Tree Automata and Applications

Exercise session 2 Solutions

Luc Lapointe
luc.lapointe@ens-paris-saclay.fr
home.lmf.cnrs.fr/LucLapointe/

Exercise 1 - Quizz

Be precise in your answers.

1. What is the most expressive, top-down or bottom-up?
2. What are the general techniques to prove that an automaton recognizes a language L?
3. What is the minimal number of states required by a bottom-up NFTA to recognize the language made of a single term with one node and two leaves?
4. Can the language of trees such that all branches contain a specific symbol f be recognized by a NFTA?
5. Can it be recognized by a top-down DFTA?

Solution

1. In the deterministic case, bottom-up is more expressive. But only if there exist a symbol of arity greater than 1.
2. Technique 1: Prove by induction what is the language of each state (the language recognized by $\mathcal{A}$ when starting in this specific state).
   Technique 2: Closure properties.
3. Three if the leaves have distinct labels, two otherwise.
   Proof that two is not enough with $L = \{f(a, b)\}$: Assume $\mathcal{A}$ recognizes L with states $Q = \{q_I, q_1\}$, and $q_I \in I$.
   Case $q_I(a) \longrightarrow a$ or $q_I(b) \longrightarrow b$ Then a or b is recognized: contradiction.
   Case $q_1(a) \longrightarrow a$ and $q_1(b) \longrightarrow b$ Then $f(b, a)$ is recognized: contradiction.
   Case everything else Then $f(a, b)$ is not recognized: contradiction.
4. Yes. Use a bottom-up NFTA. Whenever a f is met, set a specific win state. For every node, propagate the win state only if all children are winners.
5. In general, no. If f is the only leaf symbol, or if all symbols have arity lower than 1, then the language is Σ^*, so yes.

Exercise 2 - Tree Automata

Let $\mathcal{F} = \{f(2), g(2), a(0)\}$. Give NFTAs for the following languages:

1. $\{t \in \mathcal{F} \mid \text{on some branch in } t \text{ there are two consecutive occurrences of } f\}$
2. $\{t \in \mathcal{F} \mid \text{on all branches in } t \text{ there are two consecutive occurrences of } f\}$

Correct automata

I don't give the proofs, as they are similar to the ones of exercise session 1. Do not hesitate to send me an email if needed.

1. $Q = \{q_0, q_1, q_\top\}, I = \{q_0\}$. q_1 means that the first f has already been met, and $q_\top$ means that no more f is needed. Transitions as follow:

$$q_0(f(u,v)) \longrightarrow f(q_1(u), q_\top(v)) \mid f(q_\top(u), q_1(v)) \mid f(q_0(u), q_\top(v)) \mid f(q_\top(u), q_0(v))$$

$$q_0(g(u,v)) \longrightarrow g(q_0(u), q_\top(v)) \mid g(q_\top(u), q_0(v))$$

$$q_\top(f(u,v)) \longrightarrow f(q_\top(u), q_\top(v)) \qquad q_\top(g(u,v)) \longrightarrow g(q_\top(u), q_\top(v)) \qquad q_\top(a) \longrightarrow a$$

$$q_1(f(u,v)) \longrightarrow f(q_\top(u), q_\top(v))$$

2. $Q = \{q_0, q_1, q_\top\}, I = \{q_0\}$. q_1 means that the first f has already been met, and $q_\top$ means that no more f is needed. Transitions as follow:

$$q_0(f(u,v)) \longrightarrow f(q_1(u), q_1(v)) \mid f(q_1(u), q_0(v)) \mid f(q_0(u), q_1(v)) \mid f(q_0(u), q_0(v))$$

$$q_0(g(u,v)) \longrightarrow g(q_0(u), q_0(v))$$

$$q_\top(f(u,v)) \longrightarrow f(q_\top(u), q_\top(v)) \qquad q_\top(g(u,v)) \longrightarrow g(q_\top(u), q_\top(v)) \qquad q_\top(a) \longrightarrow a$$

$$q_1(f(u,v)) \longrightarrow f(q_\top(u), q_\top(v))$$

Exercise 3 - Regular expressions

1. Describe the language recognized by the following regular expressions over alphabet $\mathcal{F} = \{f(2), g(2), a(0), b(0)\}$:

$$E = f(\square_1, \square_1)^{*\square_1} \cdot \square_1 \left[g(\square_2, \square_2) \cdot \square_2 (f(\square_1, \square_1))^{*\square_1} \cdot \square_1 b\right]$$

