SS 2017
Software Verification
LTL monitoring

Prof. Dr. Holger Schlingloff \textsuperscript{1,2}
Dr. Esteban Pavese \textsuperscript{1}

(1) Institut für Informatik der Humboldt Universität
(2) Fraunhofer Institut für offene Kommunikationssysteme FOKUS
Recap

• What is an \( \omega \)-regular language?
• Can you give an \( \omega \)-regular expression for the following language over the alphabet \( \{a,b\} \):
  - finitely many a’s
  - infinitely many a’s
  - between any two adjacent a’s there is a b
• What is a Büchi-automaton?
• Give Büchi-automata for the above languages!
• Are Büchi-automata closed under union, intersection, concatenation, complementation, determinisation?
• Is emptyness of Büchi-automata decidable?
• What is a Muller-automaton?
Fairness in LTL and CTL

- As we have seen, LTL can express reactivity properties ("infinitely many requests lead to infinitely many replies")
- Thus, LTL can encode Büchi acceptance conditions (with additional propositions): \((FG s_1 \lor \ldots \lor FG s_n)\)
- Thus, a model checker for LTL on Kripke structures can be used to check LTL on Büchi automata
  - again, modulo state/transition markings
- CTL can not express fairness
- However, often we would like to restrict attention to fair paths only
- Model checking CTL on (computation trees of) Büchi automata?
Fairness in nuSMV

- A *fairness constraint* restricts the attention only to *fair execution paths*. When evaluating specifications, the model checker considers path quantifiers to apply only to fair paths.

- NuSMV supports two types of fairness constraints, namely justice constraints and compassion constraints.
  - A *justice constraint* consists of a formula \( f \) which is assumed to be true infinitely often in all the fair paths. In NuSMV justice constraints are identified by keywords JUSTICE and, for backward compatibility, FAIRNESS.
  - A *compassion constraint* consists of a pair of formulas \( (p,q) \); if property \( p \) is true infinitely often in a fair path, then also formula \( q \) has to be true infinitely often in the fair path. In NuSMV compassion constraints are identified by keyword COMPASSION.(3)

- Fairness constraints are declared using the following syntax:
  - fairness_declaration :: "FAIRNESS" simple_expr [";"] | "JUSTICE" simple_expr [";" ] | "COMPASSION" "(" simple_expr "," simple_expr ")" [";" ]
  - “running” is a shortcut for all transitions in the current module
  - A path is considered fair if and only if it satisfies all the constraints declared in this manner.
A Hilbert-style axiom system for LTL

- **(PL)** all propositionally valid formulas
- **(K)** ⊨ (\(X(\phi \rightarrow \psi) \rightarrow (X\phi \rightarrow X\psi)\))
- **(U)** ⊨ (\(X \neg \phi \rightarrow \neg X \phi\))
- **(D)** ⊨ (\(\neg X \phi \rightarrow X \neg \phi\))
- **(Rec)** ⊨ (\(X(\psi \lor \phi \land (\phi U^+ \psi)) \rightarrow (\phi U^+ \psi)\))
- **(N)** \(\phi \vdash X \phi\)
- **(MP)** \(\phi, (\phi \rightarrow \psi) \vdash \psi\)
- **(Ind)** (\(X(\psi \lor (\phi \land \xi)) \rightarrow \xi\)) ⊨ ((\(\phi U^+ \psi\) → \(\xi\))
- **(Rec)** ⊨ ((\(\phi W^+ \psi\) → \(X (\psi \lor \phi \land (\phi W^+ \psi))\))
- **(Ind)** (\(\xi \rightarrow X(\psi \lor (\phi \land \xi))\)) ⊨ (\(\xi \rightarrow (\phi W^+ \psi)\))
- **(Rec)** ⊨ (\(G^+ \phi \rightarrow X(\phi \land G^+ \phi)\))
- **(Ind)** (\(\xi \rightarrow X(\phi \land \xi)\)) ⊨ (\(\xi \rightarrow G^+ \phi\))
Valid (Derivable) LTL Properties

- $\vdash (X\varphi \to X\psi) \iff X(\varphi \to \psi)$
- $\vdash (G^+(\varphi \to \psi) \to (G^+\varphi \to G^+\psi))$
- $\vdash ((F^+\varphi \to F^+\psi) \to F^+(\varphi \to \psi))$
- $\vdash ((G^+\varphi \land F^+\psi) \to F^+(\varphi \land \psi))$
- $\vdash ((G^+F^+\varphi \land G^+F^+\psi) \iff G^+F^+(\varphi \land F^+\psi))$
- $\vdash (G^+(\varphi \to X^+\varphi) \to (\varphi \to G^+\varphi))$
- $\vdash (F^+\varphi \iff X(\varphi \lor F^+\varphi)) \quad \vdash (F^+\varphi \iff (X\varphi \lor XF^+\varphi))$
- $\vdash (G^+\varphi \iff X(\varphi \land G^+\varphi)) \quad \vdash (G^+\varphi \iff (X\varphi \land XG^+\varphi))$
- $\vdash ((\varphi U^+\psi) \iff X(\psi \lor (\varphi \land (\varphi U^+\psi))))$
- $\vdash ((\varphi W^+\psi) \iff X(\psi \lor (\varphi \land (\varphi W^+\psi))))$
Model checking

• Given a model $M$ and a formula $\varphi$, model checking is answering the question whether $M \models \varphi$
  ▪ somewhat easier than checking validity or satisfiability of $\varphi$
  ▪ usually easier than checking $\models \varphi_M \rightarrow \varphi$
  ▪ sometimes easier than checking $L(M) \subseteq L(\varphi)$ or $M \in L(\varphi)$

• Several variants, depending on the logic and the way the model is given
  ▪ e.g., consider $\text{PL}$ and a lookup truth table for propositions
    $\Rightarrow$ linear in $|\varphi|$ 
  ▪ e.g., consider $\text{FOL}$ and a „computation engine“ for predicates
    $\Rightarrow$ in general model checking is an undecidable problem

• Here, $\text{LTL}$ and $\text{CTL}$ are of interest
We want to check whether $M \models \varphi$

- $\varphi$ is a LTL formula (for simplicity, excluding past)
- $M$ is a natural model or sequence of proposition interpretations
  - if $M$ is finite, then the problem is easy (exercise!)
    - $M \models \varphi$ iff $\text{check}(M,0,\varphi) = \text{true}$
    - $\text{check}(M,i,p) = \text{true}$ iff $p \in M(i)$
    - $\text{check}(M,i, \bot) = \text{false}$
    - $\text{check}(M,i, (\varphi \rightarrow \psi)) = \text{true}$ iff $\text{check}(M,i,\varphi)$ implies $\text{check}(M,i, \psi)$
    - $\text{check}(M,i, (\varphi U^+ \psi)) = \text{true}$ iff for some $j>i$, $\text{check}(M,j,\psi) = \text{true}$ and for all $i<k<j$, $\text{check}(M,k,\varphi) = \text{true}$
  - better:
    - $\text{check}(M,i, \varphi U^+ \psi) = i+1<|M|$ and $\text{check}(M,i+1, (\psi \lor \varphi \land \varphi U^+ \psi))$