LTL Model Checking
Lecture #15 of Model Checking

Joost-Pieter Katoen
Lehrstuhl 2: Software Modeling & Verification

E-mail: katoen@cs.rwth-aachen.de

May 23, 2007
Overview Lecture #15

⇒ Repetition: LTL and GNBA

• From LTL to GNBA
Recall: Linear Temporal Logic

modal logic over infinite sequences [Pnueli 1977]

• Propositional logic
  - \( a \in AP \)
  - \( \neg \varphi \) and \( \varphi \land \psi \)

• Temporal operators
  - \( \Box \varphi \)
  - \( \varphi U \psi \)
    - \( \varphi \) holds Until a \( \psi \)-state is reached

• Auxiliary temporal operators
  - \( \Diamond \varphi \equiv \text{true} U \varphi \)
  - \( \square \varphi \equiv \neg \Diamond \neg \varphi \)
    - eventually \( \varphi \)
    - always \( \varphi \)
LTL model-checking problem

The following decision problem:

Given finite transition system $TS$ and LTL-formula $\varphi$:
yields “yes” if $TS \models \varphi$, and “no” (plus a counterexample) if $TS \not\models \varphi$
NBA for LTL-formulae
A first attempt

$TS \models \varphi$ if and only if

\[ \text{Traces}(TS) \subseteq \underbrace{\text{Words}(\varphi)}_{\mathcal{L}_\omega(A_\varphi)} \]

if and only if

\[ \text{Traces}(TS) \cap \mathcal{L}_\omega(\overline{A_\varphi}) = \emptyset \]

but complementation of NBA is quadratically exponential

if $A$ has $n$ states, $\overline{A}$ has $c^{n^2}$ states in worst case

use the fact that $\mathcal{L}_\omega(\overline{A_\varphi}) = \mathcal{L}_\omega(A_{\neg \varphi})$!
Observation

\[ TS \models \varphi \quad \text{if and only if} \quad \text{Traces}(TS) \subseteq \text{Words}(\varphi) \]

\[ \text{if and only if} \quad \text{Traces}(TS) \cap (2^{AP})^\omega \setminus \text{Words}(\varphi) = \emptyset \]

\[ \text{if and only if} \quad \text{Traces}(TS) \cap \text{Words}(\neg \varphi) = \emptyset \]

\[ \text{if and only if} \quad TS \otimes A_{\neg \varphi} \models \square \Diamond \neg F \]

\textit{LTL model checking is thus reduced to persistence checking!}
Overview of LTL model checking

System

Model of system

Negation of property

LTL-formula \( \neg \varphi \)

model checker

Transition system \( TS \)

Product transition system \( TS \otimes A_{\neg \varphi} \)

Generalised Büchi automaton \( G_{\neg \varphi} \)

Büchi automaton \( A_{\neg \varphi} \)

\( TS \otimes A_{\neg \varphi} \models P_{\text{pers}}(A_{\neg \varphi}) \)

‘Yes’

‘No’ (counter-example)
Recall: Generalized Büchi automata

For the purposes of this monograph, it suffices to consider a slight variant of nondeterministic Büchi automata, called generalized nondeterministic Büchi automata, GNBA for short. The difference between NBA and GNBA is that the acceptance condition for GNBA requires to visit several sets $F_1, \ldots, F_k$ infinitely often. Formally, the syntax of GNBA is as for NBA, except that the acceptance condition is a set $\mathcal{F}$ consisting of finitely many acceptance sets $F_1, \ldots, F_k$ with $F_i \subseteq Q$. That is, if $Q$ is the state space of the automaton then the acceptance condition of an GNBA is an element $\mathcal{F}$ of $2^{2^Q}$. Recall that for NBA, it is an element $F \in 2^Q$. The accepted language of a GNBA $\mathcal{G}$ consists of all infinite words which have an infinite run in $\mathcal{G}$ that visits all sets $F_i \in \mathcal{F}$ infinitely often. Thus, the acceptance criterion in a generalized Büchi automaton can be understood as the conjunction of a number of Büchi acceptance conditions.
Recall: Generalized Büchi automata

A generalized NBA (GNBA) $G$ is a tuple $(Q, \Sigma, \delta, Q_0, \mathcal{F})$ where:

- $Q$ is a finite set of states with $Q_0 \subseteq Q$ a set of initial states
- $\Sigma$ is an alphabet
- $\delta : Q \times \Sigma \rightarrow 2^Q$ is a transition function
- $\mathcal{F} = \{ F_1, \ldots, F_k \}$ is a (possibly empty) subset of $2^Q$

