université PARIS-SACLAY | FACULTÉ DES SCIENCES D'ORSAY

27/02/2024

Automates et Applications
M1–MPRI Orsay

Exam

Instructions

The exam totals 20 points and last for two hours. Personal lectures notes (handwritten or typeset) are authorized. The use of electronic devices is prohibited. You may answer in French or in English. The points for each exercise is given as an indication of its difficulty and may slightly change in the final grading.

1 Regular tree languages (12 points)

1.1 *Caterpillar expressions* (8 points)

We study a variant of *caterpillar expressions* (Brüggemann-Klein et Wood, 2000). Let Σ be a fixed ranked alphabet. We consider the set Δ of operations defined as :

$$\Delta = \{\mathsf{label}(a), \mathsf{is\_root}, \mathsf{is\_child}_n, \uparrow, \downarrow_n\}$$

where $a \in \Sigma$ et $1 \leq n \leq \mathsf{max}\{\ |a| \text{ for } a \in \Sigma\}$. *caterpillar expressions* are regular expression over elements of Δ. For instance, if $\Sigma = \{f^2, g^2, a^0, b^0\}$, we can defined a carterpilar expression r_0 as:

$$r_0 = \mathsf{is\_root} \cdot \mathsf{label}(f) \cdot \downarrow_1^* \cdot \mathsf{label}(a)$$

We limit ourseleves to union ($|$), concatenation ($\cdot$) and Kleene star (*) in regular expressions.

The semantics of these expressions is given by for tree t and a path π by a judgement

$$\pi \vdash_t r : \pi'$$

with $r \in \mathsf{Regexp}(\Delta)$ and $\pi, \pi' \in \mathsf{dom}(t)$.

This judgement means that in the three t, one can go from the path π to the path π' by executing r. The rules to derive this judgements are the following, grouped in three categories:

tests:

$$\epsilon \vdash_t \mathsf{is\_root} : \epsilon \qquad \frac{t(\pi) = a}{\pi \vdash_t \mathsf{label}(a) : \pi} \qquad \frac{\exists \pi' \in \mathsf{dom}(t).\ t.q. \quad \pi' n \equiv \pi}{\pi \vdash_t \mathsf{is\_child}_n : \pi}$$

moves:

$$\frac{\pi, \pi n \in \mathsf{dom}(t)}{\pi \vdash_t \downarrow_n : \pi n} \qquad \frac{\pi, \pi n \in \mathsf{dom}(t)}{\pi n \vdash_t \uparrow : \pi}$$

regexp:

$$\frac{\pi \vdash_t r_1 : \pi' \quad \pi' \vdash_t r_2 : \pi''}{\pi \vdash_t r_1 \cdot r_2 : \pi''} \qquad \frac{\pi \vdash_t r_1 : \pi'}{\pi \vdash_t r_1 | r_2 : \pi'} \qquad \frac{\pi \vdash_t r_2 : \pi'}{\pi \vdash_t r_1 | r_2 : \pi'} \qquad \pi \vdash_t r^* : \pi \qquad \frac{\pi \vdash_t r : \pi' \quad \pi' \vdash_t r^* : \pi''}{\pi \vdash_t r^* : \pi''}$$

We say that a *caterpillar expression* r recognizes a tree t if and only if:

$$\exists \pi \in \mathsf{dom}(t).\ t.q. \quad \epsilon \vdash_t r : \pi$$

In other words, a tree is recognized by r if one can reach any node starting from the root without being stuck. We can remark that it is not necessary to requires that one "returns" to the root since a non stuck caterpillar expression r can always be completed by $r \cdot (\uparrow \ | \ \mathsf{is\_root})^*$.

Lastly we call the language recognized by an expression r is the set $L_r = \{t \in \mathsf{Tree}(\Sigma) \mid \exists \pi. \epsilon \vdash_t r : \pi\ \}$.

Remark : the name *caterpillar expressions* comes from the fact that such expressions are little programs that go up and down the tree, much like a caterpillar.

Questions In what follows, we fix $\Sigma = \{f^2, g^2, a^0, b^0\}$.

1.1.1 (1.5 point) Give an MSO formula recognizing the same language as the expression

$$r_0 = \mathsf{is\_root} \cdot \mathsf{label}(f) \cdot \downarrow_1^* \cdot \mathsf{label}(a)$$

You can adapt the definition of $\mathsf{descendant}(x, Y)$ from the lecture for the part $\downarrow_1^*$ (we only ask for the MSO formula, not to show that it is equivalent).

1.1.2 (1.5 point) Give a bottom-up deterministic tree automaton that recognizes r_0 (we only ask for the automaton).

1.1.3 (4.5 points) Consider an arbitrary caterpillar expression r. Pour each of the following cases, give an MSO formula $\phi(x, Y)$ such that for each tree t and for all path $\pi \in \mathsf{dom}(t)$ and $P \subseteq \mathsf{dom}(t)$ we have

$$t, \{x \mapsto \pi\}, \{Y \mapsto P\} \vdash_{MSO} \phi(x, Y)$$

if and only if

$$\forall \pi' \in P,\ \pi \vdash_t r : \pi'$$

where $\vdash_{MSO}$ is the semantics of MSO formula from the lecture (we only ask for the corresponding formulæ, no equivalence proof).

(a) (0.5 point) $r \equiv \mathsf{label}(f)$

(b) (0.5 point) $r \equiv \uparrow$

(c) (0.5 point) $r \equiv \downarrow_n$

(d) (0.5 point) $r \equiv \mathsf{is\_root}$

(e) (0.5 point) $r \equiv \mathsf{is\_child}_n$

(f) (0.5 point) $r \equiv r_1 \mid r_2$ by calling $\phi_i(x, Y)$ the formula corresponding to r_i

(g) (0.5 point) $r \equiv r_1 \cdot r_2$ by calling $\phi_i(x, Y)$ the formula corresponding to r_i correspondant à r_i

(h) (1 point) $r \equiv r_0^*$ by calling $\phi_0(x, Y)$ the formula corresponding to r_0.

1.1.4 (0.5 point) Consider a fixed Σ. We recall that Rec is the set of tree languages recognizable by a bottom-up tree automaton. We call Cat the set of tree languages recognizable by a *caterpillar expression*. Which property is proved by the previous question?

- $\mathsf{Rec} \subseteq \mathsf{Cat}$
- $\mathsf{Rec} = \mathsf{Cat}$
- $\mathsf{Cat} \subseteq \mathsf{Rec}$

Give a brief justification.

1.2 Properties of tree languages (4 points)

Consider the ranked alphabet $\Sigma = \{N^2, R^2, \#^0\}$. We recall that a *red-black* tree is a type of binary search tree with the following properties:

(i) Each internal node is either red (R) or black (N)

(ii) Leves ($\#$) are considered black

(iii) A red node has two black children

(iv) All paths from the root to a leaf have the same number of black nodes

It is easy to show that the set of trees *(i)*, *(ii)* and *(iii)* for which is a regular tree languages (one can write an automaton that accepts only such trees). Show that adding property *(iv)* makes the languages non-regular.

2 ω-regular languages and temporal logic (8 points)

2.1 ω-régularity (3 points)

Consider the set L of infinite words over $\Sigma = \{a, b, c\}$ for which the exists at least a pair of letters a and b separated by three letters that are not b. Give a *deterministic* Büchi automaton that recognizes L.

2.2 LTL (5 points)

2.2.1 Consider two users Alice and Bob and a printer. Each user can perform one of three actions described by three atomic properties req_U, use_U, rel_U, for $U \in \{A, B\}$. The properties are:

req_U**:** user U requests to use the printer

use_U**:** user U is using the printer

rel_U**:** user U releases the printer

Give LTL formulæ to specify the following properties:

(a) (0.5 point) Mutual exclusion : A and B never use the printer at the same time

(b) (2.5 points) Correct access :

- a printer must be requested before use
- once requested by a user, the printer must be used and then released before being requested again by the same user
- once released by a user, the printer is never released again before being used
- a printer can be used finitely many times ($\equiv$ in use at several instants) after having been asked and before being released

You can use macros for each of the points above, parameterized by the user name before giving the final formula as the conjunction of all the above for both users.

2.2.2 (2 points) Show that $\mathcal{G}(\phi \Rightarrow \psi) \quad \Rightarrow \quad (\mathcal{G}\phi) \Rightarrow (\mathcal{G}\psi)$. We recall that:

- $\mathcal{F}\phi \equiv \top \mathcal{U} \phi$
- $\mathcal{G}\phi \equiv \neg \mathcal{F} \neg \phi$