Introduction to Model Checking

René Thiemann

Institute of Computer Science

University of Innsbruck

WS 2007/2008

Outline

- Last Lecture
- "Model Checking" Lecture
- Repetition
  - NanoPromela to Transition Systems
  - LTL to NBA

CTL*-Model Checking Algorithm [Emerson, Lei]

1. Eliminate existential path quantifiers

$$\mathsf{E}\,\varphi \equiv \neg\mathsf{A}\,\neg\varphi$$

2. Use bottom-up CTL-model checking procedure
   - $Sat(\mathrm{true}) = S$
   - $Sat(a) = \{s \mid a \in L(s)\}$
   - $Sat(\neg\,\Phi) = S \setminus Sat(\Phi)$
   - $Sat(\Phi \wedge \Psi) = Sat(\Phi) \cap Sat(\Psi)$
3. Integrate LTL-model checker for universal path formulas

Use LTL-Model Checker for Universal Formulas

Idea: to compute $s \in Sat(\mathsf{A}\,\varphi)$ perform LTL model checking of φ

- Replace every maximal state-formula Ψ in φ which is not an atomic proposition by a new atomic proposition a_Ψ, result: LTL-formula φ'
- Extend labeling of states: Whenever $s \in Sat(\Psi)$ then add a_Ψ to the set of labels of s.
- Apply LTL-model checker to determine $Sat(\mathsf{A}\,\varphi)$:

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

Exercise 1

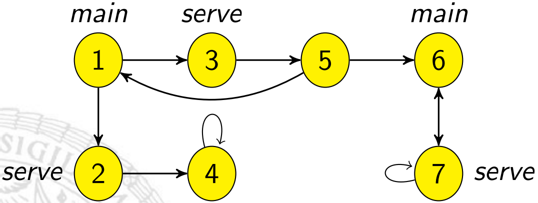
Prove that there is no LTL-formula which is equivalent to
$\Phi = \mathsf{A}\,\mathsf{F}\,(b \wedge \mathsf{A}\,\mathsf{X}\,b)$

Exercise 2

Consider the formula $\Phi = \mathsf{A}\,\mathsf{G}\,((\neg \mathsf{E}\,\mathsf{F}\, \mathit{serve}) \vee (\mathsf{E}\,\mathsf{G}\,\mathsf{F}\, \mathit{main}))$

- Try to formulate the meaning in words

- Apply the CTL*-model checking algorithm on the following example



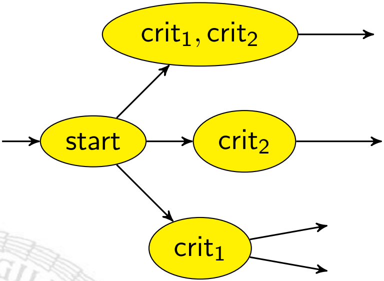
Selection of Topics

- Model checking on the fly
- μ-calculus
- S1S
- Model checking of real-time systems
- Controlling the state-space explosion problem
- ...

Model Checking on the Fly

Consider $\varphi = \mathsf{G}\,\neg(\mathsf{crit}_1 \wedge \mathsf{crit}_2)$

Transition system:



. . . 5,000,000 other states

Full construction of $TS \otimes \mathcal{A}_{\neg\varphi}$ to check emptyness is too expensive
$\Rightarrow$ only construct part of TS which suffices to check $TS \models \varphi$

Model Checking on the Fly

μ-Calculus

In CTL: semantics based on least and greatest fixpoint

In μ-calculus:

- explicit least- and greatest fixpoint operators
- easy to implement
- many logics can be translated into μ-calculus
- parallel model checking algorithms available

$\Rightarrow$ μ-calculus as efficient basis for model-checking for several logics

S1S

Consider the following property:

Between every green and red phase there is at least one orange phase.

