Computation Tree Logic

EECS 4315

www.cse.yorku.ca/course/4315/

Syntax of CTL

Definition

The *state formulas* are defined by

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

The *path formulas* are defined by

$$\varphi ::= \bigcirc\Phi \mid \Phi \, \mathsf{U} \, \Phi$$

Semantics of CTL

Definition

The relation $\models$ is defined by

$$\begin{array}{rcl} s \models \text{true} & & \\ s \models a & \text{iff} & a \in \ell(s) \\ s \models \Phi \wedge \Psi & \text{iff} & s \models \Phi \text{ and } s \models \Psi \\ s \models \neg\Phi & \text{iff} & \text{not}(s \models \Phi) \\ s \models \exists\varphi & \text{iff} & \exists\pi \in \mathit{Paths}(s) : \pi \models \varphi \\ s \models \forall\varphi & \text{iff} & \forall\pi \in \mathit{Paths}(s) : \pi \models \varphi \end{array}$$

and

$$\begin{array}{rcl} \pi \models \bigcirc\Phi & \text{iff} & \pi[1] \models \Phi \\ \pi \models \Phi \, \mathsf{U} \, \Psi & \text{iff} & \exists i \geq 0 : \pi[i] \models \Psi \text{ and } \forall 0 \leq j < i : \pi[j] \models \Phi \end{array}$$

Satisfaction Set

Definition

The *satisfaction set* $Sat(\Phi)$ is defined by

$$Sat(\Phi) = \{\, s \in S \mid s \models \Phi \,\}.$$

Basic idea

Compute $Sat(\Phi)$ by recursion on the structure of Φ.

$TS \models \Phi$ iff $I \subseteq Sat(\Phi)$.

Alternative view

Label each state with the subformulas of Φ that it satisfies.

Model checking CTL

Definition

The *state formulas* are defined by

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

The *path formulas* are defined by

$$\varphi ::= \bigcirc\Phi \mid \Phi \, \mathsf{U} \, \Phi$$

Model checking CTL

Definition

The *state formulas* are defined by

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

The *path formulas* are defined by

$$\varphi ::= \bigcirc\Phi \mid \Phi \,\mathsf{U}\, \Phi$$

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \,\mathsf{U}\, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \,\mathsf{U}\, \Phi)$$

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \, \mathsf{U} \, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \, \mathsf{U} \, \Phi)$$

Question

What is $\mathit{Sat}(\text{true})$?

Model checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \,\mathsf{U}\, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \,\mathsf{U}\, \Phi)$$

Question

What is $Sat(\text{true})$?

Answer

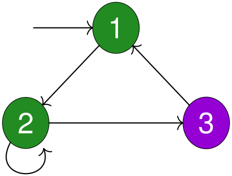
$Sat(\text{true}) = S$

Alternative view

Label each state with true.

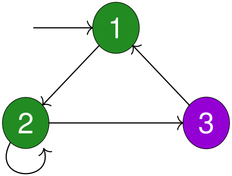
Example

true



Example

true



$$
\begin{array}{ccc}
1 & \mapsto & \{\text{true}\} \\
2 & \mapsto & \{\text{true}\} \\
3 & \mapsto & \{\text{true}\}
\end{array}
$$

Model checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \, \mathsf{U} \, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \, \mathsf{U} \, \Phi)$$

Question

What is $Sat(a)$?

Model checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \land \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \,\mathsf{U}\, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \,\mathsf{U}\, \Phi)$$

Question

What is $Sat(a)$?

Answer

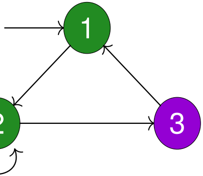
$Sat(a) = \{ s \in S \mid a \in \ell(s) \}$

Alternative view

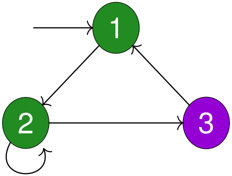
Label each state s satisfying $a \in \ell(s)$ with a.

Example

green



green



$$\begin{array}{lll} 1 & \mapsto & \{\text{green}\} \\ 2 & \mapsto & \{\text{green}\} \\ 3 & \mapsto & \emptyset \end{array}$$

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \,\mathsf{U}\, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \,\mathsf{U}\, \Phi)$$

Question

What is $Sat(\Phi \wedge \Psi)$?

