Introduction to Model Checking

René Thiemann

Institute of Computer Science
University of Innsbruck

WS 2007/2008

Outline

Computation Tree Logic

- Formulas over states (capital greek letters)
  - true
  - $a \in AP$ — atomic proposition
  - $\neg \Phi$ and $\Phi \wedge \Psi$ — negation and conjunction
  - $\mathsf{E}\,\varphi$ — there exists a path fulfilling φ
  - $\mathsf{A}\,\varphi$ — all paths fulfill φ
- Formulas over paths (lower case greek letters)
  - $\mathsf{X}\,\Phi$ — the next state fulfills Φ
  - $\Phi\,\mathsf{U}\,\Psi$ — Φ holds until a Ψ-state is reached

Semantics of CTL

A state-formula Φ holds in state s (written $s \models \Phi$) iff

$s \models \text{true}$

$s \models a$ iff $a \in L(s)$

$s \models \neg \Phi$ iff $s \not\models \Phi$

$s \models \Phi \wedge \Psi$ iff $s \models \Phi$ and $s \models \Psi$

$s \models \mathsf{E}\,\varphi$ iff $\pi \models \varphi$ for some path π that starts in s

$s \models \mathsf{A}\,\varphi$ iff $\pi \models \varphi$ for all paths π that start in s

A path-formula φ holds for path π (written $\pi \models \varphi$) iff

$\pi \models \mathsf{X}\,\Phi$ iff $\pi[1] \models \Phi$

$\pi \models \Phi\,\mathsf{U}\,\Psi$ iff $(\exists j \geqslant 0.\, \pi[j] \models \Psi$ and $(\forall\, 0 \leqslant k < j.\, \pi[k] \models \Phi))$

where $\pi[i]$ denotes the state s_i in the path $\pi = s_0\, s_1\, s_2 \ldots$

Transition System Semantics

- For CTL-state-formula Φ, the satisfaction set $Sat(\Phi)$ is defined by:

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

- TS satisfies CTL-state-formula Φ iff Φ holds in all its initial states:

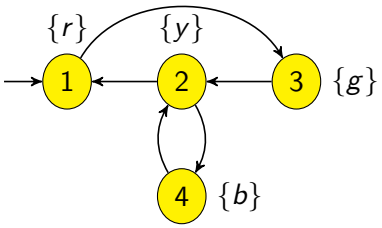
$$I \subseteq Sat(\Phi)$$

Exercise

Provide two transition systems TS_1, TS_2 (same atomic propositions, every state has at least one outgoing edge) and a CTL-formula Φ such that $Traces(TS_1) = Traces(TS_2)$ and $TS_1 \models \Phi$, but $TS_2 \not\models \Phi$.

Exercise

Consider the following transition system which models a traffic light which can blink.



Compute the set $Sat(\Phi)$ for the following formulas.

- $\mathsf{A}\,\mathsf{F}\,y$
- $\mathsf{A}\,\mathsf{G}\,y$
- $\mathsf{A}\,\mathsf{G}\,\mathsf{A}\,\mathsf{F}\,y$
- $\mathsf{A}\,\mathsf{F}\,g$
- $\mathsf{E}\,\mathsf{F}\,g$
- $\mathsf{E}\,\mathsf{G}\,g$
- $\mathsf{A}\,(\neg b\,\mathsf{U}\,b)$
- $\mathsf{E}\,(\neg b\,\mathsf{U}\,b)$
- $\mathsf{A}\,\mathsf{G}\,\mathsf{A}\,\mathsf{F}\,b$

Main Idea

- Semantics of CTL-state formula Φ is defined inductively on sub-formulas ($Sat(\Phi)$ is set of satisfying states)

$\Rightarrow$ compute $Sat(\Psi)$ inductively for all sub-state-formulas Ψ of Φ

$\Rightarrow$ afterwards model checking can be done by checking $I \subseteq Sat(\Phi)$

CTL Model checking boils down to simple set operations
no Büchi automata required

CTL Model Checking: Everything but Until

Let $TS = (S, \rightarrow, I, AP, L)$

- $s \models \text{true}$

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

- $s \models a$ iff $a \in L(s)$

$\Rightarrow Sat(a) = \{s \mid a \in L(s)\}$

- $s \models \neg \Phi$ iff $s \not\models \Phi$

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

- $s \models \Phi \wedge \Psi$ iff $s \models \Phi$ and $s \models \Psi$

$\Rightarrow Sat(\Phi \wedge \Psi) = Sat(\Phi) \cap Sat(\Psi)$

- $s \models \mathsf{EX}\,\Phi$ iff there is a path $s\ s'\ s'' \ldots$ such that $s' \models \Phi$

$\Rightarrow Sat(\mathsf{EX}\,\Phi) = \{s \mid \exists s' : s \rightarrow s', s' \in Sat(\Phi)\}$

- $s \models \mathsf{AX}\,\Phi$ iff for all paths $s\ s'\ s'' \ldots$ it holds: $s' \models \Phi$

