program

```
        method mult(int y, int z){
          int x;
$\ell_0$          x := 0;
$\ell_1$
          while( y > 0 ) {
$\ell_2$            y := y-1;
$\ell_3$            x := x+z;
          }
$\ell_4$    }
```



Contents

1. Hybrid systems
2. Labeled state transition systems
3. Labeled transition systems
4. Temporal logics
5. CTL model checking

Temporal logics

Assume

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

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

Propositional logic

- Abstract syntax:

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

Propositional logic

- Abstract syntax:

$$\varphi \ ::= \ a \mid (\varphi \land \varphi) \mid (\neg \varphi)$$

  with $a \in AP$.

- Syntactic sugar: *true*, *false*, $\lor, \rightarrow, \leftrightarrow, \ldots$
- Omit parentheses when no confusion
- Semantics (in the context of a state $\sigma \in \Sigma$):

Propositional logic

- Abstract syntax:

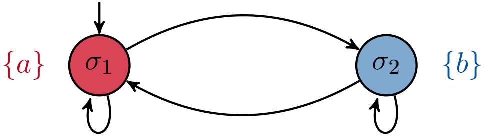
$$\varphi \ ::= \ a \mid (\varphi \land \varphi) \mid (\neg \varphi)$$

  with $a \in AP$.

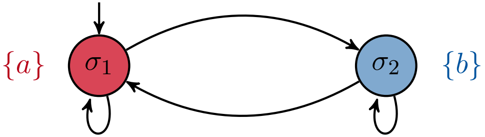
- Syntactic sugar: *true*, *false*, $\lor, \rightarrow, \leftrightarrow, \ldots$
- Omit parentheses when no confusion
- Semantics (in the context of a state $\sigma \in \Sigma$):

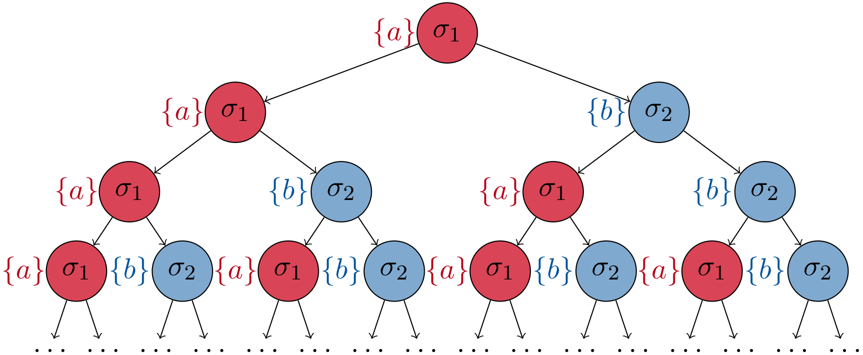
$$\begin{array}{lll} \sigma \models a & \textit{iff} & a \in L(\sigma), \\ \sigma \models (\varphi_1 \land \varphi_2) & \textit{iff} & \sigma \models \varphi_1 \textit{ and } \sigma \models \varphi_2, \\ \sigma \models (\neg \varphi) & \textit{iff} & \sigma \not\models \varphi. \end{array}$$