Model checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \land \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \mathbin{\mathsf{U}} \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \mathbin{\mathsf{U}} \Phi)$$

Question

What is $\mathit{Sat}(\Phi \land \Psi)$?

Answer

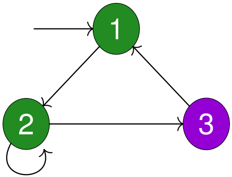
$\mathit{Sat}(\Phi \land \Psi) = \mathit{Sat}(\Phi) \cap \mathit{Sat}(\Psi)$

Alternative view

Label states, that are labelled with both Φ and Ψ, also with $\Phi \land \Psi$.

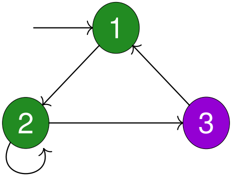
Example

$\text{green} \wedge \text{purple}$



Example

$$\text{green} \wedge \text{purple}$$



$$\begin{array}{lcl} 1 & \mapsto & \{\text{green}\} \\ 2 & \mapsto & \{\text{green}\} \\ 3 & \mapsto & \{\text{purple}\} \end{array}$$

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \,\mathsf{U}\, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \,\mathsf{U}\, \Phi)$$

Question

What is $Sat(\neg\Phi)$?

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \,\mathsf{U}\, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \,\mathsf{U}\, \Phi)$$

Question

What is $Sat(\neg\Phi)$?

Answer

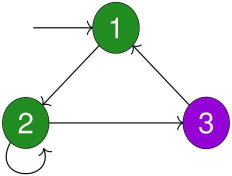
$$Sat(\neg\Phi) = S \setminus Sat(\Phi)$$

Alternative view

Label each state, that is not labelled with Φ, with $\neg\Phi$.

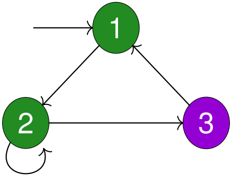
Example

$\neg(\text{green} \wedge \text{purple})$



Example

$$\neg(\text{green} \wedge \text{purple})$$



$$\begin{array}{lll} 1 & \mapsto & \{\text{green}, \neg(\text{green} \wedge \text{purple})\} \\ 2 & \mapsto & \{\text{green}, \neg(\text{green} \wedge \text{purple})\} \\ 3 & \mapsto & \{\text{purple}, \neg(\text{green} \wedge \text{purple})\} \end{array}$$

Model checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \ \mathsf{U} \ \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \ \mathsf{U} \ \Phi)$$

Question

What is $\mathit{Sat}(\exists\bigcirc\Phi)$?

Model checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \; \mathsf{U} \; \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \; \mathsf{U} \; \Phi)$$

Question

What is $Sat(\exists\bigcirc\Phi)$?

Answer

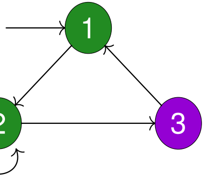
$Sat(\exists\bigcirc\Phi) = \{ s \in S \mid Post(s) \cap Sat(\Phi) \neq \emptyset \}$ where
$Post(s) = \{ s' \in S \mid s \rightarrow s' \}$.

Alternative view

Labels those states, that have a direct successor labelled with Φ, also with $\exists\bigcirc\Phi$.

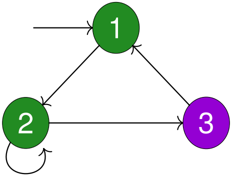
Example

$\exists\bigcirc$green



Example

$\exists\bigcirc \text{green}$



$$\begin{aligned}1 &\mapsto \{\text{green}, \exists\bigcirc\text{green}\}\\ 2 &\mapsto \{\text{green}, \exists\bigcirc\text{green}\}\\ 3 &\mapsto \{\exists\bigcirc\text{green}\}\end{aligned}$$

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \mathbin{\mathsf{U}} \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \mathbin{\mathsf{U}} \Phi)$$

Question

What is $\mathit{Sat}(\exists(\Phi \mathbin{\mathsf{U}} \Psi))$?

Model checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \, \mathsf{U} \, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \, \mathsf{U} \, \Phi)$$

Question

What is $Sat(\exists(\Phi \, \mathsf{U} \, \Psi))$?