The size of $G$, denoted $|G|$, is the number of states and transitions in $G$:

$$|G| = |Q| + \sum_{q \in Q} \sum_{A \in \Sigma} |\delta(q, A)|$$
Recall: Language of a GNBA

- GNBA $G = (Q, \Sigma, \delta, Q_0, F)$ and word $\sigma = A_0A_1A_2 \ldots \in \Sigma^\omega$

- A run for $\sigma$ in $G$ is an infinite sequence $q_0 q_1 q_2 \ldots$ such that:
  - $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for all $0 \leq i$

- Run $q_0 q_1 \ldots$ is accepting if for all $F \in F$: $q_i \in F$ for infinitely many $i$

- $\sigma \in \Sigma^\omega$ is accepted by $G$ if there exists an accepting run for $\sigma$

- The accepted language of $G$:

$$L_\omega(G) = \{ \sigma \in \Sigma^\omega \mid \text{there exists an accepting run for } \sigma \text{ in } G \}$$
Recall: From GNBA to NBA

For any GNBA $G$ there exists an NBA $A$ with:

$$L_\omega(G) = L_\omega(A) \quad \text{and} \quad |A| = \mathcal{O}(|G| \cdot |F|)$$

where $F$ denotes the set of acceptance sets in $G$

- Sketch of transformation GNBA (with $k$ accept sets) into equivalent NBA:
  - make $k$ copies of the automaton
  - initial states of NBA := the initial states in the first copy
  - final states of NBA := accept set $F_1$ in the first copy
  - on visiting in $i$-th copy a state in $F_i$, then move to the $(i+1)$-st copy
Overview Lecture #15

- Repetition: LTL and GNBA

⇒ From LTL to GNBA
From LTL to GNBA

GNBA $G_\varphi$ over $2^{AP}$ for LTL-formula $\varphi$ with $L_\omega(G_\varphi) = \text{Words}(\varphi)$:

- Assume $\varphi$ only contains the operators $\land$, $\lnot$, $\bigcirc$ and $U$
  - $\lor$, $\rightarrow$, $\lozenge$, $\square$, $W$, and so on, are expressed in terms of these basic operators
- States are *elementary sets* of sub-formulas in $\varphi$
  - for $\sigma = A_0A_1A_2 \ldots \in \text{Words}(\varphi)$, expand $A_i \subseteq AP$ with sub-formulas of $\varphi$
  - $\ldots$ to obtain the infinite word $\sigma = B_0B_1B_2 \ldots$ such that

$$\psi \in B_i \quad \text{if and only if} \quad \sigma^i = A_iA_{i+1}A_{i+2} \ldots \models \psi$$

- $\bar{\sigma}$ is intended to be a run in GNBA $G_\varphi$ for $\sigma$
- Transitions are derived from semantics $\bigcirc$ and expansion law for $U$
- Accept sets guarantee that: $\bar{\sigma}$ is an accepting run for $\sigma$ iff $\sigma \models \varphi$
From LTL to GNBA: the states (example)

- Let $\varphi = a \mathbin{U} (\neg a \land b)$ and $\sigma = \{ a \} \{ a, b \} \{ b \} \ldots$
  - $B_i$ is a subset of $\{ a, b, \neg a \land b, \varphi \} \cup \{ \neg a, \neg b, \neg (\neg a \land b), \neg \varphi \}$
  - this set of formulas is also called the closure of $\varphi$

- Extend $A_0 = \{ a \}$, $A_1 = \{ a, b \}$, $A_2 = \{ b \}$, $\ldots$ as follows:
  - extend $A_0$ with $\neg b$, $\neg (\neg a \land b)$, and $\varphi$ as they hold in $\sigma^0 = \sigma$ (and no others)
  - extend $A_1$ with $\neg (\neg a \land b)$ and $\varphi$ as they hold in $\sigma^1$ (and no others)
  - extend $A_2$ with $\neg a$, $\neg a \land b$ and $\varphi$ as they hold in $\sigma^2$ (and no others)
  - $\ldots$ and so forth
  - this is not effective and is performed on the automaton (not on words)

- Result:
  - $\bar{\sigma} = \{ a, \neg b, \neg (\neg a \land b), \varphi \} \{ a, b, \neg (\neg a \land b), \varphi \} \{ \neg a, b, \neg a \land b, \varphi \} \ldots$
Closure

