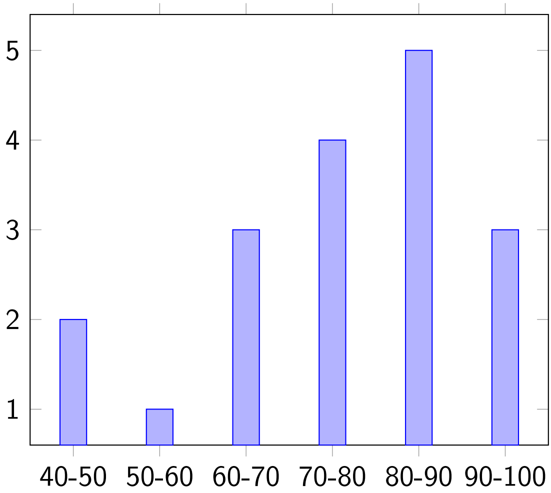
Quiz 3: grade distribution



Average: 78%

Computational Tree Logic

EECS 4315

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

Syntax

The *state formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists g \mid \forall g$$

The *path formulas* are defined by

$$g ::= \bigcirc f \mid f \, \mathsf{U} \, f$$

Syntax

The *state formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists g \mid \forall g$$

The *path formulas* are defined by

$$g ::= \bigcirc f \mid f \,\mathsf{U}\, f$$

The *formulas* are defined by

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

Semantics of CTL

$$
\begin{aligned}
s \models a & \quad \text{iff} \quad a \in \ell(s) \\
s \models f_1 \wedge f_2 & \quad \text{iff} \quad s \models f_1 \wedge s \models f_2 \\
s \models \neg f & \quad \text{iff} \quad s \not\models f \\
s \models \exists g & \quad \text{iff} \quad \exists p \in \mathit{Paths}(s) : p \models g \\
s \models \forall g & \quad \text{iff} \quad \forall p \in \mathit{Paths}(s) : p \models g
\end{aligned}
$$

and

$$
\begin{aligned}
p \models \bigcirc f & \quad \text{iff} \quad p[1] \models f \\
p \models f_1 \mathrel{\mathsf{U}} f_2 & \quad \text{iff} \quad \exists i \geq 0 : p[i] \models f_2 \wedge \forall 0 \leq j < i : p[j] \models f_1
\end{aligned}
$$

Semantics of CTL

Question

Recall that

$$\exists\Diamond f = \exists(\text{true}\ \mathsf{U}\ f).$$

How is

$$s \models \exists\Diamond f$$

defined?

Semantics of CTL

Question

Recall that

$$\exists\Diamond f = \exists(\text{true} \; \mathsf{U} \; f).$$

How is

$$s \models \exists\Diamond f$$

defined?

Answer

$\exists p \in \mathit{Paths}(s) : \exists i \geq 0 : p[i] \models f$

Question

Recall that

$$\forall\Diamond f = \forall(\text{true} \, \mathsf{U} \, f).$$

How is

$$s \models \forall\Diamond f$$

defined?

Semantics of CTL

Question

Recall that

$$\forall\Diamond f = \forall(\text{true}\ \mathsf{U}\ f).$$

How is

$$s \models \forall\Diamond f$$

defined?

Answer

$\forall p \in \mathit{Paths}(s) : \exists i \geq 0 : p[i] \models f$

Semantics of CTL

Question

Recall that

$$\exists \Box f = \neg \forall (\mathsf{true}\ \mathsf{U}\ \neg f).$$

How is

$$s \models \exists \Box f$$

defined?

Semantics of CTL

Question

Recall that

$$\exists\Box f = \neg\forall(\text{true}\ \mathsf{U}\ \neg f).$$

How is

$$s \models \exists\Box f$$

defined?

Answer

$\exists p \in Paths(s) : \forall i \geq 0 : p[i] \models f$

Question

Recall that

$$\forall \Box f = \neg \exists (\text{true} \, \mathsf{U} \, \neg f).$$

How is

$$s \models \forall \Box f$$

defined?

Semantics of CTL

Question

Recall that

$$\forall\Box f = \neg\exists(\text{true}\,\mathsf{U}\,\neg f).$$

How is

$$s \models \forall\Box f$$

defined?