Proposition

$Sat(\exists(\Phi \, \mathsf{U} \, \Psi))$ is the smallest subset T of S such that

(a) $Sat(\Psi) \subseteq T$ and

(b) if $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset$ then $s \in T$.

Model checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \,\mathsf{U}\, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \,\mathsf{U}\, \Phi)$$

Question

What is $Sat(\exists(\Phi \,\mathsf{U}\, \Psi))$?

Proposition

$Sat(\exists(\Phi \,\mathsf{U}\, \Psi))$ is the smallest subset T of S such that

(a) $Sat(\Psi) \subseteq T$ and

(b) if $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset$ then $s \in T$.

Question

Does such a smallest subset exist?

Crash Course on Order Theory

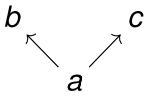
Partially Ordered Set

Definition

A *partially ordered set* is a tuple $\langle A, \sqsubseteq \rangle$ consisting of

- a set A and
- a relation $\sqsubseteq \subseteq A \times A$ satisfying for all a, b, and $c \in A$,
  - $a \sqsubseteq a$,
  - if $a \sqsubseteq b$ and $b \sqsubseteq a$ then $a = b$, and
  - if $a \sqsubseteq b$ and $b \sqsubseteq c$ then $a \sqsubseteq c$.

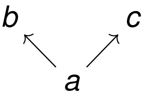
Examples

- 

  depicts the partially ordered set

$$\langle \{a, b, c\}, \{(a, a), (a, b), (a, c), (b, b), (c, c)\} \rangle.$$

Examples

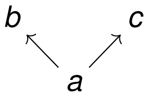
- 

  depicts the partially ordered set

  $$\langle \{a, b, c\}, \{(a, a), (a, b), (a, c), (b, b), (c, c)\} \rangle.$$

- $\langle [0, 1], \leq \rangle$ is a partially ordered set.

Examples

- 

  depicts the partially ordered set

  $$\langle \{a, b, c\}, \{(a, a), (a, b), (a, c), (b, b), (c, c)\} \rangle.$$

- $\langle [0, 1], \leq \rangle$ is a partially ordered set.
- Let S be a set (of states). Then 2^S denotes the set of subsets of S. $\langle 2^S, \subseteq \rangle$ is a partially ordered set.

Least Upper Bound

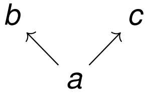
Definition

Let $\langle A, \sqsubseteq \rangle$ be a partially ordered set and $B \subseteq A$.

- $a \in A$ is an *upper bound* of B iff $b \sqsubseteq a$ for all $b \in B$.
- $a \in A$ is a *least upper bound* of B iff
  - a is an upper bound of B, and
  - for all $a' \in A$, if a' is an upper bound of B then $a \sqsubseteq a'$.

Examples

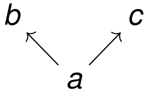
- The subset $\{b, c\}$ of



does not have a least upper bound.

Examples

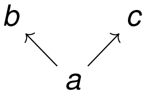
- The subset $\{b, c\}$ of



  does not have a least upper bound.
- $\langle [0, 1], \leq \rangle$
  The least upper bound of $(0, 0.5)$ is 0.5.

Examples

- The subset $\{b, c\}$ of



  does not have a least upper bound.
- $\langle [0, 1], \leq \rangle$
  The least upper bound of $(0, 0.5)$ is 0.5.
- $\langle 2^S, \subseteq \rangle$
  For $X \subseteq 2^S$, its least upper bound is $\bigcup X$.

Least Upper Bound

Proposition

Let $\langle A, \sqsubseteq \rangle$ be a partially ordered set and $B \subseteq A$. If B has a least upper bound, then it is unique.

Notation

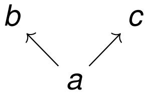
The least upper bound of B is denoted by $\sqcup B$.

Monotone Function

Definition

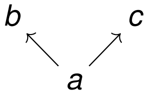
Let $\langle A, \sqsubseteq \rangle$ be a partially ordered set. A function $F : A \to A$ is *monotone* iff for all $a, b \in A$, if $a \sqsubseteq b$ then $F(a) \sqsubseteq F(b)$.

