Introduction to Model Checking

René Thiemann

Institute of Computer Science
University of Innsbruck

WS 2007/2008

Outline

- Last Lecture
- Expressiveness of LTL
- Branching Time Logics
- Syntax and Semantics of CTL

Last Lecture

- Translation from LTL to NBAs (exponential)
- Theorem:
  there are LTL formulas which require expontially sized NBAs
- Theorem:
  LTL model checking is co-NP hard
  (reduction via Hamiltonian Path Problem)
- Theorem: LTL model checking is PSPACE complete

Reduction via SAT

Comparing LTL with NBAs

Recall Translation Theorem:

Theorem (Vardi,Wolper)

Every LTL formula φ can be translated into an NBA $\mathcal{A}_\varphi$ such that

$$\mathcal{L}(\varphi) = \mathcal{L}(\mathcal{A}_\varphi).$$

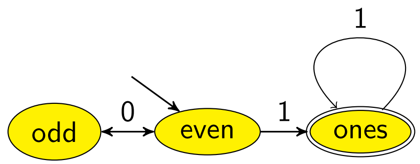
Hence, NBAs are more expressive than LTL

Comparing LTL with NBAs (2)

Consider the language

$$\mathcal{L} = \{w \in 2^\omega \mid w = 0\dots01^\omega \text{ with an even number of 0's}\}$$

- $\mathcal{L}$ is recognized by the following NBA



- There is no LTL formula which defines $\mathcal{L}$

  Proof idea: – show that $\mathcal{L}$ is (modulo) counting
  – show that $\mathcal{L}(\varphi)$ is non-counting for every LTL formula φ

Hence, NBAs are strictly more expressive than LTL

Other Limits of LTL (and NBAs)

- Currently: Model Checking $TS \models \varphi$ iff $\mathcal{L}(TS) \subseteq \mathcal{L}(\varphi)$

$\Rightarrow$ The following transition system satisfies every formula

Properties only speaking about allowed traces are limited

Examples which cannot be expressed (contain existence)

- a beverage will be delivered
- at every time there is a way to reach the main menu

Linear and branching temporal logic

- *Linear* temporal logic:

  "statements about all paths (starting in a state)"

  - $s \models \mathsf{G}\, a$ iff for all possible paths starting in s always a

- *Branching* temporal logic:

  "statements about all or some paths starting in a state"

  - $s \models \mathsf{AG}\, a$ iff for **all** paths starting in s always a
  - $s \models \mathsf{EG}\, a$ iff for **some** path starting in s always a
  - nesting of path quantifiers is allowed

- Checking $s \models \mathsf{E}\, \varphi$ in LTL can be done using $s \not\models \mathsf{A}\, \neg\varphi$
  - ... but this does not work for nested formulas such as $\mathsf{AG}\,\mathsf{EF}\, a$

Branching temporal logics

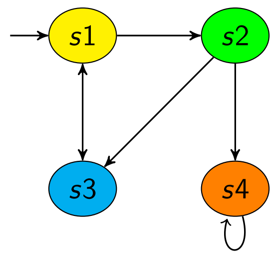
There are various branching temporal logics:

- Computation Tree Logic (CTL)
- Extended Computation Tree Logic (CTL*)
  - combines LTL and CTL into a single framework
- Alternation-free modal μ-calculus
- Modal μ-calculus
- S2S
- ...

Linear vs Branching Temporal Logic

- Semantics is based on a branching notion of time
  - an infinite tree of states obtained by unfolding transition system
  - one “time instant” may have several possible successor “time instants”
- Incomparable expressiveness
  - there are properties that can be expressed in LTL, but not in CTL
  - there are properties that can be expressed in CTL, but not in LTL
- Distinct model-checking algorithms, and their time complexities

Transition Systems and Trees



“behavior” in a state s	path-based: $trace(s)$	state-based: computation tree of s
temporal logic	LTL: path formulas φ $s \models \varphi$ iff $\forall w \in Traces(s).\ w \models \varphi$	CTL: state formulas existential path quantification $\exists\varphi$ universal path quantification: $\forall\varphi$
complexity of the model checking problems	PSPACE–complete $\mathcal{O}\left(\lvert TS\rvert \cdot 2^{\lvert\varphi\rvert} \cdot \lvert\varphi\rvert\right)$	PTIME $\mathcal{O}(\lvert TS\rvert \cdot \lvert\Phi\rvert)$