Detailed solution

We will describe the language of parts of this expression before describing the whole. Let

$$E_1 \overset{\text{def}}{=} (f(\square_1, \square_1))^{*\square_1} \cdot \square_1 b$$

$$E_2 \overset{\text{def}}{=} g(\square_2, \square_2) \cdot \square_2 E_1$$

so that

$$E = f(\square_1, \square_1)^{*\square_1} \cdot \square_1 E_2$$

The language of E_1 is the set of trees over $\mathcal{F}' \stackrel{\text{def}}{=} \{f(2), b(0)\}$. To prove this, prove by induction on n that $(f(\square_1, \square_1))^{n,\square_1} \cdot_{\square_1} b$ is exactly the set of trees of height at most n over $\mathcal{F}'$.

The language of E_2 is the set of trees of $\mathcal{F}$ with g at their root and nowhere else, and no a at their leaves. This proof follows directly from the language of E_1.

The language of E is the set of trees over $\mathcal{F}'' \stackrel{\text{def}}{=} \{f(2), g(2), b(0)\}$ with exactly one g on each path from the root to a leaf.

To prove this, prove by induction on $n \geq 1$ that

$$(f(\square_1, \square_1))^{n,\square_1} \cdot_{\square_1} E_2 \setminus (f(\square_1, \square_1))^{n-1,\square_1} \cdot_{\square_1} E_2$$

is exactly the set of trees over $\mathcal{F}''$ of height n with exactly one g on each path from the root to a leaf.

2. Give a regular expression for:
   1. the set $T(\mathcal{F})$ of all finite trees on alphabet $\mathcal{F} = \{f(2), g(2), a(0), b(0)\}$;
   2. $\{t \in T(\mathcal{F}) \mid t \text{ contains the subtree } f(a, b)\}$ where $\mathcal{F} = \{f(2), a(0), b(0)\}$;
   3. $\{t \in T(\mathcal{F}) \mid \text{the frontier word of } t \text{ contains an infix } ab\}$ with same $\mathcal{F}$.

Correct expressions

1. $(f(\square, \square) + g(\square, \square))^{*\square} \cdot_{\square} (a + b)$
2. $(f(\square_1, \square_2) + f(\square_2, \square_1))^{*\square_1} \cdot_{\square_1} f(a, b) \cdot_{\square_2} (f(\square, \square)^{*\square} \cdot_{\square} (a + b))$
3. Let $E_{\mathcal{F}}$ describing the language of all trees over $\mathcal{F}$:

$$E_{\mathcal{F}} \stackrel{\text{def}}{=} f(\square, \square)^{*\square} \cdot_{\square} (a + b)$$

Let E_a describing the language of trees whose frontier word ends with a:

$$E_a \stackrel{\text{def}}{=} \left(f(\square_{\mathcal{F}}, \square)^{*\square} \cdot_{\square} a\right) \cdot_{\square_{\mathcal{F}}} E_{\mathcal{F}}$$

Let E_b describing the language of trees whose frontier word starts with b:

$$E_b \stackrel{\text{def}}{=} \left(f(\square, \square_{\mathcal{F}})^{*\square} \cdot_{\square} b\right) \cdot_{\square_{\mathcal{F}}} E_{\mathcal{F}}$$

The following expression describes the language of trees whose frontier word contains an infix ab:

$$[f(\square_{\mathcal{F}}, \square) + f(\square, \square_{\mathcal{F}})]^{*\square} \cdot_{\square} f(E_a, E_b) \cdot_{\square_{\mathcal{F}}} E_{\mathcal{F}}$$

Homework - Satisfiability (again)

Let $\mathcal{F} = \{\mathrm{and}(2), \mathrm{or}(2), \mathrm{not}(1), 0(0), 1(0), x(1), s(1), z(0)\}$, i.e. we now handle an arbitrary number of variables instead of a fixed one (encoding x_2 as $x(s(s(z)))$). The same variable may appear several times in a formula, and should be evaluated consistently.

1. Is the set of well-formed formulae using this syntax recognizable by an NFTA?
2. Is the set of satisfiable formulae using this syntax recognizable by an NFTA?