Examples

- 

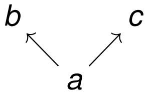
  The function $F : \{a, b, c\} \to \{a, b, c\}$ defined by $F(a) = a$, $F(b) = a$ and $F(c) = c$ is monotone.

Examples

- 

  The function $F : \{a, b, c\} \to \{a, b, c\}$ defined by $F(a) = a$, $F(b) = a$ and $F(c) = c$ is monotone.
- $\langle [0, 1], \leq \rangle$
  The function $F : [0, 1] \to [0, 1]$ defined by $F(r) = \frac{r}{2}$ is monotone.

Examples

- 

  The function $F : \{a, b, c\} \to \{a, b, c\}$ defined by $F(a) = a$, $F(b) = a$ and $F(c) = c$ is monotone.
- $\langle [0, 1], \leq \rangle$
  The function $F : [0, 1] \to [0, 1]$ defined by $F(r) = \frac{r}{2}$ is monotone.
- $\langle 2^S, \subseteq \rangle$
  Let $X \subseteq S$. The function $F : 2^S \to 2^S$ defined by $F(Y) = Y \cap X$ is monotone.

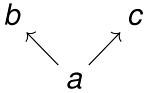
Complete Lattice

Definition

A partially ordered set $\langle A, \sqsubseteq \rangle$ is a *complete lattice* if every subset of A has a least upper bound and a greatest lower bound.

Examples

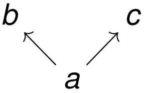
- The partially ordered set



is not a complete lattice.

Examples

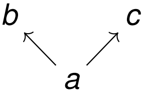
- The partially ordered set



  is not a complete lattice.
- $\langle [0, 1], \leq \rangle$ is a complete lattice.

Examples

- The partially ordered set



  is not a complete lattice.
- $\langle [0, 1], \leq \rangle$ is a complete lattice.
- $\langle 2^S, \subseteq \rangle$ is a complete lattice.

Fixed Points

Definition

Consider the function $F : A \rightarrow A$. Then

- $a \in A$ is a *fixed point* of F iff $F(a) = a$,
- $a \in A$ is a *pre-fixed point* of F iff $F(a) \sqsubseteq a$, and
- $a \in A$ is a *post-fixed point* of F iff $a \sqsubseteq F(a)$.

Corollary of Knaster-Tarski Fixed Point Theorem

Theorem

Let $\langle A, \sqsubseteq \rangle$ be a complete lattice. If the function $F : A \to A$ is monotone, then it has a least fixed point (which is the least pre-fixed point) and a greatest fixed point (which is the greatest post-fixed point).

Bronislaw Knaster (1893–1980)

- Recipient of the Nagroda panstwowa (1963)
- **Knaster's fixed point theorem** *If the function $F : 2^S \rightarrow 2^S$ is monotone then F has a least fixed point.*

Source: Konrad Jacobs

Alfred Tarski (1901–1983)

- Member of the United States National Academy of Sciences (1965)
- Fellow of the British Academy (1966)
- Member of the Royal Netherlands Academy of Arts and Science (1958)
- Strongly influenced the dissertation of Dana Scott (Turing award winner of 1976)
- **Tarski's fixed point theorem** *If* $\langle A, \sqsubseteq \rangle$ *is a complete lattice and* $F : A \to A$ *is a monotone function then the set of fixed points of* F *is a complete lattice.*

Source: George M. Bergman

Least Fixed Point

Theorem

Let $\langle A, \sqsubseteq \rangle$ be a finite *complete lattice and $F : A \to A$ a monotone function. Let*

$$A_n = \begin{cases} \sqcup\emptyset & \text{if } n = 0 \\ F(A_{n-1}) & \text{otherwise} \end{cases}$$

Then $F(A_n) = A_n$ for some $n \in \mathbb{N}$ and A_n is the least fixed point of F.

Comment

$\sqcup\emptyset$ is the least element of A, that is, $\sqcup\emptyset \sqsubseteq a$ for all $a \in A$.

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi \, \mathsf{U} \, \Phi) \mid \forall\bigcirc\Phi \mid \forall(\Phi \, \mathsf{U} \, \Phi)$$

Proposition

$Sat(\exists(\Phi \, \mathsf{U} \, \Psi))$ is the smallest subset T of S such that

- $Sat(\Psi) \subseteq T$ and
- if $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset$ then $s \in T$.

