Model Checking CTL

EECS 4315

www.eecs.yorku.ca/course/4315/

Model checking CTL

Definition

The *satisfaction set* $Sat(f)$ is defined by

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

Basic idea

Compute $Sat(f)$ by recursion on the structure of f.

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

Alternative view

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

Model checking CTL

$$
\begin{aligned}
Sat(a) &= \{\, s \in S \mid a \in \ell(s) \,\} \\
Sat(f \wedge g) &= Sat(f) \cap Sat(g) \\
Sat(\neg f) &= S \setminus Sat(f) \\
Sat(\exists \bigcirc f) &= \{\, s \in S \mid succ(s) \cap Sat(f) \neq \emptyset \,\} \\
Sat(\forall \bigcirc f) &=? \\
Sat(\exists (f \mathbin{\mathsf{U}} g)) &=? \\
Sat(\forall (f \mathbin{\mathsf{U}} g)) &=?
\end{aligned}
$$

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(\forall \bigcirc f)$?

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(\forall \bigcirc f)$?

Answer

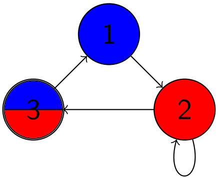
$Sat(\forall \bigcirc f) = \{\, s \in S \mid succ(s) \subseteq Sat(f) \,\}$.

Alternative view

Labels those states, with all direct successors labelled with f, with $\forall \bigcirc f$.

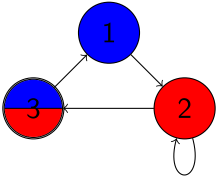
Example

$\forall \bigcirc$ red



Example

$\forall \bigcirc$ red



$$1 \mapsto \{\forall \bigcirc \text{red}\}$$
$$2 \mapsto \{\text{red}, \forall \bigcirc \text{red}\}$$
$$3 \mapsto \{\text{red}\}$$

The *formulas* are defined by

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

Question

What is $Sat(\exists(f \mathsf{U} g))$?

Model checking CTL

$$
\begin{aligned}
& s \in Sat(\exists(f \mathbin{\mathsf{U}} g)) \\
\text{iff } & s \models \exists(f \mathbin{\mathsf{U}} g) \\
\text{iff } & \exists p \in Paths(s) : p \models f \mathbin{\mathsf{U}} g \\
\text{iff } & \exists p \in Paths(s) : \exists i \geq 0 : p[i] \models g \wedge \forall 0 \leq j < i : p[j] \models f \\
\text{iff } & \exists p \in Paths(s) : p[0] \models g \vee (\exists i \geq 1 : p[i] \models g \wedge \forall 0 \leq j < i : p[j] \models f) \\
\text{iff } & \exists p \in Paths(s) : p[0] \models g \vee \\
& \qquad (p[0] \models f \wedge \exists i \geq 1 : p[i] \models g \wedge \forall 1 \leq j < i : p[j] \models f) \\
\text{iff } & s \models g \vee (s \models f \wedge \exists s \rightarrow t : t \models \exists(f \mathbin{\mathsf{U}} g)) \\
\text{iff } & s \in Sat(g) \vee (s \in Sat(f) \wedge \exists t \in succ(s) : t \in Sat(\exists(f \mathbin{\mathsf{U}} g))) \\
\text{iff } & s \in Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap Sat(\exists(f \mathbin{\mathsf{U}} g)) \neq \emptyset \,\}
\end{aligned}
$$

Model checking CTL

As we have seen

$$s \in Sat(\exists(f \,\mathsf{U}\, g))$$
$$\text{iff } s \in Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap Sat(\exists(f \,\mathsf{U}\, g)) \neq \emptyset \,\}$$

As we have seen

$$s \in Sat(\exists(f \mathbin{\mathsf{U}} g))$$
$$\text{iff } s \in Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap Sat(\exists(f \mathbin{\mathsf{U}} g)) \neq \emptyset \,\}$$

Hence, the set $Sat(\exists(f \mathbin{\mathsf{U}} g))$ is a subset T of S such that

$$T = Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap T \neq \emptyset \,\}$$

Model checking CTL

As we have seen

$$s \in Sat(\exists(f \mathbin{\mathsf{U}} g))$$
$$\text{iff } s \in Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap Sat(\exists(f \mathbin{\mathsf{U}} g)) \neq \emptyset \,\}$$

Hence, the set $Sat(\exists(f \mathbin{\mathsf{U}} g))$ is a subset T of S such that

$$T = Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap T \neq \emptyset \,\}$$

Proposition

The set $Sat(\exists(f \mathbin{\mathsf{U}} g))$ is the smallest subset T of S such that

$$T = Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap T \neq \emptyset \,\}$$

Model checking CTL

As we have seen

$$s \in Sat(\exists(f \mathbin{\mathsf{U}} g))$$
$$\text{iff } s \in Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap Sat(\exists(f \mathbin{\mathsf{U}} g)) \neq \emptyset \,\}$$

Hence, the set $Sat(\exists(f \mathbin{\mathsf{U}} g))$ is a subset T of S such that

$$T = Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap T \neq \emptyset \,\}$$

Proposition

The set $Sat(\exists(f \mathbin{\mathsf{U}} g))$ is the smallest subset T of S such that

$$T = Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap T \neq \emptyset \,\}$$

Question

Does such a smallest subset exist?

