Formal verification - Tutorial 02

March 16, 2026

Linear temporal logic

Fix a set of propositional variables P. Recall that linear temporal logic (LTL) formulas over P are defined by the following (minimalistic) grammar:

$$\varphi, \psi ::= \mathbf{true} \mid \mathbf{false} \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\psi,$$

where $p \in P$. We define the derived operators

$$\mathbf{F}\varphi = \mathbf{true}\mathbf{U}\varphi$$
$$\mathbf{G}\varphi = \neg\mathbf{F}\neg\varphi.$$

Exercise 1. *Write LTL formulas for the following properties:*

1. *p holds infinitely often;*
2. *whenever p holds, then q holds afterwards;*
3. *p holds* precisely *at every even step;*

Solution. To express that p holds infinitely often, we can write $\mathbf{GF}p$. The second property is expressed by $\mathbf{G}(p \to \mathbf{F}q)$. The last property can be expressed in a variety of ways. For instance, we can write

$$p \wedge \mathbf{G}((p \implies \mathbf{X}\neg p) \wedge (\neg p \implies \mathbf{X}p)).$$ □

Exercise 2. *Consider the property "for every prefix of the computation, the number of times p holds is at most the number of times q holds". Is this property expressible in LTL?*

Solution. No, the stated property is not expressible in LTL. The reason is that LTL formulas can only express ω-regular properties, and the property in question is not ω-regular. □

Exercise 3. *Consider the property "p holds at all even time steps" (but possibly also at odd steps). Is this property expressible in LTL?*

Solution. This property is not expressible in LTL, but in order to show it, we need to use a more advanced tool (next tutorial). □

From LTL to alternating Büchi automata

We recall the translation from LTL to alternating Büchi automata. To this end, we first convert an LTL formula φ to negation normal form, i.e., we push negations inside until they only occur in front of propositional variables. To achieve this, we introduce the De Morgan dual of the until operator, which is called the *release* operator:

$$\varphi \mathbf{R} \psi = \neg(\neg\varphi \mathbf{U} \neg\psi).$$

Note that $\mathbf{G}\varphi$ is equivalent to $\mathbf{false}\mathbf{R}\varphi$.

We will use the following fixpoint expansions of the until and release operators:

$$\varphi \mathbf{U} \psi = \psi \vee (\varphi \wedge \mathbf{X}(\varphi \mathbf{U} \psi)), \tag{1}$$

$$\varphi \mathbf{R} \psi = \psi \wedge (\varphi \vee \mathbf{X}(\varphi \mathbf{R} \psi)). \tag{2}$$

Given an LTL formula φ in negation normal form, we construct an alternating Büchi automaton $A = (\Sigma, Q, q_0, F, \delta)$ as follows. States of the automaton Q are the subformulas of φ, to which we add **true** and **false**. The initial state is $q_0 = \{\varphi\}$ and the set of accepting states F contains all the release formulas and **true**. The transition function is defined by induction on the structure of states $\psi \in Q$. The cases for propositional variables and positive Boolean connectives are straightforward:

$$\begin{aligned}
\delta(\mathbf{true}, a) &= \mathbf{true}, \\
\delta(\mathbf{false}, a) &= \mathbf{false}, \\
\delta(p, a) &= \mathbf{true} \text{ if } p \in a, \text{ and } \mathbf{false} \text{ otherwise}, \\
\delta(\neg p, a) &= \mathbf{true} \text{ if } p \notin a, \text{ and } \mathbf{false} \text{ otherwise}, \\
\delta(\psi \wedge \chi, a) &= \delta(\psi, a) \wedge \delta(\chi, a), \\
\delta(\psi \vee \chi, a) &= \delta(\psi, a) \vee \delta(\chi, a).
\end{aligned}$$

The most interesting cases are those for the temporal operators.

$$\begin{aligned}
\delta(\mathbf{X}\psi, a) &= \psi, \\
\delta(\psi \mathbf{U} \chi, a) &= \delta(\chi, a) \vee (\delta(\psi, a) \wedge \psi \mathbf{U} \chi), \\
\delta(\psi \mathbf{R} \chi, a) &= \delta(\chi, a) \wedge (\delta(\psi, a) \vee \psi \mathbf{R} \chi).
\end{aligned}$$

For until and release we have applied the transition function $\delta(\cdot, a)$ to the fixpoint expansions given above (1) and (2).

Exercise 4. *Apply the translation above to construct an alternating Büchi automaton for the LTL formula* $\mathbf{GF}p$.

Exercise 5. *Using the automaton above, how can we construct an automaton for* $\mathbf{GF}p \wedge \mathbf{GF}q$*?*