Model Checking CTL

Definition

The *formulas* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg \Phi \mid \exists \bigcirc \Phi \mid \exists(\Phi \; \mathsf{U} \; \Phi) \mid \forall \bigcirc \Phi \mid \forall(\Phi \; \mathsf{U} \; \Phi)$$

Proposition

$Sat(\exists(\Phi \; \mathsf{U} \; \Psi))$ is the smallest subset T of S such that

- $Sat(\Psi) \subseteq T$ and
- if $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset$ then $s \in T$.

Question

How can we use Knaster's theorem to prove that such a set T exists?

Model Checking CTL

Proposition

$Sat(\exists(\Phi \, \mathsf{U} \, \Psi))$ is the smallest subset T of S such that

- $Sat(\Psi) \subseteq T$ and
- if $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset$ then $s \in T$.

Proposition

$Sat(\exists(\Phi \mathsf{U} \Psi))$ is the smallest subset T of S such that

- $Sat(\Psi) \subseteq T$ and
- if $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset$ then $s \in T$.

Definition

The function $F : 2^S \rightarrow 2^S$ is defined by

$$F(T) = Sat(\Psi) \cup \{ s \in Sat(\Phi) \mid Post(s) \cap T \neq \emptyset \}.$$

Model Checking CTL

Proposition

$Sat(\exists(\Phi \mathbin{U} \Psi))$ is the smallest subset T of S such that

- $Sat(\Psi) \subseteq T$ and
- if $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset$ then $s \in T$.

Definition

The function $F : 2^S \to 2^S$ is defined by

$$F(T) = Sat(\Psi) \cup \{\, s \in Sat(\Phi) \mid Post(s) \cap T \neq \emptyset \,\}.$$

Proposition

The function F is monotone.

Model Checking CTL

Proposition

$Sat(\exists(\Phi \, \mathsf{U} \, \Psi))$ is the smallest subset T of S such that

- $Sat(\Psi) \subseteq T$ and
- if $s \in Sat(\Phi)$ and $Post(s) \cap T \neq \emptyset$ then $s \in T$.

Definition

The function $F : 2^S \to 2^S$ is defined by

$$F(T) = Sat(\Psi) \cup \{ s \in Sat(\Phi) \mid Post(s) \cap T \neq \emptyset \}.$$

Proposition

The function F is monotone.

Corollary

F has a least pre-fixed point, that is, there exists a smallest set T such that $F(T) \subseteq T$.

Model Checking CTL

Sat(Φ):
switch (Φ):

true	:	**return** S
a	:	**return** $\{ s \in S \mid a \in \ell(s) \}$
$\Phi \wedge \Psi$	:	**return** $Sat(\Phi) \cap Sat(\Psi)$
$\neg\Phi$	:	**return** $S \setminus Sat(\Phi)$
$\exists\bigcirc\Phi$	:	**return** $\{ s \in S \mid Post(s) \cap Sat(\Phi) \neq \emptyset \}$
$\exists(\Phi \cup \Psi)$	:	$T := \emptyset$
		while $T \neq F(T)$
		$\quad T := F(T)$
		return T
…		

Model checking CTL

Definition

The function $G : 2^S \rightarrow 2^S$ is defined by

$$G(T) = \begin{cases} Sat(\Psi) & \text{if } T = \emptyset \\ T \cup \{\, s \in Sat(\Phi) \mid Post(s) \cap T \neq \emptyset \,\} & \text{otherwise} \end{cases}$$

Model checking CTL

Definition

The function $G : 2^S \to 2^S$ is defined by

$$G(T) = \begin{cases} Sat(\Psi) & \text{if } T = \emptyset \\ T \cup \{ s \in Sat(\Phi) \mid Post(s) \cap T \neq \emptyset \} & \text{otherwise} \end{cases}$$

Proposition

For all $n \geq 0$, $F^n(\emptyset) \subseteq F^{n+1}(\emptyset)$.

Model checking CTL

Definition

The function $G : 2^S \to 2^S$ is defined by

$$G(T) = \begin{cases} Sat(\Psi) & \text{if } T = \emptyset \\ T \cup \{\, s \in Sat(\Phi) \mid Post(s) \cap T \neq \emptyset \,\} & \text{otherwise} \end{cases}$$

Proposition

For all $n \geq 0$, $F^n(\emptyset) \subseteq F^{n+1}(\emptyset)$.