Smallest subset

Definition

A function $G : 2^S \to 2^S$ is *monotone* if for all $T, U \in 2^S$,

$$\text{if } T \subseteq U \text{ then } G(T) \subseteq G(U).$$

Smallest subset

Definition

A function $G : 2^S \to 2^S$ is *monotone* if for all $T, U \in 2^S$,

$$\text{if } T \subseteq U \text{ then } G(T) \subseteq G(U).$$

Knaster's fixed point theorem

If the set S is finite and the function $G : 2^S \to 2^S$ is monotone, then there exists a smallest $T \in 2^S$ such that $G(T) = T$.

Smallest subset

Definition

A function $G : 2^S \to 2^S$ is *monotone* if for all $T, U \in 2^S$,

$$\text{if } T \subseteq U \text{ then } G(T) \subseteq G(U).$$

Knaster's fixed point theorem

If the set S is finite and the function $G : 2^S \to 2^S$ is monotone, then there exists a smallest $T \in 2^S$ such that $G(T) = T$.

This smallest $T \in 2^S$ is known as the *least fixed point* of G.

Bronislaw Knaster (1893–1980)

- Polish mathematician
- Received his Ph.D. degree from University of Warsaw
- Proved his fixed point theorem in 1928

Source: Konrad Jacobs

Knaster's fixed point theorem

Definition

For each $n \in \mathbb{N}$, the set G_n is defined by

$$G_n = \begin{cases} \emptyset & \text{if } n = 0 \\ G(G_{n-1}) & \text{otherwise} \end{cases}$$

Knaster's fixed point theorem

Definition

For each $n \in \mathbb{N}$, the set G_n is defined by

$$G_n = \begin{cases} \emptyset & \text{if } n = 0 \\ G(G_{n-1}) & \text{otherwise} \end{cases}$$

Proposition

For all $n \in \mathbb{N}$, $G_n \subseteq G_{n+1}$.

Definition

For each $n \in \mathbb{N}$, the set G_n is defined by

$$G_n = \begin{cases} \emptyset & \text{if } n = 0 \\ G(G_{n-1}) & \text{otherwise} \end{cases}$$

Proposition

For all $n \in \mathbb{N}$, $G_n \subseteq G_{n+1}$.

Proof

We prove this by induction on n. In the base case, $n = 0$, we have that

$$G_0 = \emptyset \subseteq G_1.$$

In the inductive case, we have $n \geq 1$. By induction, $G_{n-1} \subseteq G_n$. Since G is monotone, we have that

$$G_n = G(G_{n-1}) \subseteq G(G_n) = G_{n+1}.$$

Knaster's fixed point theorem

Proposition

$G_n = G_{n+1}$ for some $n \in \mathbb{N}$.

Knaster's fixed point theorem

Proposition

$G_n = G_{n+1}$ for some $n \in \mathbb{N}$.

Proof

Suppose that S contains m elements. Towards a contradiction, assume that $G_n \neq G_{n+1}$ for all $n \in \mathbb{N}$. Then $G_n \subset G_{n+1}$ for all $n \in \mathbb{N}$. Hence, G_n contains at least n elements. Therefore, G_{m+1} contains more elements than S. This contradicts that $G_{m+1} \subseteq S$.

Knaster's fixed point theorem

Proposition

$G_n = G_{n+1}$ for some $n \in \mathbb{N}$.

Proof

Suppose that S contains m elements. Towards a contradiction, assume that $G_n \neq G_{n+1}$ for all $n \in \mathbb{N}$. Then $G_n \subset G_{n+1}$ for all $n \in \mathbb{N}$. Hence, G_n contains at least n elements. Therefore, G_{m+1} contains more elements than S. This contradicts that $G_{m+1} \subseteq S$.

We denote the G_n with $G_n = G_{n+1}$ by $\mathit{fix}(G)$.

Knaster's fixed point theorem

Proposition

For all $T \subseteq S$, if $G(T) = T$ then $\mathit{fix}(G) \subseteq T$.

Knaster's fixed point theorem

Proposition

For all $T \subseteq S$, if $G(T) = T$ then $\mathit{fix}(G) \subseteq T$.