Computation tree



Computation tree

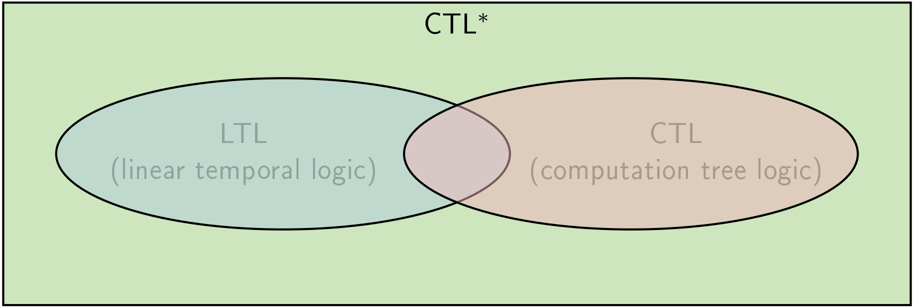




Temporal logics

In the computation tree, temporal logic formulas can describe

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



Examples for path formulae

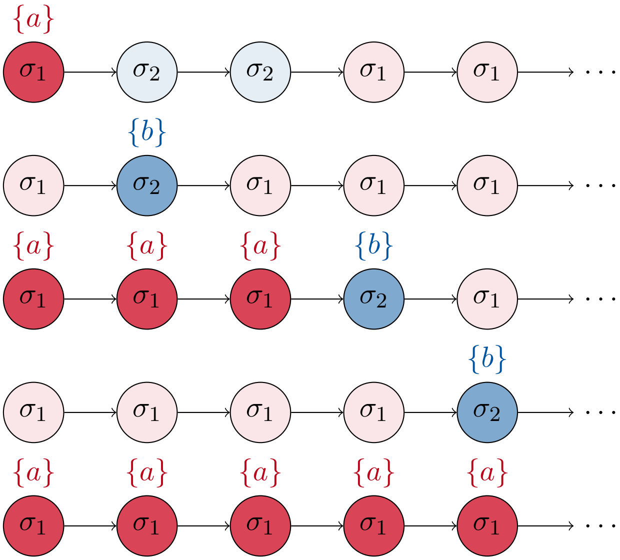
proposition a :

next $\mathcal{X} b$:

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

finally $\mathcal{F} b$:

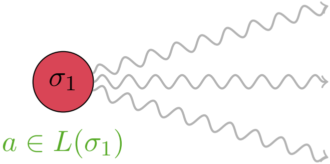
globally $\mathcal{G} a$:



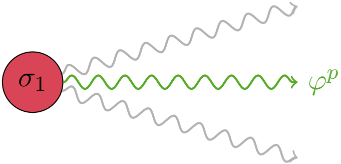
$\{a\}$
σ_1 σ_2 σ_2 σ_1 σ_1 …
$\{b\}$
σ_1 σ_2 σ_1 σ_1 σ_1 …
$\{a\}$ $\{a\}$ $\{a\}$ $\{b\}$
σ_1 σ_1 σ_1 σ_2 σ_1 …
$\{b\}$
σ_1 σ_1 σ_1 σ_1 σ_2 …
$\{a\}$ $\{a\}$ $\{a\}$ $\{a\}$ $\{a\}$
σ_1 σ_1 σ_1 σ_1 σ_1 …

Examples for state formulae

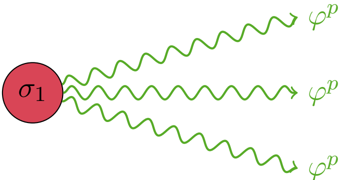
proposition a :



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



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



CTL* syntax

CTL* state formulae:

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

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

CTL* path formulae:

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

where φ^s are CTL* state formulae.

CTL* 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", state formula)
- $\mathcal{F}\varphi^p := \mathit{true}\,\mathcal{U}\varphi^p$ ("finally" or "eventually", path formula)
- $\mathcal{G}\varphi^p := \neg\mathcal{F}\neg\varphi^p$ ("globally" or "always", path formula)

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.

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

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*

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$\mathcal{L}, \sigma \models a \qquad \textit{iff} \quad a \in L(\sigma)$$

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*	$a \in L(\sigma)$
$\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s$	*iff*	

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 $\pi = \sigma_0 \to \sigma_1 \to \dots$ of $\mathcal{LSTS}$, let $\pi(i)$ denote σ_i, and let π^i denote $\sigma_i \to \sigma_{i+1} \to \dots$.

$$\begin{array}{lll} \mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\ \mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \end{array}$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$
\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg \varphi^s & \textit{iff} &
\end{array}
$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

