

## Introduction to Model Checking

René Thiemann

Institute of Computer Science  
University of Innsbruck

WS 2007/2008

### Last Lecture

## Outline

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

### Reduction via SAT

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

## Comparing LTL with NBAs

Recall Translation Theorem:

Theorem (Vardi, Wolper)

Every LTL formula  $\varphi$  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**

## 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**)

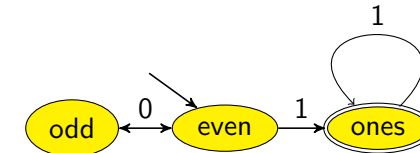
- a beverage will be delivered
- at every time there is a way to reach the main menu

## Comparing LTL with NBAs (2)

Consider the language

$$\mathcal{L} = \{w \in 2^\omega \mid w = 0\dots 01^\omega \text{ with an even number of 0's}\}$$

- $\mathcal{L}$  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  $\varphi$

Hence, NBAs are **strictly** more expressive than LTL

## Linear and branching temporal logic

- Linear** temporal logic:
  - "statements about **all paths** (starting in a state)"
  - $s \models G a$  iff for all possible paths starting in  $s$  always  $a$
- Branching** temporal logic:
  - "statements about **all or some paths** starting in a state"

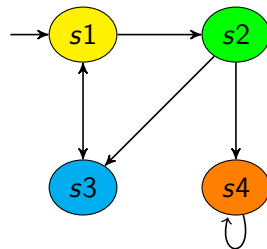
- $s \models AG a$  iff for **all** paths starting in  $s$  always  $a$
- $s \models EG a$  iff for **some** path starting in  $s$  always  $a$
- nesting of path quantifiers is allowed
- Checking  $s \models E\varphi$  in LTL can be done using  $s \not\models A\neg\varphi$ 
  - ... but this does not work for nested formulas such as  $AG EF a$

# 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  $\mu$ -calculus
- Modal  $\mu$ -calculus
- S2S
- ....

# Transition Systems and Trees



# 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

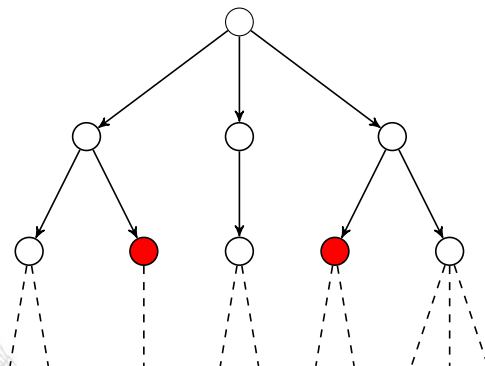
"behavior" in a state $s$	path-based: $trace(s)$	state-based: computation tree of $s$
temporal logic	LTL: path formulas $\varphi$ $s \models \varphi$ iff $\forall w \in Traces(s). w \models \varphi$	CTL: state formulas existential path quantification $\exists\varphi$ universal path quantification: $\forall\varphi$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

# 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
    - $E \varphi$  there *exists* a path fulfilling  $\varphi$
    - $A \varphi$  *all* paths fulfill  $\varphi$
  - **Formulas over paths** (lower case greek letters)
    - $X \Phi$  the next state fulfills  $\Phi$
    - $\Phi U \Psi$   $\Phi$  holds until a  $\Psi$ -state is reached
- ⇒ note that  $X$  and  $U$  *alternate* with  $A$  and  $E$
- $AXX \Phi$  and  $AEX \Phi \notin \text{CTL}$ , but  $AXAX \Phi$  and  $AXEX \Phi \in \text{CTL}$
  - Convention: Unary operators bind stronger than binary ones (e.g.,  $\neg a U b \equiv (\neg a) U b$ )

## Visualization of semantics



EF red

# Derived operators

**potentially**  $\Phi$ :  $EF \Phi \equiv E(\text{true} U \Phi)$

**inevitably**  $\Phi$ :  $AF \Phi \equiv A(\text{true} U \Phi)$

**potentially always**  $\Phi$ :  $EG \Phi \equiv \neg AF \neg \Phi$

**invariantly**  $\Phi$ :  $AG \Phi \equiv \neg EF \neg \Phi$

the boolean connectives are derived as usual

## Example properties in CTL

## Semantics of CTL **state**-formulas

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

- $s \models a$       iff  $a \in L(s)$
- $s \models \neg \Phi$     iff  $s \not\models \Phi$
- $s \models \Phi \wedge \Psi$  iff  $s \models \Phi$  and  $s \models \Psi$
- $s \models E\varphi$     iff  $\pi \models \varphi$  for **some** path  $\pi$  that starts in  $s$
- $s \models A\varphi$     iff  $\pi \models \varphi$  for **all** paths  $\pi$  that start in  $s$

## Transition System Semantics

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

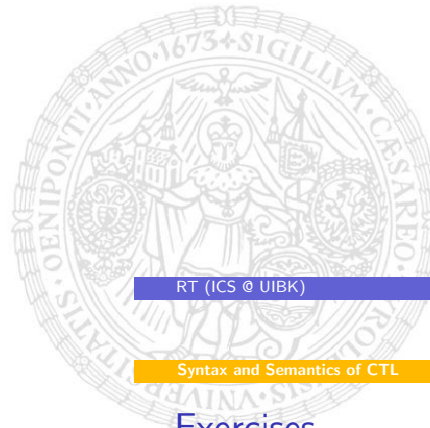
$$Sat(\Phi) = \{s \in S \mid s \models \Phi\}$$

- $TS$  satisfies CTL-formula  $\Phi$  iff  $\Phi$  holds in all its initial states:

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

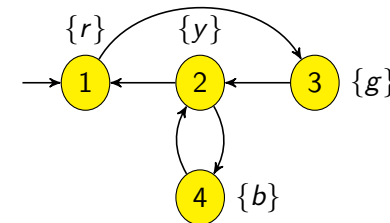
- this is equivalent to  $I \subseteq Sat(\Phi)$
- **Point of attention:**  $TS \models \Phi$  and  $TS \not\models \neg\Phi$  is possible!
- because of several initial states, e.g.  $s_0 \models EG\Phi$  and  $s'_0 \not\models EG\Phi$

## Semantics of CTL **path**-formulas



## Exercises

- Provide two transition systems  $TS_1, TS_2$  (same atomic propositions, every state has at least one outgoing edge) and a CTL-formula  $\Phi$  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.

- $AFy$
- $AGy$
- $AGAFy$
- $AFg$
- $EFg$
- $EGg$
- $A(\neg b U b)$
- $E(\neg b U b)$
- $AGAFb$