Finite-state model checking for linear temporal logic

Stanislav Moiseev

May 2018

Syntax of linear temporal logic

Definition.

Given a set *Props* of *atomic propositions*.

Linear temporal logic formulae are defined according to the following grammar:

$$\varphi ::= \top \mid a \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \mathit{Next}\, \varphi \mid (\varphi_1\, \mathit{Until}\, \varphi_2)$$

where $a \in \mathit{Props}$.

Automata-based LTL model checking

Input

- Finite transition system *TS*.
- LTL formula φ (both over one set of atomic propositions *Props*).

Output

- "Yes" if $TS \models \varphi$.
- "No" plus a counterexample, otherwise.

Automata-based LTL model checking

Algorithm

0. Reduce regains in formula: $\varphi \mapsto \mathit{simpl}(\varphi)$.
1. Construct a generalized nondeterministic Büchi automaton $A_{\neg\varphi}$ such that

$$L(A_{\neg\varphi}) = L(\neg\varphi).$$

2. Transform GNBA $A_{\neg\varphi}$ into an equivalent nondeterministic Büchi automaton $A'_{\neg\varphi}$.
3. Construct the product transition system $TS \otimes A'_{\neg\varphi}$.
4. Check whether there exists a path π in $TS \otimes A'_{\neg\varphi}$ satisfying the accepting condition of $A'_{\neg\varphi}$.
   If it exists then return "No" plus a prefix of π; otherwise, return "Yes".

Reduction of negations

Definition.

- Given an LTL formula φ.
- *simpl* φ is the formula obtained from φ by a reduction of all sequences of $\neg$'s in all subformulae, e.g.

$$\mathit{simpl}\big(\neg\neg(A \wedge \neg\neg\neg B)\big) = \big(A \wedge \neg B\big).$$

- *Later on we assume that the formula is simplified.*

Generalized nondeterministic Büchi automaton

Definition.
A generalized NBA is a tuple $G = \langle Q, \Sigma, \delta, Q_0, F \rangle$ where

- Q is a finite set of states.
- Σ is a finite alphabet.
- $Q_0 \subseteq Q$ is the set of initial states.
- $\delta \colon Q \times \Sigma \to 2^Q$ is the transition function.
- $F \subseteq 2^Q$ is a (possible empty) set of acceptance sets.

The language of GNBA

Definition.

A *run* of a GNBA G for an ω-word $\langle a_1, a_2, \ldots \rangle \in \Sigma^\omega$ is a (finite or infinite) sequence of states $\pi = \langle q_0, q_1, \ldots \rangle$ such that

- $q_0 \in Q_0$.
- $q_k \in \delta(q_{k-1}, a_k)$ for all $k \in \{1, 2, \ldots\}$.

Definition.

The *accepted language* $L(G)$ is the set of all ω-words in Σ^ω that have (at least one) infinite run π such that for all $F_0 \in F$

$$F_0 \cap \mathit{Lim}(\pi) \neq \varnothing$$

where $\mathit{Lim}(\pi)$ is the set of all states $q \in Q$ that occur infinitely many times in π.

The language of GNBA: comments

- If $F = \varnothing$ then $L(G)$ consists of all ω-words that have an infinite run.
  I.e. if $F = \varnothing$ then G is equivalent to a GNBA with the set of accepting sets being $\{Q\}$.
- If $F = \{F_0\}$ is a singleton set then GNBA G is a NBA.

GNBA for LTL formula

Given an LTL formula φ over a set of atomic propositions *Props*.

An equivalent GNBA is $G = \langle Q, 2^{Props}, \delta, Q_0, F \rangle$ where

- Q is the set of all elementary sets of formulae $B \subseteq closure(\varphi)$.
- $Q_0 = \{B \in Q \mid \varphi \in B\}$.
- $F = \{F_{\varphi_1\, Until\, \varphi_2} \mid (\varphi_1\, Until\, \varphi_2) \in closure(\varphi)\}$ where

  $$F_{\varphi_1\, Until\, \varphi_2} = \{B \in Q \mid (\varphi_1\, Until\, \varphi_2) \notin B \vee \varphi_2 \in B\}.$$

GNBA for LTL formula

The transition relation $\delta : Q \times 2^{\mathit{Props}} \rightarrow 2^{Q}$ is defined as follows:

- If $A \neq B \cap \mathit{Props}$, then $\delta(B, A) = \varnothing$.
- If $A = B \cap \mathit{Props}$, then $\delta(B, A)$ is the set of all elementary sets B' of formulae satisfying:
  1. for every $\mathit{Next}\,\psi \in \mathit{closure}(\psi)$:

     $$\mathit{Next}\,\psi \in B \Longleftrightarrow \psi \in B'.$$

  2. for every $(\varphi_1\, \mathit{Until}\, \varphi_2) \in \mathit{closure}(\varphi)$:

     $$((\varphi_1\, \mathit{Until}\, \varphi_2) \in B) \Longleftrightarrow$$
     $$\Longleftrightarrow (\varphi_2 \in B \;\vee\; (\varphi_1 \in B \;\wedge\; (\varphi\, \mathit{Until}\, \varphi_2) \in B')).$$

Closure of φ

Definition.

- The set *closure*(φ) is the set of all subformulae of *simpl* φ and their negations:

$$\textit{closure}(\varphi) = \bigcup_{\substack{\psi \text{ is a subformula} \\ \text{of } \textit{simpl}\,\varphi}} \{\psi,\ \textit{neg}\,\psi\}.$$

- *neg* is defined as follows:

$$\textit{neg}(\neg\eta) = \eta$$
$$\textit{neg}(\eta) = \neg\eta$$

where η does not start with $\neg$.

Elementary sets of formulae

Definition.
$B \subseteq \mathit{closure}(\varphi)$ is *elementary* if

1. B is consistent with respect to propositional logic.
2. B is is locally consistent with respect to the *Until* operator.
3. B is maximal.

Elementary sets of formulae

Definition.
$B \subseteq closure(\varphi)$ is *consistent with respect to propositional logic* if

1. For all $(\varphi_1 \wedge \varphi_2) \in closure(\varphi)$:
$$((\varphi_1 \wedge \varphi_2) \in B) \Longleftrightarrow (\varphi_1 \in B \wedge \varphi_2 \in B).$$
2. $(\psi \in B) \Longrightarrow (neg\, \psi \notin B)$.
3. $(\top \in closure(\varphi)) \Longrightarrow (\top \in B)$.

Elementary sets of formulae

Definition.
$B \subseteq \mathit{closure}(\varphi)$ is *locally consistent with respect to the Until operator* if for all $(\varphi_1 \,\mathit{Until}\, \varphi_2) \in \mathit{closure}(\varphi)$:

1. $(\varphi_2 \in B) \Longrightarrow ((\varphi_1 \,\mathit{Until}\, \varphi_2) \in B)$.
2. $((\varphi_1 \,\mathit{Until}\, \varphi_2) \in B \ \wedge \ \varphi_2 \notin B) \Longrightarrow (\varphi_1 \in B)$.

Definition.
$B \subseteq \mathit{closure}(\varphi)$ is *maximal* if for all $\psi \in \mathit{closure}(\varphi)$:

$$(\psi \notin B) \Longrightarrow (\mathit{neg}\, \psi \in B).$$