

## Introduction to Model Checking

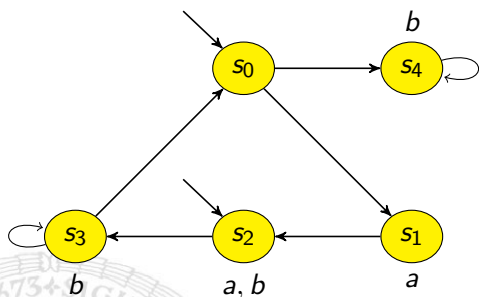
René Thiemann

Institute of Computer Science  
University of Innsbruck

WS 2007/2008

### Last Exercises

$$\Phi = A(aU b) \vee EX(AG b)$$



## Outline

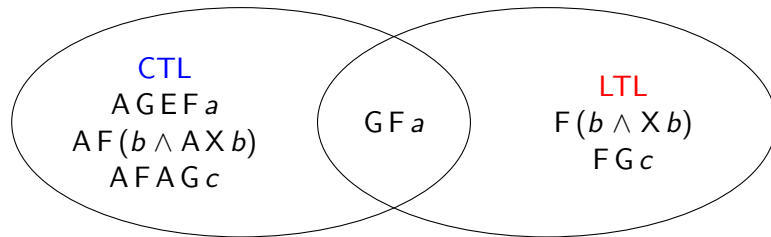
- Last Exercises
- Expressiveness of CTL and LTL
- CTL\*
- CTL\*-Model Checking

### Last Exercises

Provide a direct fixpoint characterization of  $Sat(EG \Phi)$  without using the following equivalence.

$$EG \Phi \equiv \neg AF \neg \Phi \equiv \neg A \text{ true } U \neg \Phi$$

## Expressiveness of CTL and LTL



## Theorem (Clarke, Draghicescu)

Let  $\Phi$  be a CTL-state-formula and  $\varphi$  the LTL-formula that is obtained by eliminating all path quantifiers in  $\Phi$ . Then:

$\Phi \equiv \varphi$  or there does not exist any LTL-formula that is equivalent to  $\Phi$ .

Hence, to prove that AFAG  $c$  is not LTL-expressible it suffices to show that AFAG  $a$  and FG  $c$  are not equivalent.

## CTL\*

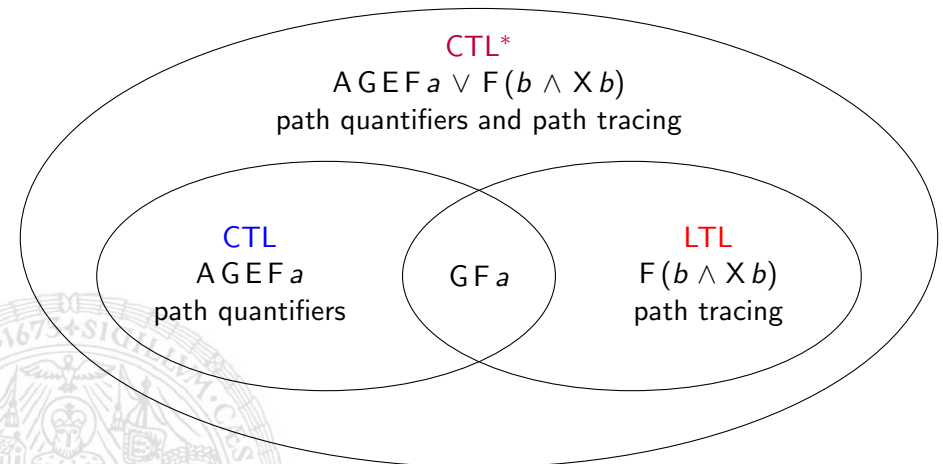
- Formulas over states (capital greek letters)

- true
- $a \in AP$  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  $\varphi$  in the next moment  $\varphi$  holds
- $\varphi U \psi$   $\varphi$  holds until  $\psi$
- $\neg \varphi$  and  $\varphi \wedge \psi$  negation and conjunction
- $\Phi$  the current state satisfies  $\Phi$

## Expressiveness of CTL and LTL



## Semantics of CTL\*

A state-formula  $\Phi$  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  $s \models \Psi$
- $s \models E \varphi$  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  $s$