Formulating these kinds of properties in LTL is doable, but not intuitive

$$\mathsf{G}\,(\mathsf{red} \Rightarrow \mathsf{X}\,(\mathsf{G}\,\neg\mathsf{green}) \vee (\neg\mathsf{green} \wedge (\mathsf{X}\,\neg\mathsf{green}\,\mathsf{U}\,\mathsf{orange})))$$

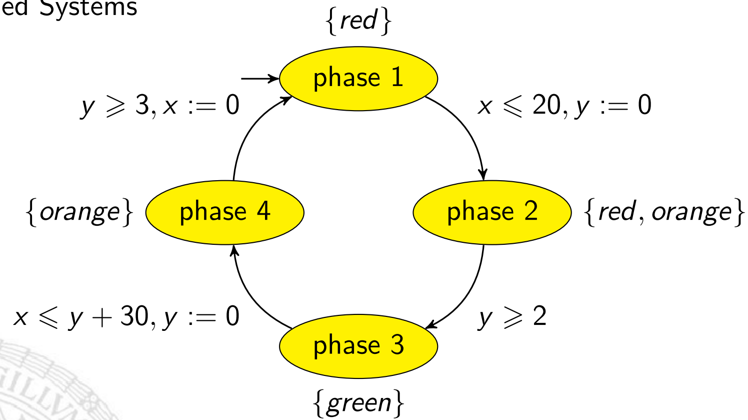
Use S1S instead:

$$\forall t_1, t_2 : (t_1 < t_2 \wedge \mathsf{green}(\mathsf{t}_1) \wedge \mathsf{red}(\mathsf{t}_2)) \Rightarrow \exists t_3 : t_1 < t_3 < t_2 \wedge \mathsf{orange}(\mathsf{t}_3)$$

- Allows readable and succinct specifications
- One can perform model checking by NBAs

Model Checking of Real-Time Systems

- Timed Systems



- Timed Specifications

  One does not have to wait more than 50 seconds for green:

$$\Phi = \mathsf{G}\,\mathsf{F}^{\leqslant 50}\text{green}$$

Controlling the State-Space Explosion Problem

Reduce search space in various ways

- Abstraction:
  instead of 16-bit integer, only distinguish between even and odd, or between positive, 0, negative, or between . . .
- Partial order reduction:
  if process 1 and process 2 perform operations on local variables, then schedule process 1 always before process 2

$\Rightarrow$ less interleaving, smaller transition system

- . . .

NanoPromela to Channel System: Locations = Sub-Expr.

$$\frac{}{x := \mathsf{expr} \xrightarrow{\mathsf{true} : \mathsf{assign}(x, \mathsf{expr})} \mathsf{exit}}$$

$$\frac{\mathsf{stmt}_1 \xrightarrow{g:\alpha} \mathsf{stmt}_1' \neq \mathsf{exit}}{\mathsf{stmt}_1; \mathsf{stmt}_2 \xrightarrow{g:\alpha} \mathsf{stmt}_1'; \mathsf{stmt}_2}$$

$$\frac{\mathsf{stmt}_1 \xrightarrow{g:\alpha} \mathsf{exit}}{\mathsf{stmt}_1; \mathsf{stmt}_2 \xrightarrow{g:\alpha} \mathsf{stmt}_2}$$

$$\frac{\mathsf{stmt}_i \xrightarrow{h:\alpha} \mathsf{stmt}_i'}{\mathbf{if} \ldots :: g_i \Rightarrow \mathsf{stmt}_i \ldots \mathbf{fi} \xrightarrow{g_i \wedge h:\alpha} \mathsf{stmt}_i'}$$

$$\frac{\mathsf{stmt}_i \xrightarrow{h:\alpha} \mathsf{stmt}_i' \neq \mathsf{exit}}{\mathbf{do} \ldots :: g_i \Rightarrow stmt_i \ldots \mathbf{od} \xrightarrow{g_i \wedge h:\alpha} \mathsf{stmt}_i'; \mathbf{do} \ldots \mathbf{od}}$$

$$\frac{\mathsf{stmt}_i \xrightarrow{h:\alpha} \mathsf{exit}}{\mathbf{do} \ldots :: g_i \Rightarrow stmt_i \ldots \mathbf{od} \xrightarrow{g_i \wedge h:\alpha} \mathbf{do} \ldots \mathbf{od}}$$

Example: Mutual Exclusion of 2 Processes

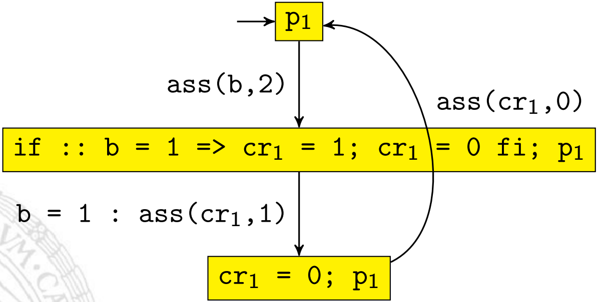
p_1 `= do :: true => b = 2; if :: b = 1 =>` cr_1 `= 1;` cr_1 `= 0 fi od`
p_2 `= do :: true => b = 1; if :: b = 2 =>` cr_2 `= 1;` cr_2 `= 0 fi od`