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

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$
\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg \varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} &
\end{array}
$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg \varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \ldots \textit{ of } \mathcal{LSTS}
\end{array}$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$
\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg \varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \ldots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} &
\end{array}
$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg\varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \ldots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} & \mathcal{L}, \pi(0) \models \varphi^s
\end{array}$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg\varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \ldots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} & \mathcal{L}, \pi(0) \models \varphi^s \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} &
\end{array}$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$
\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg\varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \ldots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} & \mathcal{L}, \pi(0) \models \varphi^s \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} & \mathcal{L}, \pi \models \varphi_1^p \textit{ and } \mathcal{L}, \pi \models \varphi_2^p
\end{array}
$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg \varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \ldots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} & \mathcal{L}, \pi(0) \models \varphi^s \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} & \mathcal{L}, \pi \models \varphi_1^p \textit{ and } \mathcal{L}, \pi \models \varphi_2^p \\
\mathcal{L}, \pi \models \neg \varphi^p & \textit{iff} &
\end{array}$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

$$\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg\varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \ldots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} & \mathcal{L}, \pi(0) \models \varphi^s \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} & \mathcal{L}, \pi \models \varphi_1^p \textit{ and } \mathcal{L}, \pi \models \varphi_2^p \\
\mathcal{L}, \pi \models \neg\varphi^p & \textit{iff} & \mathcal{L}, \pi \not\models \varphi^p
\end{array}$$

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 $\sigma_i \rightarrow \sigma_{i+1} \rightarrow \ldots$.

$$\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg \varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E} \varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \rightarrow \ldots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} & \mathcal{L}, \pi(0) \models \varphi^s \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} & \mathcal{L}, \pi \models \varphi_1^p \textit{ and } \mathcal{L}, \pi \models \varphi_2^p \\
\mathcal{L}, \pi \models \neg \varphi^p & \textit{iff} & \mathcal{L}, \pi \not\models \varphi^p \\
\mathcal{L}, \pi \models \mathcal{X} \varphi^p & \textit{iff} &
\end{array}$$

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 $\pi = \sigma_0 \to \sigma_1 \to \dots$ of $\mathcal{LSTS}$, let $\pi(i)$ denote σ_i, and
let π^i denote $\sigma_i \to \sigma_{i+1} \to \dots$.

$$\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg\varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \dots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} & \mathcal{L}, \pi(0) \models \varphi^s \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} & \mathcal{L}, \pi \models \varphi_1^p \textit{ and } \mathcal{L}, \pi \models \varphi_2^p \\
\mathcal{L}, \pi \models \neg\varphi^p & \textit{iff} & \mathcal{L}, \pi \not\models \varphi^p \\
\mathcal{L}, \pi \models \mathcal{X}\varphi^p & \textit{iff} & \mathcal{L}, \pi^1 \models \varphi^p
\end{array}$$

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 $\pi = \sigma_0 \to \sigma_1 \to \dots$ of $\mathcal{LSTS}$, let $\pi(i)$ denote σ_i, and let π^i denote $\sigma_i \to \sigma_{i+1} \to \dots$.

$$\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg \varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E} \varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \dots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \pi \models \varphi^s & \textit{iff} & \mathcal{L}, \pi(0) \models \varphi^s \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} & \mathcal{L}, \pi \models \varphi_1^p \textit{ and } \mathcal{L}, \pi \models \varphi_2^p \\
\mathcal{L}, \pi \models \neg \varphi^p & \textit{iff} & \mathcal{L}, \pi \not\models \varphi^p \\
\mathcal{L}, \pi \models \mathcal{X} \varphi^p & \textit{iff} & \mathcal{L}, \pi^1 \models \varphi^p \\
\mathcal{L}, \pi \models \varphi_1^p \, \mathcal{U} \, \varphi_2^p & \textit{iff} &
\end{array}$$

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

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

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 $\sigma_i \rightarrow \sigma_{i+1} \rightarrow \ldots$.

