Complexity and Correctness of LTL Model Checking

Lecture #16 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

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

June 5, 2007
Overview Lecture #16

⇒ Repetition: from LTL to GNBA

• Correctness proof

• Complexity results
  – LTL model checking is coNP-hard and PSPACE-complete
  – Satisfiability and validity are PSPACE-hard

• Summary of LTL model checking
Reduction to persistence checking

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

if and only if \( \text{Traces}(TS) \cap (\left(2^\mathcal{AP}\right) \omega \setminus \text{Words}(\varphi)) = \emptyset \)

if and only if \( \text{Traces}(TS) \cap \underbrace{\text{Words}(\neg \varphi)}_{\mathcal{L}_\omega(\mathcal{A}_{\neg \varphi})} = \emptyset \)

if and only if \( TS \otimes \mathcal{A}_{\neg \varphi} \models \Diamond \Box \neg F \)

*LTL model checking is thus reduced to persistence checking!*
Overview of LTL model checking

System

Model of system

Negation of property

LTL-formula $\neg \phi$

model checker

Transition system $TS$

TS $\otimes A_{\neg \phi}$

Generalised Büchi automaton $G_{\neg \phi}$

Büchi automaton $A_{\neg \phi}$

Product transition system $TS \otimes A_{\neg \phi}$

$TS \otimes A_{\neg \phi} \models P_{pers}(A_{\neg \phi})$

‘Yes’

‘No’ (counter-example)
From LTL to GNBA

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

- Assume $\varphi$ only contains the operators $\land$, $\neg$, $\square$ and $U$
  - $\lor$, $\rightarrow$, $\Diamond$, $\Box$, $W$, and so on, are expressed in terms of these basic operators
- States are *elementary sets* of sub-formulas in $\varphi$
  - for $\sigma = A_0A_1A_2\ldots \in Words(\varphi)$, 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 $\square$ and expansion law for $U$
- Accept sets guarantee that: $\bar{\sigma}$ is an accepting run for $\sigma$ iff $\sigma \models \varphi$
Elementary sets of formulae

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

1. \( B \) is \textit{logically consistent} if for all \( \varphi_1 \land \varphi_2, \psi \in \text{closure}(\varphi) \):
   
   \begin{itemize}
   \item \( \varphi_1 \land \varphi_2 \in B \iff \varphi_1 \in B \) and \( \varphi_2 \in B \)
   \item \( \psi \in B \implies \neg \psi \notin B \)
   \item \( \text{true} \in \text{closure}(\varphi) \implies \text{true} \in B \)
   \end{itemize}

2. \( B \) is \textit{locally consistent} if for all \( \varphi_1 \lor \varphi_2 \in \text{closure}(\varphi) \):

   \begin{itemize}
   \item \( \varphi_2 \in B \implies \varphi_1 \lor \varphi_2 \in B \)
   \item \( \varphi_1 \lor \varphi_2 \in B \) and \( \varphi_2 \notin B \implies \varphi_1 \in B \)
   \end{itemize}

3. \( B \) is \textit{maximal}, i.e., for all \( \psi \in \text{closure}(\varphi) \):

   \begin{itemize}
   \item \( \psi \notin B \implies \neg \psi \in B \)
   \end{itemize}
The GNBA of LTL-formula \( \varphi \)

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

- \( Q = \) all elementary sets \( B \subseteq \text{closure}(\varphi) \), \( Q_0 = \{ B \in Q \mid \varphi \in B \} \)
- \( \mathcal{F} = \{ \{ B \in Q \mid \varphi_1 \cup \varphi_2 \notin B \text{ or } \varphi_2 \in B \} \mid \varphi_1 \cup \varphi_2 \in \text{closure}(\varphi) \} \)
- The transition relation \( \delta : Q \times 2^{AP} \rightarrow 2^Q \) is given by:
  - If \( A \neq B \cap AP \) then \( \delta(B, A) = \emptyset \)
  - \( \delta(B, B \cap AP) \) is the set of all elementary sets of formulas \( B' \) satisfying:
    (i) For every \( \bigcirc \psi \in \text{closure}(\varphi) : \bigcirc \psi \in B \iff \psi \in B' \), 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 $\bigcirc a$

$Q_0 = \{ B_1, B_3 \}$ since $\bigcirc a \in B_1$ and $\bigcirc a \in B_3$

$\delta(B_2, \{ a \}) = \{ B_3, B_4 \}$ as $B_2 \cap \{ a \} = \{ a \}$, $\neg \bigcirc a = \bigcirc \neg a \in B_2$, and $\neg a \in B_3, B_4$