$\Rightarrow Sat(\mathsf{AX}\,\Phi) = \{s \mid \forall s' : s \rightarrow s' \Rightarrow s' \in Sat(\Phi)\}$

CTL Model Checking: Until

Remember LTL equivalence:

$$\Phi\,\mathsf{U}\,\Psi \quad \equiv \quad \Psi \vee (\Phi \wedge \mathsf{X}(\Phi\,\mathsf{U}\,\Psi))$$

In CTL:

$$\mathsf{E}\,\Phi\,\mathsf{U}\,\Psi \quad \equiv \quad \Psi \vee (\Phi \wedge \mathsf{EX}(\mathsf{E}\,\Phi\,\mathsf{U}\,\Psi))$$

Hence:

$$\begin{aligned} Sat(\mathsf{E}\,\Phi\,\mathsf{U}\,\Psi) &\equiv Sat(\Psi) \cup (Sat(\Phi) \cap Sat(\mathsf{EX}(\mathsf{E}\,\Phi\,\mathsf{U}\,\Psi))) \\ &\equiv Sat(\Psi) \cup (Sat(\Phi) \cap \{s \mid \exists s' : s \rightarrow s', s' \in Sat(\mathsf{E}\,\Phi\,\mathsf{U}\,\Psi)\}) \end{aligned}$$

One can prove that $Sat(\mathsf{E}\,\Phi\,\mathsf{U}\,\Psi)$ is least set T satisfying

$$T = Sat(\Psi) \cup (Sat(\Phi) \cap \{s \mid \exists s' : s \rightarrow s', s' \in T\})$$

CTL Model Checking: Until (2)

$Sat(\mathsf{E}\,\Phi\,\mathsf{U}\,\Psi)$ is least set T satisfying

$$T = Sat(\Psi) \cup (Sat(\Phi) \cap \{s \mid \exists s' : s \rightarrow s', s' \in T\})$$

Corresponding operation $o : 2^S \rightarrow 2^S$:

$$o(T) = Sat(\Psi) \cup (Sat(\Phi) \cap \{s \mid \exists s' : s \rightarrow s', s' \in T\})$$

Computation of $Sat(\mathsf{E}\,\Phi\,\mathsf{U}\,\Psi)$

Define

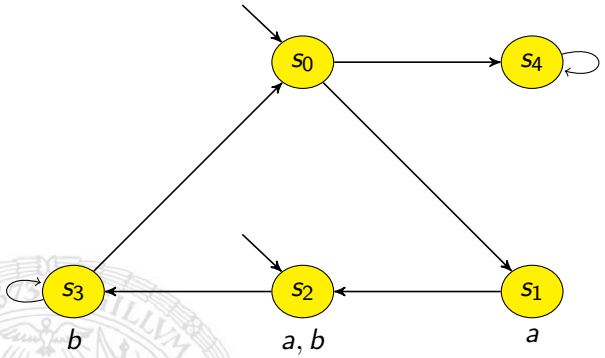
$$o(T) = Sat(\Psi) \cup (Sat(\Phi) \cap \{s \mid \exists s' : s \rightarrow s', s' \in T\})$$

Computation of $Sat(\mathsf{A}\,\Phi\,\mathsf{U}\,\Psi)$

just replace o by o':

$$o'(T) = Sat(\Psi) \cup (Sat(\Phi) \cap \{s \mid \forall s' : s \rightarrow s' \Rightarrow s' \in T\})$$

Example



Complexity of CTL Model Checking

Model-Checking $TS = (S, \rightarrow, I, AP, L) \models \Phi$:

- Number of sets $Sat(\Psi)$ to be computed: $|\Phi|$
- Complexity of computing a single set $Sat(\Psi)$: $\mathcal{O}(|S| + |\rightarrow|) \leqslant \mathcal{O}(|TS|)$
- Complexity of checking $I \subseteq Sat(\Psi)$: $\mathcal{O}(|I|) \leqslant \mathcal{O}(|TS|)$

$\Rightarrow$ overall complexity: $\mathcal{O}(|\Phi| \cdot |TS|)$

Exercises

- Consider the transition system on Slide 15 where $L(s_4)$ is changed to $\{b\}$. Does the modified model satisfy the formula Φ?

$$\Phi = \mathsf{A}\,(a\,\mathsf{U}\,b) \vee \mathsf{E}\,\mathsf{X}\,(\mathsf{A}\,\mathsf{G}\,b)$$

  Use the algorithm of this lecture with an additional rule for disjunction.

$$Sat(\Psi \vee \Psi') = Sat(\Psi) \cup Sat(\Psi')$$

- Provide a direct fixpoint characterization of $Sat(\mathsf{E}\,\mathsf{G}\,\Phi)$ without using the following equivalence.

$$\mathsf{E}\,\mathsf{G}\,\Phi \equiv \neg\mathsf{A}\,\mathsf{F}\,\neg\Phi \equiv \neg\mathsf{A}\,\mathsf{true}\,\mathsf{U}\,\neg\Phi$$