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

$\mathcal{L} \models \varphi^s$ iff

CTL* semantics

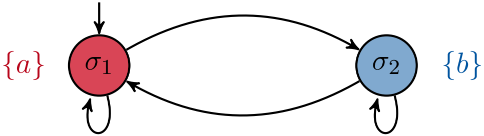
Assume $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ to be a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ along with a labeling function $L : \Sigma \to 2^{AP}$, where AP is a finite set of atomic propositions.

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

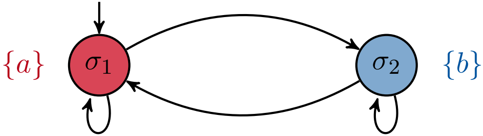
$$
\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg\varphi^s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

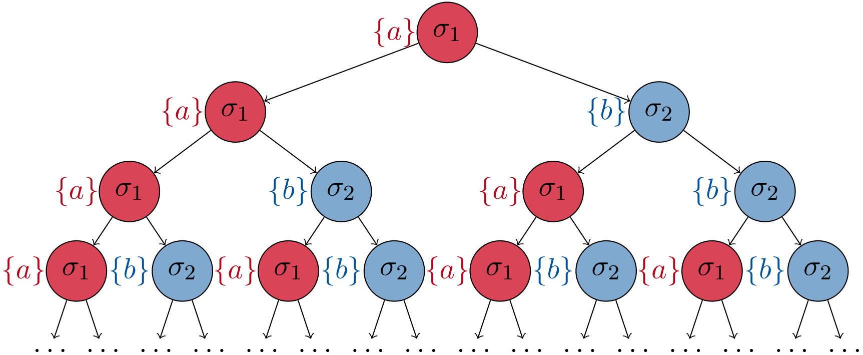
$\mathcal{L} \models \varphi^s$ iff $\mathcal{L}, \sigma_0 \models \varphi^s$ for all initial states σ_0 of $\mathcal{LSTS}$.

Computation tree



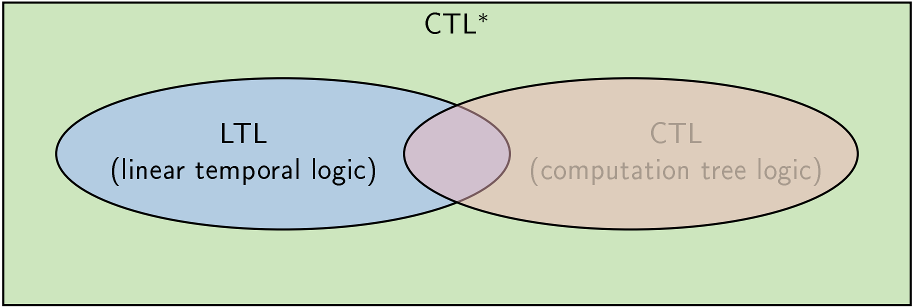
Computation tree





$\{a\}$ σ_1
$\{a\}$ σ_1
$\{b\}$ σ_2
$\{a\}$ σ_1
$\{b\}$ σ_2
$\{a\}$ σ_1
$\{b\}$ σ_2
$\{a\}$ σ_1 $\{b\}$ σ_2 $\{a\}$ σ_1 $\{b\}$ σ_2 $\{a\}$ σ_1 $\{b\}$ σ_2 $\{a\}$ σ_1 $\{b\}$ σ_2
$\dots$

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:

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

  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*.

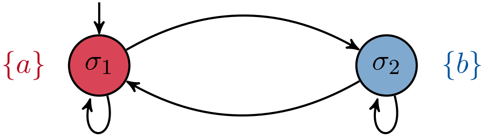
LTL semantics

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 $\sigma_i \to \sigma_{i+1} \to \ldots$.

