TD 2: LTL

Exercises 1–4 (marked with an asterisk in the margin) are to be prepared at home *before* the session.

1 Specification

Exercise 1. We would like to verify the properties of a boolean circuit with input x, output y, and two registers r_1 and r_2. We define accordingly $\mathrm{AP} = \{x, y, r_1, r_2\}$ as our set of atomic propositions, and model check infinite runs $\sigma = s_0 s_1 s_2 \cdots$ from $(2^{\mathrm{AP}})^\omega$. (∗)

Translate the following properties (a) in LTL and (b) in FO(<):

1. "it is impossible to get two consecutive 1 as output"
2. "each time the input is 1, at most two ticks later the output will be 1"
3. "each time the input is 1, the register contents remains the same over the next tick"
4. "register r_1 is infinitely often 1"

Note that there might be several, non-equivalent formal specifications matching these informal descriptions—that's the whole point of writing specifications!—but your (a) and (b) should be equivalent.

2 LTL

Exercise 2. We fix a set AP of atomic propositions including $\{p, q, r\}$, and fix the associated alphabet $\Sigma = 2^{\mathrm{AP}}$. (∗)

1. Consider the formulæ $\varphi_1 = \mathsf{G}(p \to q)$ and $\varphi_2 = \mathsf{G}(p \to ((\neg q)\ \mathsf{R}\ q))$.
   (a) Does φ_1 imply φ_2?
   (b) Does φ_2 imply φ_1?
2. Simplify the following formula:

$$\mathsf{F}(((\mathsf{G}r)\,\mathsf{U}\,p) \wedge (\neg q\,\mathsf{U}\,p)) \vee \mathsf{F}(\neg p \vee \mathsf{F}q)\ .$$

Exercise 3 (Expressiveness)**.** We fix a set AP of atomic propositions containing p, and fix the associated alphabet $\Sigma = 2^{\mathrm{AP}}$. (∗)

1. Show that the following sets are expressible in LTL:

(a) $\Sigma_p^* \cdot \Sigma_{\neg p}^\omega$, and

(b) $\Sigma_p^n \cdot \Sigma_{\neg p}^\omega$ for each fixed $n \geq 0$.

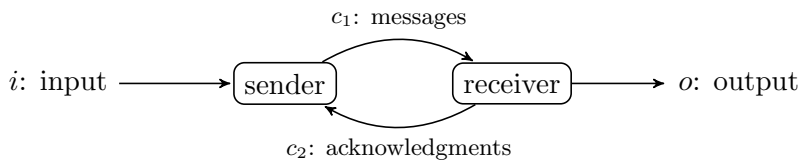
2. Is the language $(\Sigma_p \cdot \Sigma_{\neg p})^\omega$ expressible in LTL?

3. Consider the infinite sequence $\sigma_i = \Sigma_p^i \cdot \Sigma_{\neg p} \cdot \Sigma_p^\omega$ for $i \geq 0$. Show that, for any LTL formula φ_n with less than n X modalities, and for all $i, i' > n$, $\sigma_i \models \varphi_n$ iff $\sigma_{i'} \models \varphi_n$.

4. Using the previous question, show that the language $(\Sigma_p \cdot \Sigma)^\omega$ is not expressible in LTL.

3 Models

Exercise 4 (Alternating Bit Protocol)**.** The alternating bit protocol allows to exchange messages over a lossy channel, and to ensure that no messages are lost. The protocol employs two processes, a *sender* and a *receiver*, that communicate through two lossy channels c_1 and c_2 as depicted below: (∗)



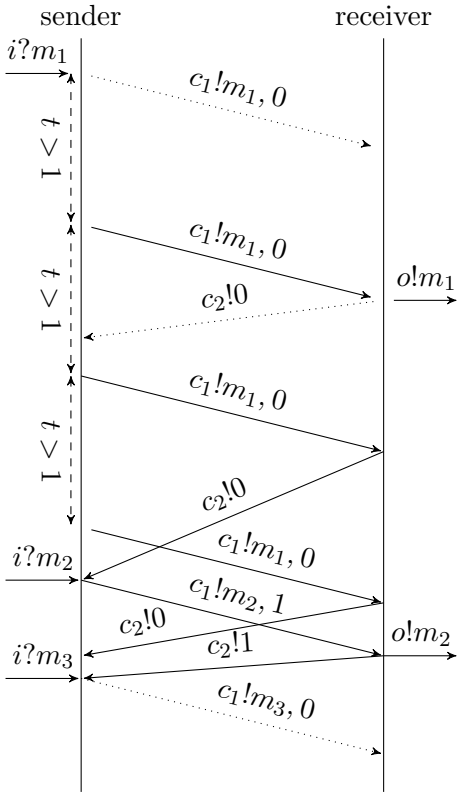
The gist of the protocol is that both the sender and the receiver will retransmit data over the lossy channels, until they receive proof that at least one of their messages has gone through. For this, an *alternating bit* is attached to all their communications, and is changed whenever the processes know that their previous message has been received. The following schema illustrates a message exchange between the processes.

Lost messages are represented with dotted arrows, and we endow the sender with a timer t that triggers resending the message if the acknowledgment with the appropriate bit value has not been received.