Proof

First, we prove that for all $n \in \mathbb{N}$, $G_n \subseteq T$ by induction on n. In the base case, $n = 0$, we have that $G_0 = \emptyset \subseteq T$. In the inductive case, we have $n \geq 1$. By induction, $G_{n-1} \subseteq T$. Since G is monotone, $G_n = G(G_{n-1}) \subseteq G(T) = T$. Since $\mathit{fix}(G) = G_n$ for some $n \in \mathbb{N}$, we can conclude that $\mathit{fix}(G) \subseteq T$.

Knaster's fixed point theorem

Proposition

For all $T \subseteq S$, if $G(T) = T$ then $\mathit{fix}(G) \subseteq T$.

Proof

First, we prove that for all $n \in \mathbb{N}$, $G_n \subseteq T$ by induction on n. In the base case, $n = 0$, we have that $G_0 = \emptyset \subseteq T$. In the inductive case, we have $n \geq 1$. By induction, $G_{n-1} \subseteq T$. Since G is monotone, $G_n = G(G_{n-1}) \subseteq G(T) = T$. Since $\mathit{fix}(G) = G_n$ for some $n \in \mathbb{N}$, we can conclude that $\mathit{fix}(G) \subseteq T$.

Corollary

$\mathit{fix}(G)$ is the smallest subset T of S such that $G(T) = T$.

Smallest subset

Definition

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

$$F(T) = Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap T \neq \emptyset \,\}$$

Smallest subset

Definition

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

$$F(T) = Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap T \neq \emptyset \,\}$$

Proposition

F is monotone.

Smallest subset

Definition

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

$$F(T) = Sat(g) \cup \{\, s \in Sat(f) \mid succ(s) \cap T \neq \emptyset \,\}$$

Proposition

F is monotone.

Proof

Let $T, U \in 2^S$. Assume that $T \subseteq U$. Let $s \in F(T)$. It remains to prove that $s \in F(U)$. Then $s \in Sat(g)$ or $s \in Sat(f)$ and $succ(s) \cap T = \emptyset$. We distinguish two cases. If $s \in Sat(g)$ then $s \in F(U)$. If $s \in Sat(f)$ and $succ(s) \cap T = \emptyset$ then $succ(s) \cap U = \emptyset$ since $T \subseteq U$. Hence, $s \in F(U)$.

Model checking CTL

```
Sat(f):
switch (f) {
case a :          return { s ∈ S | a ∈ ℓ(s) }
case f ∧ g :      return Sat(f) ∩ Sat(g)
case ¬f :         return S \ Sat(f)
case ∃◯f :        return { s ∈ S | succ(s) ∩ Sat(f) ≠ ∅ }
case ∀◯f :        return { s ∈ S | succ(s) ⊆ Sat(f) }
case ∃(f U g) :   T = ∅
                  while T ≠ F(T)
                    T = F(T)
                  return T
case ∀(f U g) :   ...
}
```

Model checking CTL

```
case ∃(f U g) :
  E = Sat(g)
  T = E
  while E ≠ ∅
    let t ∈ E
    E = E \ {t}
    for all s ∈ pred(t)
      if s ∈ Sat(f) \ T
        E = E ∪ {s}
        T = T ∪ {s}
  return T
```

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(\forall(f \,\mathsf{U}\, g))$?

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(\forall (f \, \mathsf{U} \, g))$?

Answer

The set $Sat(\forall (f \, \mathsf{U} \, g))$ is the smallest subset T of S such that

$$T = Sat(g) \cup \{ \, s \in Sat(f) \mid succ(s) \subseteq T \, \}$$

Size of a CTL formula

$$
\begin{aligned}
|a| &= 1 \\
|f \wedge g| &= 1 + |f| + |g| \\
|\neg f| &= 1 + |f| \\
|\exists \bigcirc f| &= 1 + |f| \\
|\exists (f \mathbin{\mathsf{U}} g)| &= 1 + |f| + |g| \\
|\forall \bigcirc f| &= 1 + |f| \\
|\forall (f \mathbin{\mathsf{U}} g)| &= 1 + |f| + |g|
\end{aligned}
$$

The 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 f, the model checking problem $TS \models f$ can be decided in time $O((N+K)|f|)$.

The 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 f, the model checking problem $TS \models f$ can be decided in time $O((N+K)|f|)$.

Theorem

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

The 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 f, the model checking problem $TS \models f$ can be decided in time $O((N+K)|f|)$.

Theorem

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

Theorem

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

Course evaluation

The course evaluation for this course can now be completed at `https://courseevaluations.yorku.ca`

I would really appreciate it if you would take the time to complete the course evaluation. Your feedback allows me to improve the course for future students.

Since 13 students have already completed the evaluation, I will bring cup cakes for the last lecture.