Formal Specification and Verification

Temporal logic (Part 4)

3.07.2014

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Branching Time Logic: CTL

CTL: Syntax

The class of computational tree logic (CTL) formulas is the smallest set such that

- $\top, \bot$ and each propositional variable $P \in \Pi$ are formulae;
- if F, G are formulae, then so are $F \wedge G, F \vee G, \neg F$;
- if F, G are formulae, then so are
  $A \bigcirc F$ and $E \bigcirc F$,
  $A(F \mathcal{U} G)$ and $E(F \mathcal{U} G)$.

The symbols A and E are called path quantifiers.

CTL: Semantics

Let $T = (S, \to, L)$ be a transition system. We define satisfaction of CTL formulas in T at states $s \in S$ as follows:

$(T, s) \models p$	iff	$p \in L(s)$
$(T, s) \models \neg F$	iff	$(T, s) \models F$ is not the case
$(T, s) \models F \land G$	iff	$(T, s) \models F$ and $(T, s) \models G$
$(T, s) \models F \lor G$	iff	$(T, s) \models F$ or $(T, s) \models G$
$(T, s) \models E \bigcirc F$	iff	$(T, t) \models F$ for some $t \in S$ with $s \to t$
$(T, s) \models A \bigcirc F$	iff	$(T, t) \models F$ for all $t \in S$ with $s \to t$
$(T, s) \models A(F \mathcal{U} G)$	iff	for all computations $\pi = s_0 s_1 \ldots$ of T with $s_0 = s$, there is an $m \geq 0$ such that $(T, s_m) \models G$ and $(T, s_k) \models F$ for all $k < m$
$(T, s) \models E(F \mathcal{U} G)$	iff	there exists a computation $\pi = s_0 s_1 \ldots$ of T with $s_0 = s$, such that there is an $m \geq 0$ such that $(T, s_m) \models G$ and $(T, s_k) \models F$ for all $k < m$

Equivalence

We say that two CTL formulas F and G are (globally) equivalent (written $F \equiv G$) if, for all CTL structures $T = (S, \rightarrow, L)$ and $s \in S$, we have

$$T, s \models F \text{ iff } T, s \models G.$$

Equivalence

We say that two CTL formulas F and G are (globally) equivalent (written $F \equiv G$) if, for all CTL structures $T = (S, \rightarrow, L)$ and $s \in S$, we have

$$T, s \models F \text{ iff } T, s \models G.$$

Examples:

$\neg A \Diamond F \equiv E \Box \neg F$

$\neg E \Diamond F \equiv A \Box \neg F$

$\neg A \bigcirc F \equiv E \bigcirc \neg F$

$A \Diamond F \equiv A[\top \mathcal{U} F]$

$E \Diamond F \equiv E[\top \mathcal{U} F]$

Model Checking

The CTL model checking problem is as follows:

> Given a transition system $T = (S, \to, L)$ and a CTL formula F, check whether T satifies F, i.e., whether $(T, s) \models F$ for all $s \in S$.

Model Checking

The CTL model checking problem is as follows:

Given – a transition system $T = (S, \rightarrow, L)$ with S finite and
– a CTL formula F,
check whether T satifies F, i.e., whether $(T, s) \models F$ for all $s \in S$.

Method (Idea)

(1) Arrange all subformulas F_i of F in a sequence $F_0, \ldots F_k$ in ascending order w.r.t. formula length: for $1 \leq i < j \leq k$, F_i is not longer than F_j;

(2) For all subformulas F_i of F, compute the set

$$sat(F_i) := \{s \in S | (T, s) \models F_i\}$$

in this order (from shorter to longer formulae);

(3) Check whether $S \subseteq sat(F)$.