Computation Tree Logic

modal logic over infinite trees [Clarke & Emerson 1981]

- Formulas over states (capital greek letters)
  - $a \in AP$ — 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

$\Rightarrow$ note that X and U *alternate* with A and E
  - $\mathsf{AX\,X}\ \Phi$ and $\mathsf{A\,EX}\ \Phi \notin \mathsf{CTL}$, but $\mathsf{AX\,AX}\ \Phi$ and $\mathsf{AX\,EX}\ \Phi \in \mathsf{CTL}$

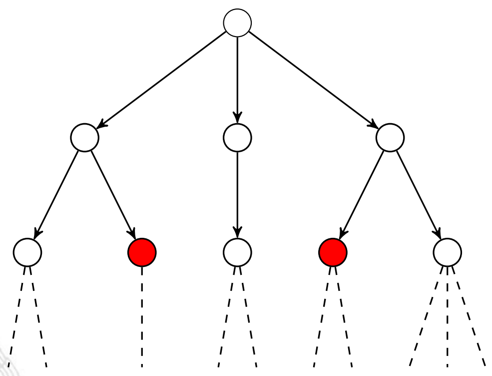
- Convention: Unary operators bind stronger than binary ones (e.g., $\neg a\,\mathsf{U}\,b \equiv (\neg a)\,\mathsf{U}\,b$)

Derived operators

potentially Φ:	$\mathsf{E\,F}\,\Phi$	$\equiv$	$\mathsf{E}\,(\mathsf{true}\,\mathsf{U}\,\Phi)$
inevitably Φ:	$\mathsf{A\,F}\,\Phi$	$\equiv$	$\mathsf{A}\,(\mathsf{true}\,\mathsf{U}\,\Phi)$
potentially always Φ:	$\mathsf{E\,G}\,\Phi$	$\equiv$	$\neg\mathsf{A\,F}\,\neg\Phi$
invariantly Φ:	$\mathsf{A\,G}\,\Phi$	$\equiv$	$\neg\mathsf{E\,F}\,\neg\Phi$

the boolean connectives are derived as usual

Visualization of semantics



E F red

Example properties in CTL

Semantics of CTL state-formulas

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

$$
\begin{array}{lll}
s \models a & \text{iff} & a \in L(s) \\
s \models \neg \Phi & \text{iff} & s \not\models \Phi \\
s \models \Phi \wedge \Psi & \text{iff} & s \models \Phi \text{ and } s \models \Psi \\
s \models \mathsf{E}\,\varphi & \text{iff} & \pi \models \varphi \text{ for } \textit{some} \text{ path } \pi \text{ that starts in } s \\
s \models \mathsf{A}\,\varphi & \text{iff} & \pi \models \varphi \text{ for } \textit{all} \text{ paths } \pi \text{ that start in } s
\end{array}
$$

Semantics of CTL path-formulas

Transition System Semantics

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

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

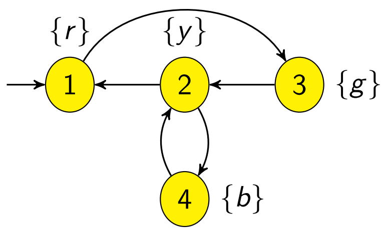
- TS satisfies CTL-formula Φ iff Φ holds in all its initial states:

$$TS \models \Phi \quad \text{if and only if} \quad \forall s_0 \in I.\, s_0 \models \Phi$$

  - this is equivalent to $I \subseteq Sat(\Phi)$
- Point of attention: $TS \not\models \Phi$ and $TS \not\models \neg\Phi$ is possible!
  - because of several initial states, e.g. $s_0 \models \mathrm{EG}\,\Phi$ and $s_0' \not\models \mathrm{EG}\,\Phi$

Exercises

- 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 $TS_2 \not\models \Phi$.
- Consider the following transition system which models a traffic light which can blink.



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

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