Answer

$\forall p \in Paths(s) : \forall i \geq 0 : p[i] \models f$

Semantics of CTL

$$TS \models f \text{ iff } \forall s \in I : s \models f.$$

Example

Question

How to express "Each red light is preceded by an amber light" in CTL?

Example

Question

How to express "Each red light is preceded by an amber light" in CTL?

Answer

$\neg \text{red} \wedge \forall \Box (\exists \bigcirc \text{red} \Rightarrow \text{amber})$

Example

Question

How to express “The light is infinitely often green” in CTL?

Example

Question

How to express "The light is infinitely often green" in CTL?

Answer

$\forall\Box\forall\Diamond$green

Expressiveness of LTL and CTL

Theorem

The property

$$\forall s \in I : \forall p \in Paths(s) : \forall m \geq 0 : \exists q \in Paths(p[m]) : \exists n \geq 0 : q[n] \models a$$

cannot be captured by LTL, but is captured by the CTL formula $\forall\Box\exists\Diamond a$.

Expressiveness of LTL and CTL

Theorem

The property

$$\forall s \in I : \forall p \in Paths(s) : \exists i \geq 0 : \forall j \geq i : p[j..] \models a$$

cannot be captured by CTL, but is captured by the LTL formula $\Diamond\Box a$.

Model checking CTL

Problem

Given a transition system TS and a CTL formula f, check whether $TS \models f$.

Model checking CTL

Problem

Given a transition system TS and a CTL formula f, check whether $TS \models f$.

Definition

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

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

Model checking CTL

Problem

Given a transition system TS and a CTL formula f, check whether $TS \models f$.

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.

Model checking CTL

Problem

Given a transition system TS and a CTL formula f, check whether $TS \models f$.

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 $I \subseteq Sat(f)$.

Model checking CTL

Problem

Given a transition system TS and a CTL formula f, check whether $TS \models f$.

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

The *formulas* are defined by

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

Question

What is $Sat(a)$?

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(a)$?

Answer

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

The *formulas* are defined by

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

Question

What is $Sat(a)$?

Answer

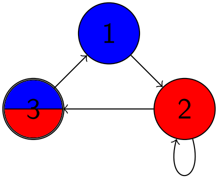
$Sat(a) = \{\, s \in S \mid a \in \ell(s) \,\}$

Alternative view

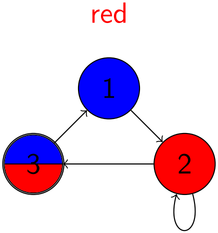
Label each state s satisfying $a \in \ell(s)$ with a.

Example

red



Example



$$1 \mapsto \emptyset$$

$$2 \mapsto \{\text{red}\}$$

$$3 \mapsto \{\text{red}\}$$

The *formulas* are defined by

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

Question

What is $Sat(f \wedge g)$?

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(f \wedge g)$?

Answer

$Sat(f \wedge g) = Sat(f) \cap Sat(g)$

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(f \wedge g)$?

Answer

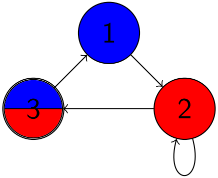
$Sat(f \wedge g) = Sat(f) \cap Sat(g)$

Alternative view

Label states, that are labelled with both f and g, also with $f \wedge g$.

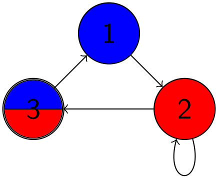
Example

red $\wedge$ blue



Example

red $\wedge$ blue



$1 \mapsto \{\text{blue}\}$

$2 \mapsto \{\text{red}\}$

$3 \mapsto \{\text{red}, \text{blue}, \text{red} \wedge \text{blue}\}$

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(\neg f)$?

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(\neg f)$?

Answer

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

Model checking CTL

The *formulas* are defined by

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

Question

What is $Sat(\neg f)$?

