Formal verification - Tutorial 03

09/03/2026

Alternating Büchi automata

Exercise 1. *Fix the alphabet $\Sigma = \{a, b\}$. Consider the following ABA A recognising all words containing infinitely many a's and infinitely many b's: $A = (Q, \Sigma, q_0, \delta, F)$, where*

$$\begin{aligned} Q &= \{p_0, p_a, p_b, p'_a, p'_b\}, \\ F &= \{p'_a, p'_b\}, \\ \delta(p_0, x) &= p_x \wedge p_0, \text{ for } x \in \{a, b\}, \\ \delta(p_a, a) &= p'_a, \\ \delta(p_a, b) &= p_a, \\ \delta(p'_a, a) &= \delta(p'_a, b) = p_a, \end{aligned}$$

The transitions for p_b and p'_b are similar. Use the breakpoint construction of Miyano and Hayashi to construct an equivalent non-deterministic Büchi automaton.

In the next exercise, we investigate the conciseness of alternating Büchi automata compared to non-deterministic ones, and thus the inherent cost of the breakpoint construction.

Exercise 2. *For every $k \in \mathbb{N}$, construct an alternating Büchi automaton A_k with $O(k)$ states s.t. every language equivalent non-deterministic Büchi automaton has at least 2^k states.*

Solution. Let $\Sigma = \{a, b\}$. For instance, let L_k the set of words $w \in \Sigma^\omega$ of the form $w = u^\omega$ for some $u \in \Sigma^k$. It is easy to construct an ABA A_k for L_k with $O(k)$ states. Moreover, every equivalent NBA B needs at least 2^k states, since (intuitively) it needs to keep track of the last k letters read. Formally, if B had $< 2^k$ states, then there are two *distinct* words $u, v \in \Sigma^k$ such that B reaches the same state after reading u and v. Then B would accept uv^ω, which is not in L_k. □

Linear temporal logic

Exercise 3. *For every $k \in \mathbb{N}$, construct a formula of LTL φ_k of size $O(k)$ s.t. every equivalent non-deterministic Büchi automaton has at least 2^k states.*

Solution. Let $P = \{p\}$. For every $k \in \mathbb{N}$, let φ_k be the LTL formula $\varphi_k = G((p \to X^k p) \wedge (\neg p \to X^k \neg p))$. Then models of φ_k are precisely the words $w \in (2^P)^\omega$ s.t. $w = u^\omega$ for some $u \in (2^P)^k$. The same argument as in Exercise 2 shows that every NBA equivalent to φ_k has at least 2^k states. □

Exercise 4 ([**?**, Ch. 2, Lemma 6])**.** *Let $P = \{p\}$. Consider the property "p holds only at even time steps". Show that this property is not expressible in LTL.*

Solution. For every $i \in \mathbb{N}$, let w_i be the infinite word over $2^{\{p\}} = \{\varnothing, \{p\}\}$ where p holds precisely at time i, and nowhere else:

$$w_i = \varnothing \cdots \varnothing \, \{p\} \, \varnothing\varnothing \cdots$$

The intuition is that a fixed LTL formula cannot distinguish between w_i and w_{i+1}, for sufficiently large i. For an LTL formula φ, let $X(\varphi)$ be the number of $\mathbf{X}$ operators in φ. We show that for every formula φ,

$$\forall i > X(\varphi): \quad w_i \models \varphi \quad \text{iff} \quad w_{i+1} \models \varphi. \tag{1}$$

We proceed by structural induction on φ.

1. If $\varphi = p$, then $X(\varphi) = 0$ and the claim clearly holds since both w_i and w_{i+1} do not satisfy p when $i > 0$.

2. Let $\varphi = \psi_0 \wedge \psi_1$. Let $i > X(\varphi)$. Note that $X(\varphi) = X(\psi_0) + X(\psi_1)$, and thus $i > X(\varphi) \geqslant X(\psi_0), X(\psi_1)$. The claim follows imediately from the LTL semantics and the induction hypothesis for ψ_0 and ψ_1:

$$\begin{aligned} w_i \models \varphi \quad &\text{iff} \quad w_i \models \psi_0 \text{ and } w_i \models \psi_1 \\ &\text{iff} \quad w_{i+1} \models \psi_0 \text{ and } w_{i+1} \models \psi_1 \\ &\text{iff} \quad w_{i+1} \models \varphi. \end{aligned}$$

3. The case $\varphi = \neg\psi$ is similar.

4. Let $\varphi = \mathbf{X}\psi$ and assume $i > X(\varphi)$. Note that $X(\varphi) = 1 + X(\psi)$. We then have $i - 1 > X(\psi)$, and thus

$$\begin{aligned} w_i \models \varphi \quad &\text{iff} \quad w_{i-1} \models \psi \\ &\text{iff} \quad w_i \models \psi \\ &\text{iff} \quad w_{i+1} \models \varphi. \end{aligned}$$

5. Finally, consider the case $\varphi = \psi_0 \mathbf{U} \psi_1$. By the semantics of until, we have the following two equivalences:

$$w_i \models \varphi \quad \text{iff} \quad w_i \models \psi_1 \text{ or } (w_i \models \psi_0 \text{ and } w_{i-1} \models \varphi), \text{ and} \tag{2}$$

$$w_{i+1} \models \varphi \quad \text{iff} \quad w_{i+1} \models \psi_1 \text{ or } (w_{i+1} \models \psi_0 \text{ and } w_i \models \varphi). \tag{3}$$

Now assume $w_i \models \varphi$. By (2), if $w_i \models \psi_1$, then $w_{i+1} \models \psi_1$ by the induction hypothesis, and thus $w_{i+1} \models \varphi$ by (3). Otherwise, we have $w_i \models \psi_0$, thus $w_{i+1} \models \psi_0$ by the induction hypothesis, which together with $w_i \models \varphi$ by (3) gives $w_{i+1} \models \varphi$, as required.