1. Propose two models, one for the sender and one for the receiver.

2. Label some of your states with the atomic propositions sent_1, sent_2, rec_1, and rec_2. How can you express the following fairness constraints on the two channels in LTL: "if infinitely many messages are sent, then infinitely many are received"?

3. Let $m_1, \ldots, m_n$ be a *fixed* sequence of messages to send. How can one specify in LTL that exactly the same sequence will be received, in the same order, and without duplicates?



4 LTL with Past

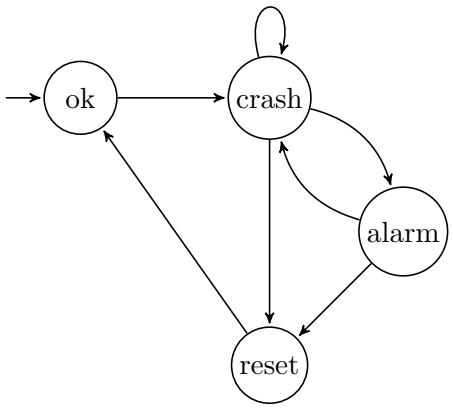
Exercise 5 (Specifying with Past)**.** Provide LTL formulæ over AP = {ok, crash, alarm, reset} with and without past modalities for the following properties:

1. "Whenever the alarm rings, there has been a crash immediately before."
2. "Whenever the alarm rings, there has been a crash some time before, and no reset in the meantime."

Exercise 6 (History Variables)**.** One way of getting rid of past modalities is to tweak both the model and the formula, by adding *history variables* to the model and by replacing the past subformulæ by atomic propositions on these variables, i.e. from a pair $\langle M, \varphi \rangle$ where M is a Kripke model and φ a LTL formula with past modalities, construct $\langle M', \varphi' \rangle$ where M' is a modified version of M with extra atomic propositions, and φ' is a pure future LTL formula, such that $M \models \varphi$ iff $M' \models \varphi'$.

For instance, a subformula $\mathsf{Y}\psi$ will be replaced by a boolean variable $h_{\mathsf{Y}\psi}$ in the specification, and the model will update this variable according to whether or not ψ holds in the previous state. Two new atomic propositions are introduced, corresponding to $h_{\mathsf{Y}\psi} = \text{true}$ and $h_{\mathsf{Y}\psi} = \text{false}$.

1. Apply this technique to the specification of the previous exercise and the following alarm system:



2. What is the cost of the model transformation?

Exercise 7 (Succinctness of Past Formulæ)**.** Let $\mathrm{AP}_{n+1} = \{p_0, \ldots, p_n\} = \mathrm{AP}_n \cup \{p_n\}$ be a set of atomic proositions, defining the alphabet $\Sigma_{n+1} = 2^{\mathrm{AP}_{n+1}}$. We want to show the existence of an $O(n)$-sized LTL formula with past such that any equivalent pure future LTL formula is of size $\Omega(2^n)$.

First consider the following LTL formula of exponential size:

$$\bigwedge_{S \subseteq \mathrm{AP}_n} \Bigg((\bigwedge_{p_i \in S} p_i \wedge \bigwedge_{p_j \notin S} \neg p_j \wedge p_n) \Rightarrow \mathsf{G}((\bigwedge_{p_i \in S} p_i \wedge \bigwedge_{p_j \notin S} \neg p_j) \Rightarrow p_n) \\ \wedge (\bigwedge_{p_i \in S} p_i \wedge \bigwedge_{p_j \notin S} \neg p_j \wedge \neg p_n) \Rightarrow \mathsf{G}((\bigwedge_{p_i \in S} p_i \wedge \bigwedge_{p_j \notin S} \neg p_j) \Rightarrow \neg p_n) \Bigg) \qquad (\varphi_n)$$

1. Describe which words of Σ_{n+1}^ω are the models of φ_n.

2. Can an LTL formula with past modalities check whether it is at the initial position of a word?

3. Provide an LTL formula with past ψ_n of size $O(n)$ initially equivalent to φ_n.

4. Consider the language $L_n = \{\sigma \in \Sigma_{n+1}^\omega \mid \sigma \models \mathsf{G}\varphi_n\}$. We want to prove that any generalized Büchi automaton that recognizes L_n requires at least 2^{2^n} states.

   For this we fix a permutation $a_0 \cdots a_{2^n-1}$ of the symbols in Σ_n and we consider all the different subsets $K \subseteq \{0, \ldots, 2^n - 1\}$. For each K we consider the word

$$w_K = b_0 \cdots b_{2^n-1}$$

in $\Sigma_{n+1}^{2^n}$, defined for each i in $\{0, \ldots, 2^n - 1\}$ by

$$\begin{aligned} b_i &= a_i && \text{if } i \in K \\ b_i &= a_i \cup \{p_n\} && \text{otherwise.} \end{aligned}$$

Thus K is the set of positions of w_K where p_n does not hold.

Using the w_K for different values of K, prove that any generalized Büchi automaton for $\mathsf{G}\varphi_n$ requires at least 2^{2^n} states.

5. Conclude using the fact that any pure future LTL formula φ can be given a generalized Büchi automaton with at most $2^{|\varphi|}$ states.