Automata Theoretic LTL Model Checking

Model-checking exercises

Giuseppe Perelli

SAPIENZA
UNIVERSITÀ DI ROMA

Formal Methods 2020/21

- Recap on automata constructions
- A model-checking procedure
- Exercises

Theorem

For an LTL formula φ, we can construct a (generalized) nondeterministic Büchi automaton $\mathcal{N}_\varphi = \langle Q, \Sigma, I, \delta, F \rangle$ such that $\mathcal{L}(\mathcal{N}_\varphi) = \mathcal{L}(\varphi)$.

Several constructions of $\mathcal{N}_\varphi$ are available in the literature, including online tools:

- http://www.lsv.fr/ gastin/ltl2ba/index.php
- https://owl.model.in.tum.de/try/
- https://spot.lrde.epita.fr/app/

These constructions are always hard to handle manually, as they provide exponentially sized automata.
However, the general construction is not always necessary in practice.

Exercise

From LTL to (G)NBA in practice

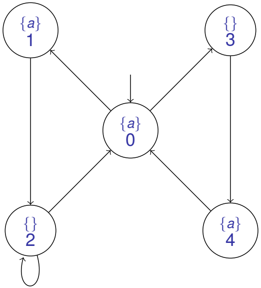
- $p \mathsf{U} q$
- $\mathsf{F} p$
- $\mathsf{G} p$
- $q \mathsf{U} (\mathsf{X}\mathsf{X} p)$
- $\mathsf{G}(p \to \mathsf{F} q)$
- $\mathsf{G}\mathsf{F} p$
- $\mathsf{F}\mathsf{G} p$
- $\mathsf{G}\mathsf{F} p \wedge \mathsf{G}\mathsf{F} q$

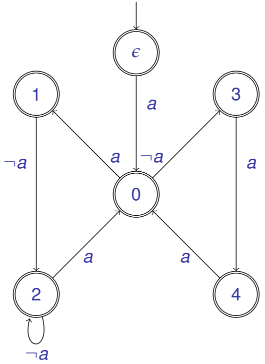
A labeled transition system $\mathcal{T} = \langle S, S_0, E(\subseteq S \times S), \lambda\rangle$ with $\lambda : S \to 2^{\mathrm{Prop}}$ is turned into a NBA $\mathcal{N}_{\mathcal{T}} = \langle \Sigma, Q, Q_0, \delta, F\rangle$ with:

- ▷ $\Sigma = 2^{\mathrm{Prop}}$
- ▷ $Q = S \cup \{\epsilon\}$
- ▷ $Q_0 = \{\epsilon\}$
- ▷ $\delta(\epsilon, \sigma) = \{s \in S_0 : \sigma = \lambda(s_0)\}$
  $\delta(s, \sigma) = \{s' \in S : (s, s') \in E \text{ and } \sigma = \lambda(s')\}$
- ▷ $F = Q$

The labeling of states is pushed backward to the incoming edges.
A root state is included to push the initial state labels backward.
Every state is accepting.

From Labeled Transition Systems to NBA





Theorem

For every labeled transition system $\mathcal{T}$, the automaton $\mathcal{N}_{\mathcal{T}}$ recognizes all and only those infinite words that are generated by $\mathcal{T}$.

LTL model checking algorithm takes:

- a model $\mathcal{T}$ and
- a formula φ

and returns

- Yes if $\mathcal{T} \models \varphi$
- No and a counter-example if $\mathcal{T} \not\models \varphi$

Here we look into the automata-based approach

(alternatively, tableaux construction)

(and indeed, in practice more alternatives)

- Consider a model $\mathcal{T}$ and an LTL property φ
- $\mathcal{T} \models \varphi$ if for all the paths π of $\mathcal{T}$, it holds that $\pi \models \varphi$, namely if $\pi \in \mathcal{L}(\varphi)$.
- Equivalently, $\mathcal{T}$ admits no path π such that $\pi \models \neg\varphi$ (no counterexample)
- More formally

$$
\begin{aligned}
\mathcal{T} \models \varphi \Leftrightarrow\ & \mathcal{L}(\mathcal{T}) \subseteq \mathcal{L}(\varphi) \\
\Leftrightarrow\ & \mathcal{L}(\mathcal{T}) \cap \overline{\mathcal{L}(\varphi)} = \emptyset \\
\Leftrightarrow\ & \mathcal{L}(\mathcal{T}) \cap \mathcal{L}(\neg\varphi) = \emptyset
\end{aligned}
$$

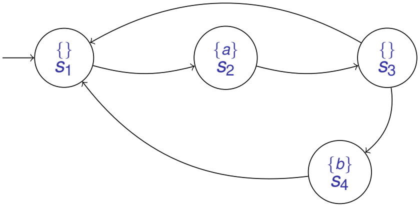
- Input:
  - a model $\mathcal{T}$ and
  - a formula φ
- Construction:
  - Construct the automaton $\mathcal{N}_{\mathcal{T}}$ from the LTS
  - Construct the automaton $\mathcal{N}_{\neg\varphi}$ from the LTL formula
  - Construct the product automaton $\mathcal{N}_{\mathcal{T},\neg\varphi} = \mathcal{N}_{\mathcal{T}} \otimes \mathcal{N}_{\neg\varphi}$
- Solve nonemptiness problem:

$$\mathcal{L}(\mathcal{N}_{\mathcal{T},\neg\varphi}) \overset{?}{\neq} \emptyset$$

Output:

- Yes if $\mathcal{L}(\mathcal{N}_{\mathcal{T},\neg\varphi}) = \emptyset$
- No if otherwise (and show a counterexample path $\pi \models \neg\varphi$)

Exercise



$$\mathsf{X}a \land (\mathsf{G}(b \to \mathsf{X}a)) \land \mathsf{F}a$$