Viceversa, assume $w_{i+1} \models \varphi$. By (3), if $w_{i+1} \models \psi_1$, then $w_i \models \psi_1$ by the induction hypothesis, and thus $w_i \models \varphi$ by (2). Otherwise, by (3) we directly have $w_i \models \varphi$, as required.

□

Let X be a countable set of first-order variables. Formulas of first-order logic (FO) over the structure $(\mathbb{N}, \leqslant, p_1, \ldots, p_n)$ are constructed as follows:

$$\varphi, \psi ::= p_i(x) \mid x \leqslant y \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \exists x.\varphi \mid \forall x.\varphi,$$

where $x, y \in X$.

Exercise 5. *Show that for every LTL formula φ there is an FO formula $\psi(x)$ of one free variable x, which is equivalent to φ in the following sense: For every infinite word w and time step i,*

$$w, [x \mapsto i] \models \psi(x) \quad \textit{iff} \quad w[i] \models \varphi. \tag{4}$$

(Recall that for $w = a_0 a_1 \cdots$ and $i \in \mathbb{N}$, we write $w[i]$ for the suffix of $w = a_i a_{i+1} \cdots$ starting at position i.) How many FO variable names do you need to construct ψ?

Solution. Define a function $[\_]$ mapping an LTL formula φ to an equivalent (in the sense of (4)) FO formula $[\varphi](x)$ with one free variable x. We proceed by structural induction on φ.

$$\begin{aligned}
[p_i](x) &= p_i(x), \\
[\neg\varphi](x) &= \neg[\varphi](x), \\
[\varphi \wedge \psi](x) &= [\varphi](x) \wedge [\psi](x), \\
[\varphi \vee \psi](x) &= [\varphi](x) \vee [\psi](x), \\
[\mathbf{X}\varphi](x) &= \exists y.(y = x + 1) \wedge [\varphi](y), \\
[\varphi \mathbf{U} \psi](x) &= \exists y.(y \geqslant x) \wedge [\psi](y) \wedge \forall z.x \leqslant z < y \to [\varphi](z).
\end{aligned}$$

We have used $y = x+1$ as a shorthand for $x < y \wedge \neg\exists z.(x < z \wedge z < y)$. We have used three variable names (they can be recycled) when constructing subformulas. This is optimal, since it is known that the until operator requires at least three variables to be expressed in FO. □

CTL

Exercise 6. *Show a CTL model checking algorithm of complexity*

$$O((|S| + |\to|) \cdot |\varphi|),$$

where S is the set of states and $\to$ the transition relation of the input Kripke structure.

Solution. Let $s_0 \in S$ be the initial state. We want to decide whether $M, s_0 \models \varphi$. For $\mathbf{EF}\varphi$, it suffices to check whether s_0 can reach a state satisfying φ, which can be done by a breadth-first search in $O(|S| + |\to|)$ time. For $\mathbf{EG}\varphi$, we perform a strongly connected component (SCC) decomposition of the transition graph of M, in $O(|S| + |\to|)$ time, and then we check whether s_0 can reach a SCC containing only states satisfying φ. The cases $\mathbf{AF}\varphi$ and $\mathbf{AG}\varphi$ can be treated by double complementation. For the more general $\mathbf{E}(\varphi \mathbf{U} \psi)$, first remove all states not satisfying φ, and then check $\mathbf{EF}\psi$ on the new structure. Similarly for $\mathbf{E}(\varphi \mathbf{R} \psi)$. □

Exercise 7. *Show that the CTL model checking problem is P-hard.*

Solution. We reduce from the *monotone circuit value problem* (MCVP), which is known to be P-hard. Formally, given a Boolean circuit C (acyclic DAG) with gates $\wedge$ and $\vee$, a single output gate, and an assignment in $\{0, 1\}$ for the input gates, the MCVP asks whether the output gate evaluates to 1. First, we modify C so that the same connective $\wedge$ or $\vee$ appears at the same level. Moreover, we assume that the output gate is labelled by $\wedge$, This can be achieved by introducing at most $|C|$ dummy gates. Then, we construct a Kripke structure K which is the same as C, but where each gate is a state and the edges are reversed. We have a single proposition $P = \{p\}$, which holds at input gates evaluating to 1, and does not hold at any other gate (including input gates evaluating to 0). Finally, we construct a CTL formula φ,

$$\mathbf{AX}\ \mathbf{EX}\ \cdots Q\ p,$$

where the number of modal operators is equal to the depth of C. Thus the last operator Q is **AX** if the depth is odd, and **EX** otherwise. □