$\delta(B_1, \{ a \}) = \{ B_1, B_2 \}$ as $B_1 \cap \{ a \} = \{ a \}$, $\bigcirc a \in B_1$ and $a \in B_1, B_2$

$\delta(B_4, \{ a \}) = \emptyset$ since $B_4 \cap \{ a \} = \emptyset \neq \{ a \}$

The set $\mathcal{F}$ is empty, since $\varphi = \bigcirc a$ does not contain an until-operator
GNBA for LTL-formula $a \cup b$

justification: on the black board
Overview Lecture #16

- Repetition: from LTL to GNBA

⇒ Correctness proof

- Complexity results
  - LTL model checking is coNP-hard and PSPACE-complete
  - Satisfiability and validity are PSPACE-hard

- Summary of LTL model checking
Correctness theorem

\[ \text{Words}(\varphi) = \mathcal{L}_\omega(G_\varphi) \]

*Proof: on the black board*
NBA are more expressive than LTL

Corollary: every LTL-formula expresses an \( \omega \)-regular property

But: there exist \( \omega \)-regular properties that cannot be expressed in LTL.

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

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

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!
Overview Lecture #16

- Repetition: from LTL to GNBA
- Correctness proof

⇒ Complexity results
  - LTL model checking is coNP-hard and PSPACE-complete
  - Satisfiability and validity are PSPACE-hard

- Summary of LTL model checking
Complexity for LTL to NBA

For any LTL-formula $\varphi$ (over $AP$) there exists an NBA $A_\varphi$ with $\text{Words}(\varphi) = \mathcal{L}_\omega(A_\varphi)$ and which can be constructed in time and space in $2^{O(|\varphi| \cdot \log |\varphi|)}$.

*Justification complexity: next slide*
Time and space complexity in $2^\mathcal{O}(|\varphi| \cdot \log |\varphi|)$

- States GNBA $G_\varphi$ are elementary sets of formulae in $\text{closure}(\varphi)$
  - sets $B$ can be represented by bit vectors with single bit per subformula $\psi$ of $\varphi$

- The number of states in $G_\varphi$ is bounded by $2^{\left|\text{subf}(\varphi)\right|}$
  - where $\text{subf}(\varphi)$ denotes the set of all subformulae of $\varphi$
  - $\left|\text{subf}(\varphi)\right| \leq 2 \cdot |\varphi|$; so, the number of states in $G_\varphi$ is bounded by $2^\mathcal{O}(|\varphi|)$

- The number of accepting sets of $G_\varphi$ is bounded above by $\mathcal{O}(|\varphi|)$

- The number of states in NBA $A_\varphi$ is thus bounded by $2^\mathcal{O}(|\varphi|) \cdot \mathcal{O}(|\varphi|)$

- $2^\mathcal{O}(|\varphi|) \cdot \mathcal{O}(|\varphi|) = 2^\mathcal{O}(|\varphi| \log |\varphi|)$

qed
There exists a family of LTL formulas $\varphi_n$ with $|\varphi_n| = \mathcal{O}(\text{poly}(n))$ such that every NBA $A_{\varphi_n}$ for $\varphi_n$ has at least $2^n$ states.
Proof (1)

Let $AP$ be non-empty, that is, $|2^{AP}| \geq 2$ and:

$$L_n = \left\{ A_1 \ldots A_n A_1 \ldots A_n \sigma \mid A_i \subseteq AP \land \sigma \in \left(2^{AP}\right)^\omega \right\}, \quad \text{for } n \geq 0$$

It follows $L_n = Words(\varphi_n)$ where

$$\varphi_n = \bigwedge_{a \in AP} \bigwedge_{0 \leq i < n} (\bigcirc^i a \leftrightarrow \bigcirc^{n+i} a)$$

$\varphi_n$ is an LTL formula of polynomial length: $|\varphi_n| \in \mathcal{O}(|AP| \cdot n)$

However, any NBA $\mathcal{A}$ with $L_\omega(\mathcal{A}) = L_n$ has at least $2^n$ states
Proof (2)

Claim: any NBA $\mathcal{A}$ for $\bigwedge_{a \in \mathcal{A}P} \bigwedge_{0 \leq i < n} (\circ^i a \leftrightarrow \circ^{n+i} a)$ has at least $2^n$ states