Model Checking

How to compute $sat(F_i)$

- $p \in \Pi \mapsto sat(p) = \{s \mid L(p,s) = 1\}$
- $sat(\neg F_i) = S \backslash sat(F_i)$
- $sat(F_i \wedge F_j) = sat(F_i) \cap sat(F_j)$
- $sat(F_i \vee F_j) = sat(F_i) \cup sat(F_j)$
- $sat(E \bigcirc F_i) = \{s \mid \exists t \in S : (s \rightarrow t) \wedge t \in sat(F_i)\}$
- $sat(A \bigcirc F_i) = \{s \mid \forall t \in S : (s \rightarrow t) \wedge t \in sat(F_i)\}$
- $sat(E(F_i \mathcal{U} F_j))$ and $sat(A(F_i \mathcal{U} F_j)$ are computed with the following procedures:

Model Checking

$F = E(F_i \mathcal{U} F_j)$

```
sat(F) := T := sat(F_j)
while T =\= {} do
  choose s in T
  T := T \ {s}
  for all t in S with t -> s do
     if t in sat(F_i) and t not in sat(F) then
        sat(F) := sat(F) U {t}
        T := T U {t}
```

$F = A(F_i \mathcal{U} F_j)$

```
sat(F) := T := sat(F_j)
while T =\= {} do
  choose s in T
  T := T \ {s}
  for all t in S with t -> s do
     flag = 1
     for all t' in S with t -> t' do
        if t' not in sat(F) then flag := 0
     if t in sat(F_i) and t not in sat(F) and flag = 1 then
        sat(F) := sat(F) U {t}
        T := T U {t}
```

Examples

- See scans linked directly.
- See also the examples in

Christel Baier and Joost-Pieter Katoen: *"Principles of Model Checking"* pages 344–348.

Model Checking

Theorem. $(T, s) \models F$ iff $s \in sat(F)$.

Consequence. CTL model checking is decidable.

Concerning the complexity, we observe the following: if F is of length n, then at most n sets $sat(F_i)$ need to be computed. How complex is it to compute each such set?

- F is a propositional letter or of the form $F_1 \wedge F_2$ or $\neg F_1$: $O(|S|)$ steps needed;
- F is of the form $E \bigcirc F_j$ or $E(F_i \mathcal{U} F_j)$: $O(|S| + |\rightarrow|)$ steps needed

  the maximum cardinality of the initial set $sat(F_j)$ is $|S|$, and, in the forall loop, each edge from $\rightarrow$ is "touched" at most once (in all iterations of the while);

- F is of the form $A(F_i \mathcal{U} F_j)$: $O(|S| + |\rightarrow|^2)$ steps needed

  the maximum cardinality of the initial set $sat(F_j)$ is $|S|$, the outer forall loop touches each edge from $\rightarrow$ at most once, and the inner forall loop touches each edge at most once for each step done by the outer forall loop.

There exist more efficient algorithms (complexity $|F| \cdot O(|S| + |\rightarrow|)$).

Model Checking

Theorem. $(T, s) \models F$ iff $s \in sat(F)$.

Idea of the proof: Structural induction, taking into account that:

- $sat(\top) = S$, $sat(\bot) = \emptyset$, $sat(p) = \{s \mid p \in L(s)\}, p \in \Pi$
- $sat(\neg F) = S \backslash sat(F); sat(F \wedge G) = sat(F) \cap sat(G)$
- $sat(E \bigcirc F) = \{s \in S \mid Post(s) \cap sat(F) \neq \emptyset\}$
- $E(F \mathcal{U} G) \equiv G \vee (F \wedge E \bigcirc E(F \mathcal{U} G))$

  $Sat(E(F \mathcal{U} G))$ is the smallest subset T of S such that

  (1) $sat(G) \subseteq T$ (2) $s \in sat(F)$ and $\text{Post}(s) \cap T \neq \emptyset$ implies $s \in T$
- $E \Box F \equiv F \wedge E \bigcirc E \Box F$

  $sat(E \Box F)$ is the largest subset T of S such that:

  (1) $T \subseteq sat(F)$ (2) $s \in T$ implies $Post(s) \cap T \neq \emptyset$
- $sat(A(F \mathcal{U} G))$ is the smallest subset T of S satisfying

$$sat(G) \cup \{s \in sat(F) \mid Post(s) \subseteq T\} \subseteq T$$

Model Checking

Lemma. $sat(E(F\mathcal{U}G))$ is the smallest set T with

(1) $sat(G) \subseteq T$

(2) $s \in sat(F)$ and $\text{Post}(s) \cap T \neq \emptyset$ implies $s \in T$

Proof: **1. Show that $T = sat(E(F\mathcal{U}G))$ satisfies (1) and (2).**

This follows from the fact that

$E(F\mathcal{U}G) = G \vee (F \wedge E \bigcirc E(F\mathcal{U}G))$.

(1) $sat(G) \subseteq T$

(2) $s \in sat(F)$ and $\text{Post}(s) \cap T \neq \emptyset$ implies $s \in T$

Model Checking

Lemma. $sat(E(F\mathcal{U}G))$ is the smallest set T with

(1) $sat(G) \subseteq T$

(2) $s \in sat(F)$ and $\text{Post}(s) \cap T \neq \emptyset$ implies $s \in T$

Proof: **2. Show that for any T satisfying (1) and (2), $sat(E(F\mathcal{U}G)) \subseteq T$**

Let $s \in sat(E(F\mathcal{U}G))$

Case 1: $s \in sat(G)$. Then by (1), $s \in T$.

Case 2: $s \notin sat(G)$.

Then there exists a path $\pi = s_0 \dots s_k \dots$ with $s_0 = s$ such that $\pi \models F\mathcal{U}G$.

Let $n \geq 0$ such that
$s_i \models F$ for $0 \leq i \leq n$
$s_{n+1} \models G$.

Model checking

Proof: **2. Show that for any T satisfying (1) and (2),** $sat(E(F\mathcal{U}G)) \subseteq T$
....continued

Then
$s_{n+1} \in sat(G) \in T$,
$s_n \in sat(F)$ and $s_{n+1} \in \text{Post}(s_n) \cap T$, so $s_n \in T$.
$s_{n-1} \in sat(F)$ and $s_n \in \text{Post}(s_{n-1}) \cap T$, so $s_{n-1} \in T$.
...
$s_0 = s \in sat(F)$ and $s_1 \in \text{Post}(s_0) \cap T$, so $s_0 = s \in T$.

Model checking

Remarks:

$EF\mathcal{U}G$ is a fixpoint of the equation $\Phi \equiv G \vee (F \wedge E \bigcirc \Phi)$.

Since $sat(EF\mathcal{U}G)$ is the smallest set T with

(1) $sat(G) \subseteq T$

(2) $s \in sat(F)$ and $\text{Post}(s) \cap T \neq \emptyset$ implies $s \in T$

it can be computed iteratively as follows:

$$T_0 := sat(G)$$

$$T_{i+1} := T_i \cup \{s \in sat(F) \mid Post(s) \cap T_i \neq \emptyset\}$$

Then: $T_0 \subseteq T_1 \subseteq \cdots \subseteq T_j \subseteq T_{j+1} \subseteq \cdots \subseteq sat(E(F\mathcal{U}G))$.

Since S is finite, there exists j such that $T_j = T_{j+1} = \ldots$.

This T_j will be $sat(E(F\mathcal{U}G))$.

Model checking

Remarks:

$sat(E\Box F)$ is the largest set T with

(1) $T \subseteq sat(F)$

(2) $s \in T$ implies $\text{Post}(s) \cap T \neq \emptyset$.

It can be computed iteratively as follows:

$$T_0 := sat(F)$$

$$T_{i+1} := T_i \cap \{s \in sat(F) \mid Post(s) \cap T_i \neq \emptyset\}$$

Then: $T_0 \supseteq T_1 \supseteq \cdots \supseteq T_j \supseteq T_{j+1} \supseteq \cdots \supseteq sat(E(F\mathcal{U}G))$.

Since S is finite, there exists j such that $T_j = T_{j+1} = \ldots$.

This T_j will be $sat(E\Box F)$.