Proposition

For all $n \geq 1$, $Sat(\Psi) \subseteq F^n(\emptyset)$.

Model checking CTL

Definition

The function $G : 2^S \to 2^S$ is defined by

$$G(T) = \begin{cases} Sat(\Psi) & \text{if } T = \emptyset \\ T \cup \{ s \in Sat(\Phi) \mid Post(s) \cap T \neq \emptyset \} & \text{otherwise} \end{cases}$$

Proposition

For all $n \geq 0$, $F^n(\emptyset) \subseteq F^{n+1}(\emptyset)$.

Proposition

For all $n \geq 1$, $Sat(\Psi) \subseteq F^n(\emptyset)$.

Proposition

For all $n \geq 1$, $F^n(\emptyset) = G^n(\emptyset)$.

Model Checking CTL

$Sat(\Phi)$:
switch (Φ):

$$\begin{array}{lll} & \ldots & \\ \exists(\Phi \, \mathsf{U} \, \Psi) & : & T := G(\emptyset) \\ & & \textbf{while } T \neq G(T) \\ & & \quad T := G(T) \\ & & \textbf{return } T \\ & \ldots & \end{array}$$

Model Checking CTL

$Sat(\Phi)$:
switch (Φ):

$\ldots$

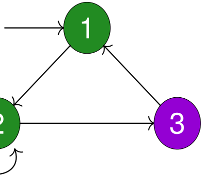
$\exists(\Phi \cup \Psi)$: $E := Sat(\Psi)$
$T := E$
while $E \neq \emptyset$
  let $s' \in E$
  $E := E \setminus \{s'\}$
  for all $s \in Pre(s')$
    if $s \in Sat(\Phi) \setminus T$
      $E := E \cup \{s\}$
      $T := T \cup \{s\}$
return T

$\ldots$

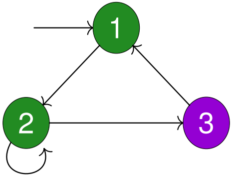
where $Pre(s') = \{ s'' \in S \mid s'' \to s' \}$.

Example

$\exists$(green U purple)



$\exists$(green U purple)



$$\begin{array}{lcl} 1 & \mapsto & \{\text{green}, \exists(\text{green U purple})\} \\ 2 & \mapsto & \{\text{green}, \exists(\text{green U purple})\} \\ 3 & \mapsto & \{\text{purple}, \exists(\text{green U purple})\} \end{array}$$

Time Complexity of CTL Model Checking

By improving the model checking algorithm (see, for example the textbook of Baier and Katoen for details), we obtain

Theorem

For a transition system *TS*, with *N* states and *K* transitions, and a CTL formula Φ, the model checking problem $TS \models \Phi$ can be decided in time $\mathcal{O}((N + K) \cdot |\Phi|)$.

Time Complexity of CTL Model Checking

By improving the model checking algorithm (see, for example the textbook of Baier and Katoen for details), we obtain

Theorem

For a transition system TS, with N states and K transitions, and a CTL formula Φ, the model checking problem $TS \models \Phi$ can be decided in time $\mathcal{O}((N + K) \cdot |\Phi|)$.

Theorem

For a transition system TS, with N states and K transitions, and a LTL formula φ, the model checking problem $TS \models \varphi$ can be decided in time $\mathcal{O}((N + K) \cdot 2^{|\varphi|})$.

Time Complexity of CTL Model Checking

By improving the model checking algorithm (see, for example the textbook of Baier and Katoen for details), we obtain

Theorem

For a transition system TS, with N states and K transitions, and a CTL formula Φ, the model checking problem $TS \models \Phi$ can be decided in time $\mathcal{O}((N + K) \cdot |\Phi|)$.

Theorem

For a transition system TS, with N states and K transitions, and a LTL formula φ, the model checking problem $TS \models \varphi$ can be decided in time $\mathcal{O}((N + K) \cdot 2^{|\varphi|})$.

Theorem

If P $\neq$ NP then there exist LTL formulas φ_n whose size is a polynomial in n, for which equivalent CTL formulas exist, but not of size polynomial in n.