Words of the form $A_1 \ldots A_n A_1 \ldots A_n \varnothing \varnothing \ldots$ are accepted by $\mathcal{A}$

$\mathcal{A}$ thus has for every word $A_1 \ldots A_n$ of length $n$, a state $q(A_1 \ldots A_n)$, say, which can be reached from an initial state by consuming $A_1 \ldots A_n$

From $q(A_1 \ldots A_n)$, it is possible to visit an accept state infinitely often by accepting the suffix $A_1 \ldots A_n \varnothing \varnothing \varnothing \ldots$

If $A_1 \ldots A_n \neq A'_1 \ldots A'_n$ then

$$A_1 \ldots A_n A'_1 \ldots A'_n \varnothing \varnothing \varnothing \ldots \notin L_n = L_\omega(\mathcal{A})$$

Therefore, the states $q(A_1 \ldots A_n)$ are all pairwise different

Given $|2^{\mathcal{A}P}|$ possible sequences $A_1 \ldots A_n$, NBA $\mathcal{A}$ has $\geq \left( |2^{\mathcal{A}P}| \right)^n \geq 2^n$ states
The time and space complexity of LTL model checking is in $O(|TS| \cdot 2^{|\phi|})$. 
On-the-fly LTL model checking

- Idea: find a counter-example *during* the generation of $\text{Reach}(TS)$ and $A_{\neg \varphi}$
  - exploit the fact that $\text{Reach}(TS)$ and $A_{\neg \varphi}$ can be generated in parallel

⇒ Generate $\text{Reach}(TS \otimes A_{\neg \varphi})$ “on demand”
  - consider a new vertex only if no accepting cycle has been found yet
  - only consider the successors of a state in $A_{\neg \varphi}$ that match current state in $TS$

⇒ Possible to find an accepting cycle *without generating* $A_{\neg \varphi}$ entirely

- This *on-the-fly* scheme is adopted in e.g. the model checker SPIN
The LTL model-checking problem is co-NP-hard

The Hamiltonian path problem is polynomially reducible to the complement of the LTL model-checking problem

In fact, the LTL model-checking problem is PSPACE-complete [Sistla & Clarke 1985]
LTL satisfiability and validity checking

- **Satisfiability problem**: $\text{Words}(\varphi) \neq \emptyset$ for LTL-formula $\varphi$?
  - does there exist a transition system for which $\varphi$ holds?

- **Solution**: construct an NBA $A_{\varphi}$ and check for emptiness
  - nested depth-first search for checking persistence properties

- **Validity problem**: is $\varphi \equiv$ true, i.e., $\text{Words}(\varphi) = (2^{AP})^\omega$?
  - does $\varphi$ hold for every transition system?

- **Solution**: as for satisfiability, as $\varphi$ is valid iff $\lnot \varphi$ is satisfiable

  run time is exponential; a more efficient algorithm most probably does not exist!
#16: Complexity and correctness

LTL satisfiability and validity checking

The satisfiability and validity problem for LTL are PSPACE-complete

Black board: show the fact that these problems are PSPACE-hard
Overview Lecture #16

• Repetition: from LTL to GNBA

• Correctness proof

• Complexity results
  – LTL model checking is coNP-hard and PSPACE-complete
  – Satisfiability and validity are PSPACE-hard

⇒ Summary of LTL model checking
Summary of LTL model checking (1)

- LTL is a logic for formalizing path-based properties

- Expansion law allows for rewriting until into local conditions and next

- LTL-formula $\varphi$ can be transformed algorithmically into NBA $A_{\varphi}$
  - this may cause an exponential blow up
  - algorithm: first construct a GNBA for $\varphi$; then transform it into an equivalent NBA

- LTL-formulae describe $\omega$-regular LT-properties
  - but do not have the same expressivity as $\omega$-regular languages
Summary of LTL model checking (2)

- \( TS \models \varphi \) can be solved by a nested depth-first search in \( TS \otimes A_{\neg \varphi} \)
  - time complexity of the LTL model-checking algorithm is linear in \( TS \) and exponential in \( |\varphi| \)

- Fairness assumptions can be described by LTL-formulae
  the model-checking problem for LTL with fairness is reducible to the standard LTL model-checking problem

- The LTL-model checking problem is PSPACE-complete

- Satisfiability and validity of LTL amounts to NBA emptiness-check

- The satisfiability and validity problem for LTL are PSPACE-complete