Automates d’arbre

TD n°1 : Recognizable Tree Languages and Finite Tree Automata

September 19, 2019

Exercise 1: First constructions of Tree Automatas

Let $\mathcal{F} = \{ f(2), g(1), a(0) \}$. Give a DFTA and a top-down DFTA for the set $G(t)$ of ground instances of the term $t = f(f(a, x), g(y))$ which is defined by:

$G(t) = \{ f(f(u, g(v)) \mid u, v \in T(\mathcal{F}) \}$

Solution:

• top-down DFTA : $Q = \{ q_{f1}, q_{f2}, q_g, q_a, q_\top \}$, $I = \{ q_{f1} \}$ and $\Delta =$
  * $q_{f1}(f(x, y)) \rightarrow f(q_{f2}(x), q_g(y))$
  * $q_{f2}(f(x, y)) \rightarrow f(q_a(x), q_\top(y))$
  * $q_g(g(x)) \rightarrow g(q_\top(x))$
  * $q_a(a) \rightarrow a$
  * $q_\top(f(x, y)) \rightarrow f(q_\top(x), q_\top(y))$
  * $q_\top(g(x)) \rightarrow g(q_\top(x))$
  * $q_\top(a) \rightarrow a$

• DFTA : $Q = \{ q_a, q_f, q_g, q_\top, q_\bot \}$, $F = \{ q_\top \}$ and $\Delta =$
  * $a \rightarrow q_a$
  * $f(q_a, q) \rightarrow q_f$ for all $q \in Q$
  * $g(q) \rightarrow q_g$ for all $q \in Q$
  * $f(q_f, q_g) \rightarrow q_\top$
  * $f(q, q') \rightarrow q_\bot$ for all $(q, q') \neq (q_a, _), (q_f, q_g)$

Exercise 2: What is recognizable by an FTA ?

Are the following tree languages recognizable (by a bottom-up FTA) ?

• $\mathcal{F} = \{ g(1), a(0) \}$ and $L$ the set of ground terms of even height.
• $\mathcal{F} = \{ f(2), g(1), a(0) \}$ and $L$ the set of ground terms of even height.

Solution:

• Yes.
• No. Remark that the pumping lemma does not apply! Assume that it is recognizable by a NFTA with $n$ states. Define :

  $t_n = f(g^{2n+1}(a), g^{2n+2}(a))$

It has height $2n+2$ and so belongs to this language. So there exists an accepting run $\rho$ for $t_n$. By the pigeonhole principle, there exists $k < k'$ such that $r(1.1^k) = r(1.1^{k'})$ and from that we deduce that for all $p \in \mathbb{N}$, the tree

  $t_{n,p} = f(g^{2n+1+p(k'-k)}(a), g^{2n+2}(a))$

also has an accepting run. But $t_{n,2}$ has height $2(n + k' - k) + 1$ which is odd. Contradiction.
Exercise 3: Bottom-up vs Top-down

1) Recall why bottom-up NFTAs, bottom-up DTAs and top-down NFTAs have the same expressiveness.

2) Let $\mathcal{F} = \{ f(2), g(1), a(0) \}$. Give a DFTA and a top-down NFTA for the set $M(t)$ of terms which have a ground instance of the term $t = f(a, g(x))$ as a subterm, i.e. $M(t) = \{ C[f(a, g(u))] \mid C \in \mathcal{C}(\mathcal{F}), u \in T(\mathcal{F}) \}$.

3) Show that NFTAs and top-down DFTAs do not have the same expressiveness.

Solution:

- **top-down NFTA:** $Q = \{ q_0, q_\bot, q_a, q_g \}$, $I = \{ q_0 \}$ and $\Delta =$
  - $q_0(f(x, y)) \rightarrow f(q_\bot(x), q_0(y))$
  - $q_0(f(x, y)) \rightarrow f(q_0(x), q_\bot(y))$
  - $q_\bot(f(x, y)) \rightarrow f(q_\bot(x), q_\bot(y))$
  - $q_\bot(g(x)) \rightarrow g(q_\bot(x))$
  - $q_\bot(a) \rightarrow a$
  - $q_0(g(x)) \rightarrow g(q_0(x))$
  - $q_0(f(x, y)) \rightarrow f(q_a(x), q_g(y))$
  - $q_a(a) \rightarrow a$
  - $q_g(g(x)) \rightarrow g(q_\bot(x))$

- **DFTA:** $Q = \{ q_a, q_g, q_{\top}, q_\bot \}$, $F = \{ q_{\top} \}$ and $\Delta =$
  - $a \rightarrow q_a$
  - $g(q_{\top}) \rightarrow q_{\top}$
  - $g(q) \rightarrow q_g$ with $q \neq q_{\top}$
  - $f(q, q') \rightarrow q_{\top}$ if $(q, q') = (q_a, q_g)$ or $q = q_{\top}$ or $q' = q_{\top}$
  - $f(q, q') \rightarrow q_\bot$ else

Let’s assume $M(t)$ can be recognized by a top-down DFTA $A$. We consider two terms $t_1 = f(t, a)$ and $t_2 = f(a, t)$. $A$ must start with the same transition on both terms, let’s say $q_0(f(x, y)) \rightarrow f(q_\bot(x), q_\bot(y))$. Then, there is an accepting run for $q_R(a)$ because $t_1$ in $M(t)$, and conversely for $q_L(a)$. Finally, $A$ accepts $f(a, a) \notin M(t)$.

***

Bonus exercice : Satisfiability

Let $\mathcal{F} = \{ \text{and}(2), \text{or}(2), \text{not}(1), 0(0), 1(0), x(0) \}$. A ground term over $\mathcal{F}$ can then be viewed as a boolean formula over $x$.

1) Give an NFTA which recognizes the set of satisfiable boolean formulae over $x$.

Let $\mathcal{F} = \{ \text{and}(2), \text{or}(2), \text{not}(1), 0(0), 1(0), x_1(0), \ldots, x_n(0) \}$, i.e we now handle $n$ variables instead of a single one. The same variable may appear several times in a formula, and should be evaluated consistently.

2) Give an NFTA which recognizes the set of satisfiable boolean formulae over $x_1, \ldots, x_n$.

Solution: