Two-variables First-Order Logic

Home Assignment 1

To hand in before or on October 13, 2015.

		29	30				
October				1	2	3	4
	5	6	7	8	9	10	11
	12	13					

Electronic versions (PDF only) can be sent by email to ⟨chatain@lsv.ens-cachan.fr⟩ and ⟨schmitz@lsv.ens-cachan.fr⟩, paper versions should be handed in on the 13th or put in a mailbox at LSV, ENS Cachan. **No delays**.

The assignment contains a mandatory part (Exercises 1 to 5) and two different options: *either* do the more theoretical Exercices 6 and 7, *or* do the more practical Exercise 8.

This assignment is the occasion of investigating the expressive power of the *two-variables fragment* of $\mathrm{FO}(\mathrm{AP},<,+1)$ on infinite words (i.e. over the $(\mathbb{N},<)$ time flow). The main result we are going to establish is that this logic is equivalent to $\mathrm{LTL}(\mathrm{AP},\mathsf{Y},\mathsf{X},\mathsf{P},\mathsf{F})$, a linear temporal logic with *past* connectives.

1 First-Order Logic with Two Variables

Fix a countable set AP of atomic propositions. The syntax of $\mathrm{FO}^2(\mathrm{AP},<,+1)$ is the same as that of $\mathrm{FO}(\mathrm{AP},<,+1)$, but restricted to a set of variables $\mathcal{X}$ of cardinality $|\mathcal{X}|=2$:

$$\varphi ::= \top \mid p(x) \mid x=y \mid x<y \mid x+1=y \mid \neg\varphi \mid \varphi\vee\varphi \mid \exists x\,.\,\varphi$$

where p ranges over AP and $x \neq y$ range over $\mathcal{X}$. An *atomic formula* is one of $\{\top, p(x), x=y, x<y, x+1=y \mid p\in\mathrm{AP}, x\neq y\in\mathcal{X}\}$. We write $\varphi(x)$ for a formula with at most one free variable x and $\varphi(x,y)$ for a formula with at most two free variables x and y. We shall use the usual definitions for the dual connectives: $\bot \overset{\text{def}}{=} \neg\top$, $\varphi\wedge\psi \overset{\text{def}}{=} \neg(\neg\varphi\vee\neg\psi)$, $\forall x\,.\,\varphi \overset{\text{def}}{=} \neg\exists x\,.\,\neg\varphi$, and for implication $\varphi\Rightarrow\psi \overset{\text{def}}{=} \neg\varphi\vee\psi$. The *quantifier depth* $\mathrm{qd}(\varphi)$ of a formula φ measures its maximal number of nested quantifiers, and is defined inductively by $\mathrm{qd}(\top)=\mathrm{qd}(p(x))=\mathrm{qd}(x=y)=\mathrm{qd}(x<y)=\mathrm{qd}(x+1=y)\overset{\text{def}}{=}0$, $\mathrm{qd}(\neg\varphi)\overset{\text{def}}{=}\mathrm{qd}(\varphi)$, $\mathrm{qd}(\varphi\vee\psi)\overset{\text{def}}{=}\max(\mathrm{qd}(\varphi),\mathrm{qd}(\psi))$, and $\mathrm{qd}(\exists x\,.\,\varphi)\overset{\text{def}}{=}1+\mathrm{qd}(\varphi)$.

The semantics of $\mathrm{FO}^2(\mathrm{AP},<,+1)$ can be specialised in the case of infinite words w in Σ^ω, where $\Sigma\overset{\text{def}}{=}2^{\mathrm{AP}}$. We write $w(i)$ for the ith letter of w, which is a set of atomic propositions. A word w satisfies a formula φ under an assignment $\nu\colon\mathcal{X} \to \mathbb{N}$, noted $w \models_\nu \varphi$, in the following inductive cases:

$$\begin{array}{llll}
w \models_\nu \top & \text{always,} & w \models_\nu p(x) & \text{iff } p \in w(\nu(x)) \;, \\
w \models_\nu x = y & \text{iff } \nu(x) = \nu(y) \;, & w \models_\nu x < y & \text{iff } \nu(x) < \nu(y) \;, \\
w \models_\nu x + 1 = y & \text{iff } \nu(x) + 1 = \nu(y) \;, & w \models_\nu \neg\varphi & \text{iff } w \not\models_\nu \varphi \;, \\
w \models_\nu \varphi \vee \psi & \text{iff } w \models_\nu \varphi \text{ or } w \models_\nu \psi \;, & w \models_\nu \exists x \,.\, \varphi & \text{iff } \exists i \in \mathbb{N} \,.\, w \models_{\nu[x\mapsto i]} \varphi \;,
\end{array}$$

where $\nu[x \mapsto i]$ denotes the assignment with $\nu[x \mapsto i](x) = i$ and $\nu[x \mapsto i](y) = \nu(y)$ for all $y \neq x$ in $\mathcal{X}$.

Exercise 1 (Basic Formulæ)**.** Show that the formulæ '$x + 2 \leq y$' can be expressed in $\mathrm{FO}^2(<, \mathrm{AP}, +1)$, where the intended semantics are

$$w \models_\nu x + 2 \leq y \qquad\qquad \text{iff } \nu(x) + 2 \leq \nu(y) \;.$$

2 Linear Temporal Logic with Past

The first step of the proof of our main result relies on an extension of linear temporal logic with *past modalities*. More precisely, we are going to work with the abstract syntax

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Y}\,\varphi \mid \mathsf{X}\,\varphi \mid \mathsf{P}\,\varphi \mid \mathsf{F}\,\varphi$$

where p ranges over AP. The Y and P modalities stand respectively for 'yesterday' and 'in the past'. The dual connectives are defined as $\bot \overset{\text{def}}{=} \neg\top$, $\varphi \wedge \psi \overset{\text{def}}{=} \neg(\neg\varphi \vee \neg\psi)$, $\mathsf{H}\,\varphi \overset{\text{def}}{=} \neg\,\mathsf{P}\,\neg\varphi$ for 'historically φ,' and $\mathsf{G}\,\varphi \overset{\text{def}}{=} \neg\,\mathsf{F}\,\neg\varphi$ for 'globally φ.'

Here are the semantics of $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ over infinite words w in Σ^ω at a position i in $\mathbb{N}$:

$$\begin{array}{llll}
w, i \models \top & \text{always,} & w, i \models p & \text{iff } p \in w(i) \;, \\
w, i \models \neg\varphi & \text{iff } w, i \not\models \varphi \;, & w, i \models \varphi \vee \psi & \text{iff } w, i \models \varphi \text{ or } w, i \models \psi \;, \\
w, i \models \mathsf{Y}\,\varphi & \text{iff } i > 0 \text{ and } w, i-1 \models \varphi \;, & w, i \models \mathsf{X}\,\varphi & \text{iff } w, i+1 \models \varphi \;, \\
w, i \models \mathsf{P}\,\varphi & \text{iff } \exists j \,.\, 0 \leq j \leq i \text{ and } w, j \models \varphi \;, & w, i \models \mathsf{F}\,\varphi & \text{iff } \exists j \,.\, i \leq j \text{ and } w, j \models \varphi \;.
\end{array}$$

Exercise 2 (Specification with Past Modalities)**.** Consider the set of atomic propositions $\mathrm{AP} \overset{\text{def}}{=} \{r, g\}$ standing for 'resource requested' and 'resource granted.' We want to write a specification for a service that receives requests to some resource and grants them.

1. Write an $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ formula expressing that 'every requested resource is eventually granted.'

2. Write an $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ formula expressing that 'every granted resource was requested beforehand.'

3. Write an $\mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$ formula for the previous specification, i.e. using only the 'next' and 'until' modalities.

In the following, we say that an $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ formula φ and an $\mathrm{FO}^2(\mathrm{AP}, <, +1)$ formula $\psi(x)$ with one free variable are *equivalent* if, for all infinite words w in Σ^ω and all positions i in $\mathbb{N}$, $w, i \models \varphi$ if and only if $w \models_{\nu[x\mapsto i]} \psi(x)$. We say that they are *initially equivalent* if for all w in Σ^ω, $w, 0 \models \varphi$ if and only if $w \models_{\nu[x\mapsto 0]} \psi(x)$. Note that equivalence implies initial equivalence.

Exercise 3 (Standard Translation)**.** Show that for any $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ formula φ, we can compute an equivalent $\mathrm{FO}^2(\mathrm{AP}, <, +1)$ formula $\mathrm{ST}_x(\varphi)$ of $O(|\varphi|)$ size.

3 From FO^2 to Past LTL

The main objective in this section is to prove the following theorem:

Theorem 1. *For any* $\mathrm{FO}^2(\mathrm{AP}, <, +1)$ *formula* $\varphi(x)$*, one can compute an equivalent* $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ *formula* $\tau(\varphi)$ *of* $2^{O(|\varphi|\cdot \mathrm{qd}(\varphi)+1)}$ *size.*

Exercise 4 (Guarded Formulæ)**.** Let us warm up with a case where the translation into Past LTL is relatively straightforward. An *order formula* is a formula from the set $\mathcal{O} \overset{\text{def}}{=} \{x+2 \leq y, x+1 = y, x = y, y+1 = x, y+2 \leq x\}$. An *order guarded formula* is a formula of the form $\exists y \,.\, o(x,y) \wedge \varphi(y)$ where $o(x,y)$ is an order formula.

Claim 4.1. For any $\mathrm{FO}^2(\mathrm{AP}, <, +1)$ formula $\varphi(y)$ with an equivalent $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ formula $\tau(\varphi)$, and any order formula $o(x,y)$, one can compute an $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ formula $\tau(\exists y \,.\, o(x,y) \wedge \varphi(y))$ of $O(|\tau(\varphi)|)$ size.

Prove Claim 4.1.

Exercise 5 (From FO^2 to Past LTL)**.** Let us prove Theorem 1 by an 'outer' recurrence on $\mathrm{qd}(\varphi)$, which uses a nested 'inner' structural induction on φ:

1. For the base case: provide a suitable $\tau(\varphi)$ when $\mathrm{qd}(\varphi) = 0$.

For the recurrence step where $\mathrm{qd}(\varphi) > 0$, we proceed by inner induction on the structure of φ: the base case of the induction is $\varphi(x) = \exists y.\varphi'(x,y)$. Although $\mathrm{qd}(\varphi') < \mathrm{qd}(\varphi)$, we cannot apply the outer recurrence hypothesis directly on $\varphi'(x,y)$ since it has two free variables. We cannot apply Claim 4.1 directly either since $\varphi'(x,y)$ might not be an order guarded formula. Nevertheless, we are going to exhibit an $\mathrm{FO}^2(\mathrm{AP}, <)$ equivalent to φ, where Claim 4.1 can be applied.

2. Show that φ is equivalent to

$$\bigvee_{o\in\mathcal{O}} \exists y \,.\, o(x,y) \wedge \varphi'(x,y) \,.$$

We still cannot apply the outer recurrence hypothesis on $\varphi'(x,y)$ because it has two free variables. Let $\bar{\psi} = \psi_0, \ldots, \psi_{n-1}$ be a finite sequence of formulæ. A *Boolean formula over* $\bar{\psi}$ is a formula defined by the abstract syntax

$$\beta ::= \top \mid \neg\beta \mid \beta \vee \beta \mid \psi_i$$

where $0 \leq i < n$; if the formulæ in $\bar{\psi}$ are $\mathrm{FO}^2(\mathrm{AP}, <)$ formulæ, then $\beta(\bar{\psi})$ is also an $\mathrm{FO}^2(\mathrm{AP}, <)$ formula. Then we obtain an equivalent formula

$$\varphi \equiv \bigvee_{o \in \mathcal{O}} \exists y \,.\, o(x,y) \wedge \beta(\bar{\alpha}, \bar{\delta}, \bar{\gamma})$$

for some Boolean formula β, where

- $\bar{\alpha} = \alpha_0(x,y), \ldots, \alpha_{r-1}(x,y)$ are atomic subformulæ of φ' with two free variables—i.e. of the form $x = y$, $x < y$, $y < x$, $x = y+1$, or $y = x+1$—,
- $\bar{\delta} = \delta_0(x), \ldots, \delta_{s-1}(x)$ are atomic or existential subformulæ of φ' with only free x, and
- $\bar{\gamma} = \gamma_0(y), \ldots, \gamma_{t-1}(y)$ are atomic or existential subformulæ of φ' with only free y.

Among these, the $\alpha_i(x,y)$ and $\delta_i(x)$ subformulæ are preventing us from applying Claim 4.1.

3. Let us first focus on the $\alpha_i(x,y)$ formulæ. Show that for all $o(x,y)$ in $\mathcal{O}$ and i, $o(x,y) \Rightarrow (\alpha_i(x,y) \equiv \alpha_i^o)$ is valid for some α_i^o in $\{\top, \bot\}$. This allows to rewrite φ as
$$\varphi \equiv \bigvee_{o \in \mathcal{O}} \exists y \,.\, o(x,y) \wedge \beta(\bar{\alpha}^o, \bar{\delta}, \bar{\gamma}) \,.$$
4. Let us turn to the $\delta_i(x)$ subformulæ. Show that φ can be rewritten as
$$\varphi \equiv \bigvee_{\bar{\delta}' \in \{\bot,\top\}^s} \left(\bigwedge_{i<s} (\delta_i \equiv \delta_i') \wedge \bigvee_{o \in \mathcal{O}} \exists y \,.\, o(x,y) \wedge \beta(\bar{\alpha}^o, \bar{\delta}', \bar{\gamma}) \right) ,$$
that is, where each $\bar{\delta}'$ is an s-tuple mixing $\top$ and $\bot$ formulæ.
5. Complete the construction of $\tau(\varphi)$ for $\varphi = \exists y.\varphi'(x,y)$.
6. Complete the proof and check that your formula $\tau(\varphi)$ has an appropriate size.

Recall that you have a choice between answering both Exercise 6 and Exercise 7, or answering Exercise 8.

4 Succinctness

The goal of this section is to show that, in spite of its exponential complexity, the previous translation from FO^2 to Past LTL is essentially optimal:

Theorem 2. *There is a family* $(\varphi_n)_n$ *of* $\mathrm{FO}^2(\mathrm{AP}, <, +1)$ *sentences each of size* $O(n)$ *such that the smallest equivalent* $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ *formula have size* $2^{\Omega(n)}$.

As a preliminary, we are going to consider a family of ω-regular languages $(L_n)_n$ that require very large Büchi automata.

Exercise 6 (Succinctness of Büchi Automata)**.** Consider for each n the set of atomic propositions $\mathrm{AP}_{n+1} \stackrel{\text{def}}{=} \{p_0, \dots, p_n\} = \mathrm{AP}_n \uplus \{p_n\}$ and the corresponding alphabet $\Sigma_{n+1} = 2^{\mathrm{AP}_{n+1}}$. Given a symbol a in Σ_{n+1}, we write $a|_n \stackrel{\text{def}}{=} \{p_i \in a \mid i < n\}$ for its projection on Σ_n.

We define for each n the language

$$L_n \stackrel{\text{def}}{=} \{w \in \Sigma_{n+1}^\omega \mid \forall i, j \in \mathbb{N} \,.\, w(i)|_n = w(j)|_n \text{ implies } w(i) = w(j)\} \,.$$

We want to prove that any generalised Büchi automaton $\mathcal{A}$ for L_n requires at least 2^{2^n} states.

Fix a permutation $a_0 \cdots a_{2^n-1}$ of the symbols in Σ_n. For every subset $K \subseteq \{0, \dots, 2^n - 1\}$, define the finite word $w_K \stackrel{\text{def}}{=} b_0 \cdots b_{2^n-1}$ over Σ_{n+1}, where b_i is defined for each $0 \leq i \leq 2^n - 1$ by

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

Hence K is the set of positions in w_K where p_n does not appear.

1. Show that, for all K, the infinite word $(w_K)^\omega$ is in L_n.
2. Show that, for all $K \neq K'$, the infinite word $w_K(w_{K'})^\omega$ is not in L_n.
3. Deduce that any generalised Büchi automaton $\mathcal{A}$ for L_n has at least 2^{2^n} states.

Exercise 7 (Succinctness of FO^2)**.**

1. Show that each language L_n from the previous exercise can be expressed by an $\mathrm{FO}^2(\mathrm{AP}_{n+1}, <, +1)$ sentence φ_n of size $O(n)$, i.e. $L_n = \{w \in \Sigma_{n+1}^\omega \mid w \models_\nu \varphi_n\}$.
2. Using the fact that any $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ formula φ has an equivalent generalised Büchi automaton $\mathcal{A}_\varphi$ of size $2^{O(|\varphi|)}$, i.e. such that $L(\mathcal{A}_\varphi) = \{w \in 2^{\mathrm{AP}} \mid w, 0 \models \varphi\}$, conclude that any $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{X}, \mathsf{P}, \mathsf{F})$ formula equivalent to φ_n must be of size $2^{\Omega(n)}$.

5 CTL$_2$ Model-Checking

We introduce a variant of CTL allowing one to express that a property must be satified at some/every *even* position along a path.

The formulas of the logic CTL$_2$ are defined by the following grammar:

$$\varphi ::= \top \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E\,X}\,\varphi \mid \mathsf{E\,G}_2\,\varphi \mid \mathsf{E}\,\varphi\,\mathsf{U}_2\,\varphi$$

where p ranges over a set AP of atomic propositions.

The intuition behind the $\mathsf{E\,G}_2$ and $\mathsf{E\,U}_2$ modalities is that they only care about the *even* positions along runs. One can interpret a CTL$_2$ formula φ by structural induction directly on a state s of a transition system $\mathfrak{S} = (S, T, I, \mathrm{AP}, \ell)$ as follows:

$\mathfrak{S}, s \models \top$ always ,

$\mathfrak{S}, s \models p$ iff $p \in \ell(s)$,

$\mathfrak{S}, s \models \neg\varphi$ iff $\mathfrak{S}, s \not\models \varphi$,

$\mathfrak{S}, s \models \varphi \vee \psi$ iff $\mathfrak{S}, s \models \varphi$ or $\mathfrak{S}, s \models \psi$,

$\mathfrak{S}, s \models \mathsf{E\,X}\,\varphi$ iff $\exists s' \in S \,.\, s \to s'$ and $\mathfrak{S}, s' \models \varphi$,

$\mathfrak{S}, s \models \mathsf{E\,G}_2\,\varphi$ iff $\exists \sigma = s_0 s_1 \cdots \in S^\omega$ a run starting at $s_0 = s$ s.t. $\forall i \,.\, i \bmod 2 = 0$ implies $\mathfrak{S}, s_i \models \varphi$,

$\mathfrak{S}, s \models \mathsf{E}\,\varphi\,\mathsf{U}_2\,\psi$ iff $\exists \sigma = s_0 s_1 \cdots \in S^\omega$ a run starting at $s_0 = s$ such that $\exists j \,.\, j \bmod 2 = 0$ and $\mathfrak{S}, s_j \models \psi$, and $\forall i \,.\, i \bmod 2 = 0$ and $i < j$ imply $\mathfrak{S}, s_i \models \varphi$.

Exercise 8 (Implementing a CTL$_2$ Model-Checker)**.**

1. Show that every (classical) CTL formula can be expressed as a CTL$_2$ formula.

2. Write a model-checker for CTL$_2$, i.e. a program that, given a transition system $\mathfrak{S}$, a state s and a CTL$_2$ formula φ, decides whether $\mathfrak{S}, s \models \varphi$.

   You can program in your favorite programming language and you are free to choose the appropriate data structures for formulæ and transition systems. No graphical user interface is requested. It is even acceptable for the models and formulæ to be defined directly in the code, provided they can easily be edited. Your code must come with a few well-chosen examples.

   Your clear, readable, amply annotated code must be sent by email to ⟨chatain@lsv.ens-cachan.fr⟩ and ⟨schmitz@lsv.ens-cachan.fr⟩.