Answer

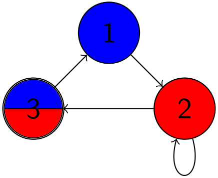
$Sat(\neg f) = S \setminus Sat(f)$

Alternative view

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

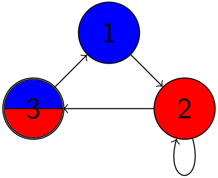
Example

$\neg(\text{red} \wedge \text{blue})$



Example

$\neg(\text{red} \wedge \text{blue})$



$1 \mapsto \{\text{blue}, \neg(\text{red} \wedge \text{blue})\}$

$2 \mapsto \{\text{red}, \neg(\text{red} \wedge \text{blue})\}$

$3 \mapsto \{\text{red}, \text{blue}, \text{red} \wedge \text{blue}\}$

Model checking CTL

The *formulas* are defined by

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

Question

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

Model checking CTL

The *formulas* are defined by

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

Question

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

Answer

$Sat(\exists \bigcirc f) = \{\, s \in S \mid succ(s) \cap Sat(f) \neq \emptyset \,\}$ where $succ(s) = \{\, t \in S \mid s \rightarrow t \,\}$.

Model checking CTL

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists(f \cup f) \mid \forall \bigcirc f \mid \forall(f \cup f)$$

Question

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

Answer

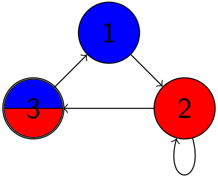
$Sat(\exists \bigcirc f) = \{ s \in S \mid succ(s) \cap Sat(f) \neq \emptyset \}$ where $succ(s) = \{ t \in S \mid s \to t \}$.

Alternative view

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

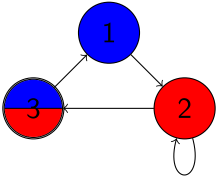
Example

$\exists \bigcirc$ red



Example

$\exists \bigcirc$ red



$1 \mapsto \{\exists \bigcirc \text{red}\}$

$2 \mapsto \{\text{red}, \exists \bigcirc \text{red}\}$

$3 \mapsto \{\text{red}\}$

Model checking CTL

The *formulas* are defined by

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

Question

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

The *formulas* are defined by

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

Question

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

Answer

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

Model checking CTL

The *formulas* are defined by

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

Question

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

Answer

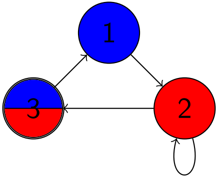
$Sat(\forall \bigcirc f) = \{ s \in S \mid succ(s) \subseteq Sat(f) \}$.

Alternative view

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

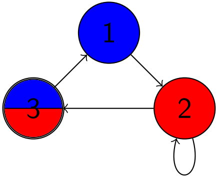
Example

$\forall \bigcirc$ red



Example

$\forall \bigcirc$ red



$1 \mapsto \{\forall \bigcirc \text{red}\}$

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

$3 \mapsto \{\text{red}\}$

Model checking CTL

The *formulas* are defined by

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

Question

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

Model checking CTL

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

Model checking CTL

As we have seen

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

Model checking CTL

As we have seen

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

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

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

Model checking CTL

As we have seen

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

Hence, the set $Sat(\exists(f \ \mathsf{U} \ g))$ 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 $Sat(\exists(f \ \mathsf{U} \ g))$ 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

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

Hence, the set $Sat(\exists(f \;\mathsf{U}\; g))$ 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 $Sat(\exists(f \;\mathsf{U}\; g))$ 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 $n \in \mathbb{N}$, $G_n \subseteq G_{n+1}$.

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

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

$G_m = G_{m+1}$ for some $m \in \mathbb{N}$.

Knaster's fixed point theorem

Proposition

$G_m = G_{m+1}$ for some $m \in \mathbb{N}$.

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 $G_{m+1} \subseteq S$.

Proposition

$G_m = G_{m+1}$ for some $m \in \mathbb{N}$.

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_m with $G_m = G_{m+1}$ 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 $\mathit{fix}(G) = G_m$ for some $m \in \mathbb{N}$, 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 $\mathit{fix}(G) = G_m$ for some $m \in \mathbb{N}$, 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 $F : 2^S \rightarrow 2^S$ 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) : T = ∅
               while T ≠ G(T)
                 T = G(T)
               return T
}
```

Project

Submit the final version of your project proposal before Tuesday February 25.