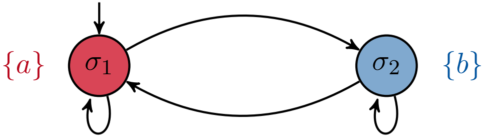
$$
\begin{array}{lll}
\mathcal{L}, \pi \models a & \textit{iff} & a \in L(\pi(0)), \\
\mathcal{L}, \pi \models \varphi_1^p \wedge \varphi_2^p & \textit{iff} & \mathcal{L}, \pi \models \varphi_1^p \textit{ and } \mathcal{L}, \pi \models \varphi_2^p, \\
\mathcal{L}, \pi \models \neg\varphi^p & \textit{iff}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

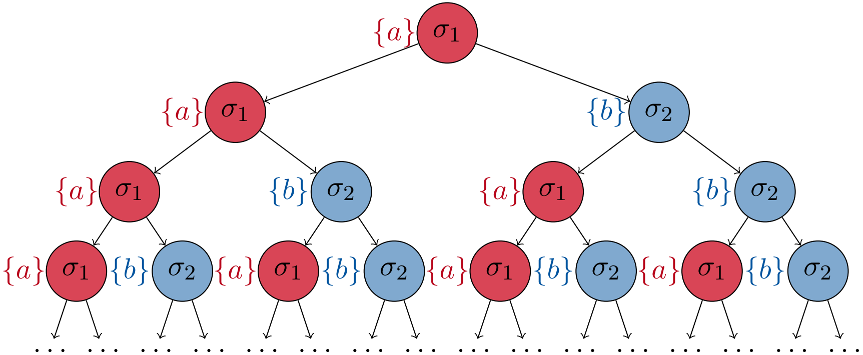
$\mathcal{LSTS} \models \varphi^p$ iff $\pi \models \varphi^p$ for all paths π of $\mathcal{LSTS}$ starting in an initial state.

Computation tree

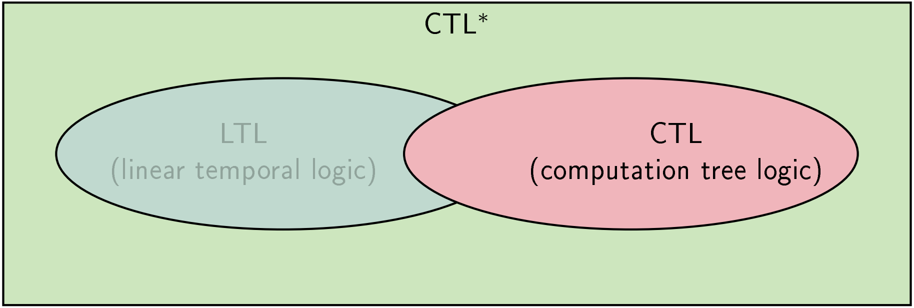


Computation tree





The relation of LTL, CTL, and CTL*



CTL syntax

CTL state formulae:

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

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

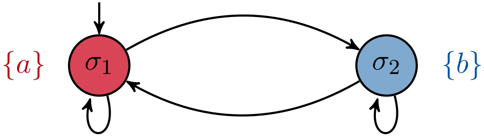
Assume $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ to be a labeled state transition system $\mathcal{LSTS} = (\Sigma, Lab, Edge, Init)$ along with a labeling function $L : \Sigma \to 2^{AP}$, where AP is a finite set of atomic propositions.
For a path $\pi = \sigma_0 \to \sigma_1 \to \ldots$ of $\mathcal{LSTS}$,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

$$
\begin{array}{lll}
\mathcal{L}, \sigma \models a & \textit{iff} & a \in L(\sigma) \\
\mathcal{L}, \sigma \models \varphi_1^s \wedge \varphi_2^s & \textit{iff} & \mathcal{L}, \sigma \models \varphi_1^s \textit{ and } \mathcal{L}, \sigma \models \varphi_2^s \\
\mathcal{L}, \sigma \models \neg \varphi^s & \textit{iff} & \mathcal{L}, \sigma \not\models \varphi^s \\
\mathcal{L}, \sigma \models \mathbf{E}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for some path } \pi = \sigma \to \ldots \textit{ of } \mathcal{LSTS} \\
\mathcal{L}, \sigma \models \mathbf{A}\varphi^p & \textit{iff} & \mathcal{L}, \pi \models \varphi^p \textit{ for all } \pi = \sigma_0 \to \sigma_1 \to \ldots \textit{ with } \sigma_0 = \sigma
\end{array}
$$