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

- $\pi \models X \varphi$  iff  $\pi[1..] \models \varphi$
- $\pi \models \varphi U \psi$  iff  $(\exists j \geq 0. \pi[j..] \models \psi \text{ and } (\forall 0 \leq k < j. \pi[k..] \models \varphi))$
- $\pi \models \varphi \wedge \psi$  iff  $\pi \models \varphi$  and  $\pi \models \psi$
- $\pi \models \neg \varphi$  iff  $\pi \not\models \varphi$
- $\pi \models \Phi$  iff  $\pi[0] \models \Phi$

## Derived Operators

As usual one can use the following shortcuts:

$$F\varphi \equiv \text{true } U \varphi$$

$$G\varphi \equiv \neg F \neg \varphi$$

## Example

On all paths it is infinitely often **serv**ed and there always is a possibility to get back to the **main** menu

## Transition System Semantics for CTL\*

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

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

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

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

## Embedding LTL in CTL\*

Let  $\varphi$  be an LTL-formula. Then

$$TS \models \varphi \text{ (LTL)} \quad \text{iff} \quad TS \models A\varphi \text{ (CTL*)}$$

## CTL\*-Model Checking Algorithm [Emerson, Lei]

## Eliminating Existential Path Quantifiers

## Lemma

For every path-formula  $\varphi$  the following equivalence is valid:

$$E\varphi \equiv \neg A\neg\varphi$$

## Proof.

- $s \models E\varphi$
- iff there is a path  $\pi$  starting in  $s$  such that  $\pi \models \varphi$
- iff it is not the case that there is no path  $\pi$  starting in  $s$  with  $\pi \models \varphi$
- iff it is not the case that all paths  $\pi$  starting in  $s$  violate  $\pi \models \varphi$
- iff it is not the case that all paths  $\pi$  starting in  $s$  satisfy  $\pi \models \neg\varphi$
- iff it is not the case that  $s \models A\neg\varphi$
- iff  $s \models \neg A\neg\varphi$

■

## Use LTL-Model Checker for Universal Formulas

- $s \in \text{Sat}(A\varphi)$  iff all paths  $\pi$  starting in  $s$  satisfy  $\pi \models \varphi$
  - Essentially,  $\varphi$  is LTL-formula but may contain CTL\*-state-formulas
- ⇒ LTL-model checker not directly applicable

## Solution

- States which satisfy contained CTL\*-state formulas are known
- ⇒
- Replace every maximal state-formula  $\Psi$  in  $\varphi$  which is not an atomic proposition by a new atomic proposition  $a_\Psi$ , result: LTL-formula  $\varphi'$
  - Extend labeling of states: Whenever  $s \in \text{Sat}(\Psi)$  then add  $a_\Psi$  to the set of labels of  $s$ .
  - Afterwards apply LTL-model checker to determine  $\text{Sat}(A\varphi)$ :

$$s \in \text{Sat}(A\varphi) \text{ iff } s \models \varphi'$$

## Use Bottom-Up CTL-Model Checking Procedure

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

Compute sets  $\text{Sat}(\cdot)$  for state-formulas in a bottom-up way:

- $\text{Sat}(\text{true}) = S$
- $\text{Sat}(a) = \{s \mid a \in L(s)\}$
- $\text{Sat}(\neg\Phi) = S \setminus \text{Sat}(\Phi)$
- $\text{Sat}(\Phi \wedge \Psi) = \text{Sat}(\Phi) \cap \text{Sat}(\Psi)$

## Example

## Comparison

| Formalism                        | LTL                    | CTL   | CTL*   |
|----------------------------------|------------------------|---|--|
| MC-algorithm                     | NBAs                   | <i>Sat</i> -computation<br>(set operations) | <i>Sat</i> -computation<br>(set ops. + NBAs) |
| MC-complexity<br>(fixed formula) | PSPACE-comp.<br>linear | linear<br>linear                            | PSPACE-comp.<br>linear                       |

## Exercises

- Prove that there is no LTL-formula which is equivalent to  $AF(b \wedge AX b)$
- Consider the formula  $\Phi = AG((\neg EF \textit{serve}) \vee (EG F \textit{main}))$ 
  - Try to formulate the meaning in words
  - Apply the CTL\*-model checking algorithm on the following example. Do not construct the NBAs for the LTL-model checking, but do LTL-model checking intuitively.