Example: Mutual Exclusion of 2 Processes

p_1 = `do :: true => b = 2; if :: b = 1 => cr`$_1$` = 1; cr`$_1$` = 0 fi od`

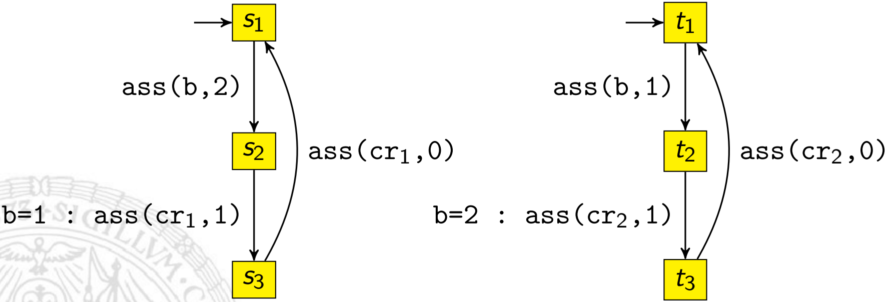
p_2 = `do :: true => b = 1; if :: b = 2 => cr`$_2$` = 1; cr`$_2$` = 0 fi od`

For p_1 obtain:

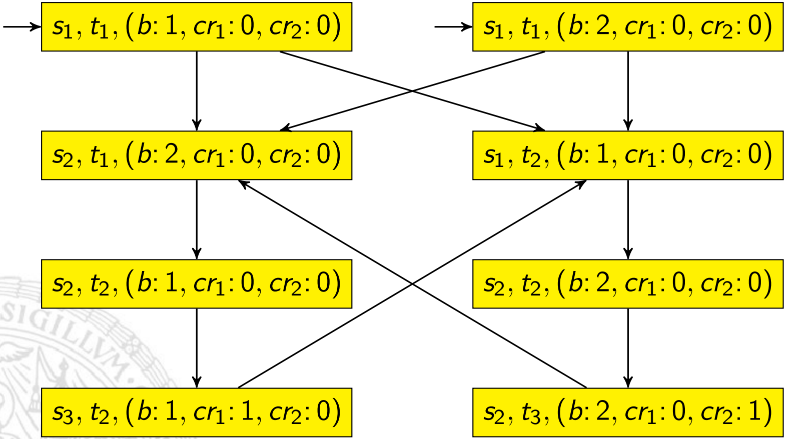


Example: Resulting Channel System

Renaming states of $\mathtt{p}_1$ to $\mathtt{s}_1,\mathtt{s}_2,\mathtt{s}_3$ and those of $\mathtt{p}_2$ to $\mathtt{t}_1,\mathtt{t}_2,\mathtt{t}_3$ yields the following channel system



Example: Resulting Transition System



LTL to NBAs: φ-Expansion and Consistency Checks

$$
\begin{array}{lll}
\varphi_j = \neg\varphi_{j_1} & \Rightarrow w[i..]^j = 1 \text{ iff} & w[i..]^{j_1} = 0 \\
\varphi_j = \varphi_{j_1} \wedge \varphi_{j_2} & \Rightarrow w[i..]^j = 1 \text{ iff} & w[i..]^{j_1} = 1 \text{ and } w[i..]^{j_2} = 1 \\
\varphi_j = \mathsf{X}\,\varphi_{j_1} & \Rightarrow w[i..]^j = 1 \text{ iff} & w[i+1..]^{j_1} = 1 \\
\varphi_j = \varphi_{j_1} \,\mathsf{U}\, \varphi_{j_2} & \Rightarrow w[i..]^j = 1 \text{ iff} & w[i..]^{j_2} = 1 \text{ or} \\
 & & (w[i..]^{j_1} = 1 \text{ and } w[i+1]^j = 1)
\end{array}
$$

States of NBA: Vectors where components represent sub-formulas

Transitions $s_1 \xrightarrow{in} s_2$ where

- s_1: preceding values of φ-expansion or start state q_0
- in: current input
- s_2: current values of φ-expansion

Transitions satisfy consistency checks and q_0 leads to states with φ-component 1

Example $\varphi = \mathsf{X}\,(a\,\mathsf{U}\,b)$