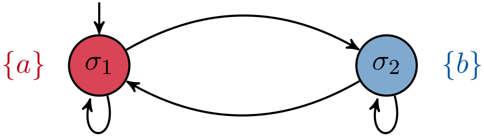
$$
\begin{array}{lll}
\mathcal{L}, \pi \models \mathcal{X}\varphi^s & \textit{iff} & \mathcal{L}, \pi(1) \models \varphi^s \\
\mathcal{L}, \pi \models \varphi_1^s \,\mathcal{U}\, \varphi_2^s & \textit{iff} & \textit{exists } 0 \leq j \textit{ with } \mathcal{L}, \pi(j) \models \varphi_2^s \textit{ and} \\
 & & \mathcal{L}, \pi(i) \models \varphi_1^s \textit{ for all } 0 \leq i < j.
\end{array}
$$

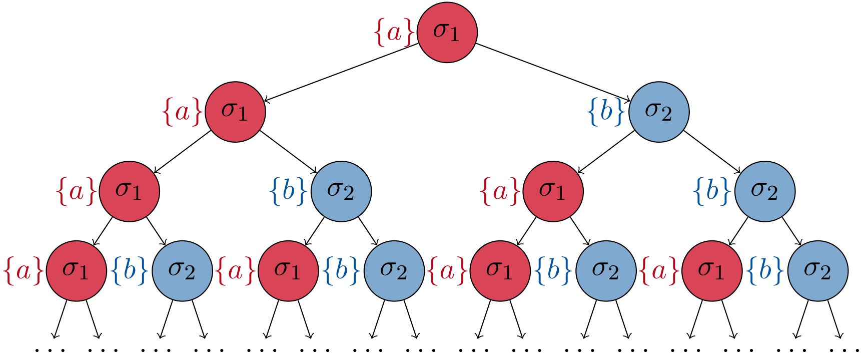
$\mathcal{L} \models \varphi^s$ iff $\mathcal{L}, \sigma_0 \models \varphi^s$ for all initial states σ_0 of $\mathcal{LSTS}$.

Computation tree

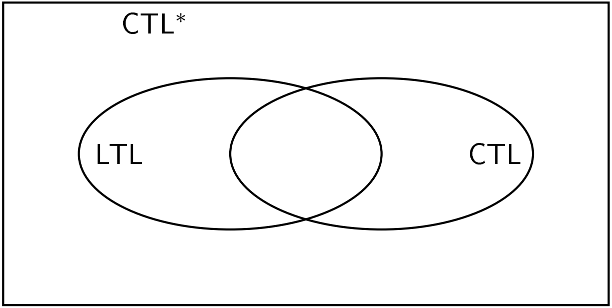


Computation tree





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.

Contents

1 Hybrid systems

2 Labeled state transition systems

3 Labeled transition systems

4 Temporal logics

5 CTL model checking

CTL (explicit) model checking

Next learn a model checking algorithm to decide whether a labeled state transition system satisfies a CTL formula.

CTL (explicit) model checking

Next learn a model checking algorithm to decide whether a labeled state transition system satisfies a CTL formula.

For $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ (being a labeled state transition system $(\Sigma, Lab, Edge, Init)$ with a labeling function L) and for a CTL formula ψ^s, CTL model checking labels the states of $\mathcal{L}$ recursively with the sub-formulae of ψ^s inside-out, such that exactly those states are labeled with each sub-formula at which the given sub-formula holds.

CTL (explicit) model checking

Next learn a model checking algorithm to decide whether a labeled state transition system satisfies a CTL formula.