For LTL-formula $\varphi$, the set $\text{closure}(\varphi)$ consists of all sub-formulas $\psi$ of $\varphi$ and their negation $\neg\psi$

(where $\psi$ and $\neg\neg\psi$ are identified)

for $\varphi = a \cup (\neg a \land b)$, $\text{closure}(\varphi) = \{ a, b, \neg a, \neg b, \neg a \land b, \neg(\neg a \land b), \varphi, \neg \varphi \}$

can we take $B_i$ as any subset of $\text{closure}(\varphi)$? no! they must be elementary
Elementary sets of formulae

$B \subseteq \text{closure}(\varphi)$ is elementary if:

1. $B$ is logically consistent if for all $\varphi_1 \land \varphi_2, \psi \in \text{closure}(\varphi)$:
   - $\varphi_1 \land \varphi_2 \in B \iff \varphi_1 \in B$ and $\varphi_2 \in B$
   - $\psi \in B \Rightarrow \neg \psi \notin B$
   - true $\in \text{closure}(\varphi) \Rightarrow$ true $\in B$

2. $B$ is locally consistent if for all $\varphi_1 \lor \varphi_2 \in \text{closure}(\varphi)$:
   - $\varphi_2 \in B \Rightarrow \varphi_1 \lor \varphi_2 \in B$
   - $\varphi_1 \lor \varphi_2 \in B$ and $\varphi_2 \notin B \Rightarrow \varphi_1 \in B$

3. $B$ is maximal, i.e., for all $\psi \in \text{closure}(\varphi)$:
   - $\psi \notin B \Rightarrow \neg \psi \in B$
Examples
The GNBA of LTL-formula $\varphi$

For LTL-formula $\varphi$, let $G_\varphi = (Q, 2^{AP}, \delta, Q_0, F)$ where

- $Q$ is the set of all elementary sets of formulas $B \subseteq \text{closure}(\varphi)$
  - $Q_0 = \{ B \in Q \mid \varphi \in B \}$

- $F = \{ \{ B \in Q \mid \varphi_1 \cup \varphi_2 \notin B \text{ or } \varphi_2 \in B \} \mid \varphi_1 \cup \varphi_2 \in \text{closure}(\varphi) \}$

- The transition relation $\delta : Q \times 2^{AP} \to 2^Q$ is given by:
  - $\delta(B, B \cap AP)$ is the set of all elementary sets of formulas $B'$ satisfying:
    (i) For every $\Box \psi \in \text{closure}(\varphi)$: $\Box \psi \in B \iff \psi \in B'$, and
    (ii) For every $\varphi_1 \cup \varphi_2 \in \text{closure}(\varphi)$:

$$
\varphi_1 \cup \varphi_2 \in B \iff \left( \varphi_2 \in B \lor (\varphi_1 \in B \land \varphi_1 \cup \varphi_2 \in B') \right)
$$
GNBA for LTL-formula $\Diamond a$
GNBA for LTL-formula $a \cup b$

$B_1 \{a, b, a \cup b\}$

$B_2 \{\neg a, b, a \cup b\}$

$B_3 \{a, \neg b, a \cup b\}$

$B_4 \{\neg a, \neg b, \neg(a \cup b)\}$

$B_5 \{a, \neg b, \neg(a \cup b)\}$
Main result

For any LTL-formula $\varphi$ (over $AP$) there exists a GNBA $G_\varphi$ over $2^{AP}$ such that:

(a) $\text{Words}(\varphi) = L_\omega(G_\varphi)$

(b) $G_\varphi$ can be constructed in time and space $O \left(2^{\|\varphi\|}\right)$

(c) \#accepting sets of $G_\varphi$ is bounded above by $O(|\varphi|)$

$\Rightarrow$ every LTL-formula expresses an $\omega$-regular property!
Proof
NBA are more expressive than LTL

There is no LTL formula \( \varphi \) with \( \text{Words}(\varphi) = P \) for the LT-property:

\[
P = \left\{ A_0 A_1 A_2 \ldots \in (2^\{\{a\}\})^\omega \mid a \in A_{2i} \text{ for } i \geq 0 \right\}
\]

But there exists an NBA \( \mathcal{A} \) with \( \mathcal{L}_\omega(\mathcal{A}) = P \)

\( \Rightarrow \) there are \( \omega \)-regular properties that cannot be expressed in LTL!