For $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ (being a labeled state transition system $(\Sigma, Lab, Edge, Init)$ with a labeling function L) and for a CTL formula ψ^s, CTL model checking labels the states of $\mathcal{L}$ recursively with the sub-formulae of ψ^s inside-out, such that exactly those states are labeled with each sub-formula at which the given sub-formula holds.

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

CTL (explicit) model checking

Next learn a model checking algorithm to decide whether a labeled state transition system satisfies a CTL formula.

For $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ (being a labeled state transition system $(\Sigma, Lab, Edge, Init)$ with a labeling function L) and for a CTL formula ψ^s, CTL model checking labels the states of $\mathcal{L}$ recursively with the sub-formulae of ψ^s inside-out, such that exactly those states are labeled with each sub-formula at which the given sub-formula holds.

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

CTL (explicit) model checking

Next learn a model checking algorithm to decide whether a labeled state transition system satisfies a CTL formula.

For $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ (being a labeled state transition system $(\Sigma, Lab, Edge, Init)$ with a labeling function L) and for a CTL formula ψ^s, CTL model checking labels the states of $\mathcal{L}$ recursively with the sub-formulae of ψ^s inside-out, such that exactly those states are labeled with each sub-formula at which the given sub-formula holds.

- 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

Next learn a model checking algorithm to decide whether a labeled state transition system satisfies a CTL formula.

For $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ (being a labeled state transition system $(\Sigma, Lab, Edge, Init)$ with a labeling function L) and for a CTL formula ψ^s, CTL model checking labels the states of $\mathcal{L}$ recursively with the sub-formulae of ψ^s inside-out, such that exactly those states are labeled with each sub-formula at which the given sub-formula holds.

- 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.

CTL (explicit) model checking

Next learn a model checking algorithm to decide whether a labeled state transition system satisfies a CTL formula.

For $\mathcal{L} = (\Sigma, \mathit{Lab}, \mathit{Edge}, \mathit{Init}, L)$ (being a labeled state transition system $(\Sigma, \mathit{Lab}, \mathit{Edge}, \mathit{Init})$ with a labeling function L) and for a CTL formula ψ^s, CTL model checking labels the states of $\mathcal{L}$ recursively with the sub-formulae of ψ^s inside-out, such that exactly those states are labeled with each sub-formula at which the given sub-formula holds.

- 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

Next learn a model checking algorithm to decide whether a labeled state transition system satisfies a CTL formula.

For $\mathcal{L} = (\Sigma, Lab, Edge, Init, L)$ (being a labeled state transition system $(\Sigma, Lab, Edge, Init)$ with a labeling function L) and for a CTL formula ψ^s, CTL model checking labels the states of $\mathcal{L}$ recursively with the sub-formulae of ψ^s inside-out, such that exactly those states are labeled with each sub-formula at which the given sub-formula holds.

- 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.
- 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, for $\mathbf{E}\psi_1^s\ \mathcal{U}\ \psi_2^s$ 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, for $\mathbf{E}\psi_1^s\ \mathcal{U}\ \psi_2^s$ 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, for $\mathbf{E}\psi_1^s\ \mathcal{U}\ \psi_2^s$ 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, for $\mathbf{E}\psi_1^s \; \mathcal{U} \; \psi_2^s$ 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, for $\mathbf{E}\psi_1^s \,\mathcal{U}\, \psi_2^s$ 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, for $\mathbf{E}\psi_1^s\,\mathcal{U}\,\psi_2^s$ 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, for $\mathbf{A}\psi_1^s\,\mathcal{U}\,\psi_2^s$ 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, for $\mathbf{E}\psi_1^s\,\mathcal{U}\,\psi_2^s$ 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, for $\mathbf{A}\psi_1^s\,\mathcal{U}\,\psi_2^s$ 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, for $\mathbf{E}\psi_1^s\ \mathcal{U}\ \psi_2^s$ 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, for $\mathbf{A}\psi_1^s\ \mathcal{U}\ \psi_2^s$ 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.