Automates d'arbre

TD n°5 : Hedges and Alternation

Charlie Jacomme

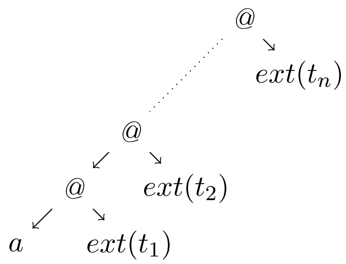
October 16, 2018

Exercise 1 : Extensions

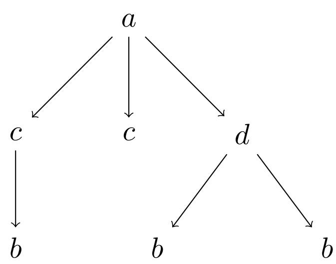
Definition 1 (extension encoding)
Let t be an unranked tree on Σ. Let $\mathcal{F}_{ext}^{\Sigma} = \{@(2)\} \cup \{a(0) \mid a \in \Sigma\}$. We define the ranked tree $ext(t)$ by induction on the size of t by :

- *for $a \in \Sigma$, $ext(a) = a$*
- *if $t = a(t_1, ..., t_n)$ with $n \geq 1$, $ext(t) = @(ext(a(t_1, ..., t_{n-1})), ext(t_n))$*

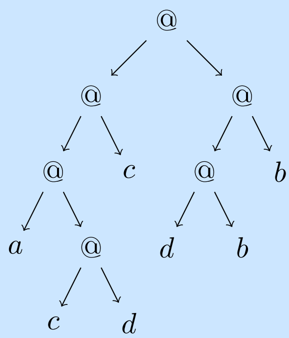
that is $ext(a(t_1, ..., t_n))$ is equal to :



Give the extension encoding of :



Solution:



Exercise 2 : The soundess of the extension

Let L be a language of unranked trees. Prove that L is recognizable by a NFHA iff $ext(L)$ is recognizable by a NFTA.

Solution:

$\Rightarrow$) Let $\mathcal{A} = \langle Q, \Sigma, \Delta, F \rangle$ be a NFHA recognizing L such that there is exactly one rule of the form $a(L_{a,q}) \longrightarrow q$ for all (a,q) and let $B_{a,q} = \langle P_{a,q}, Q, p^0_{a,q}, \delta_{a,q}, F_{a,q} \rangle$ a deterministic automaton recognizing $L_{a,q}$. We construct the expected NFTA this way :

$$\mathcal{A}' = \langle Q', \mathcal{F}^{\Sigma}_{ext}, \Delta', F' \rangle$$

where :

- $Q' = \bigcup\limits_{(a,q)} P_{a,q}$
- $F' = \bigcup\limits_{(a,q) | q \in F} F_{a,q}$
- $\Delta' =$
  - ⋆ $a \longrightarrow p^0_{a,q}$ for all (a,q)
  - ⋆ $@(p, p') \longrightarrow p''$ if $p, p'' \in P_{b,q}$, $p' \in F_{a,q'}$ with $\delta_{b,q}(p, q') = p''$ for some b, q, a, q'

$\Leftarrow$) Let $\mathcal{A} = \langle Q, \mathcal{F}^{\Sigma}_{ext}, F, \Delta \rangle$ be a NFTA recognizing $ext(L)$. We construct the expected NFHA this way :

$$\mathcal{A}' = \langle Q, \Sigma, F, \Delta' \rangle$$

where for all (a,q), $a(R_{a,q}) \longrightarrow q \in \Delta'$ where $R_{a,q}$ is the language recognized by the automaton :

$$B_{a,q} = \langle Q, Q, I_{a,q}, F_{a,q}, \Delta_{a,q} \rangle$$

with :

- $I_{a,q} = \{p \in Q \mid a \longrightarrow p \in \Delta\}$
- $F_{a,q} =$
  - ⋆ $\{q\}$ if $q \in F$ or if there exists q', q'' such that $@(q', q) \longrightarrow q'' \in \Delta$
  - ⋆ $\varnothing$ else
- $\Delta_{a,q} = \{(q_1, q_2, q_3) \mid @(q_1, q_2) \longrightarrow q_3 \in \Delta\}$

Exercise 3 : Complexity

Show that the emptiness problem for NFHA(NFA) is in PTIME.

Solution:

We run the emptiness algorithm used to dertermine the emptiness of a NFTA on the extension encoding of the automaton.

Exercise 4 : SUCH AWA

Definition 2 *If $\mathcal{X}$ is a set of propositional variables, let $\mathbb{B}(\mathcal{X})$ be the set of positive propositional formulae on $\mathcal{X}$, i.e., formulae generated by the grammar $\phi ::= \bot \mid \top \mid x \in X \mid \phi \vee \phi \mid \phi \wedge \phi$.*

Definition 3 *A AWA (Alternating Word Automata) is a tuple $\mathcal{A} = (Q, \Sigma, Q_0, Q_f, \delta)$ where Σ is a finite set (alphabet), Q is a finite set (of states), $Q_0 \subseteq Q$ (initial states), $Q_f \subseteq Q$ (final states) and δ is a function from $Q \times \Sigma$ to $\mathbb{B}(Q)$ (transition function). A run of $\mathcal{A} = (Q, \Sigma, Q_0, Q_f, \delta)$ on a word w is a tree t labelled by $Q \times \mathbb{N}$ such that :*
— *if $w = \varepsilon$, then $t = (q_0, 0)$ with $q_0 \in Q_0$.*
— *if $w = a.w'$, then $t = (q_0, k)(t_1, \ldots, t_n)$ where k is the length of w, $q_0 \in Q_0$ and such that for all i, t_i is a run of w' on $(Q, \Sigma, \{q_i\}, Q_f, \delta)$ for some q_i satisfying $\{q_1, \ldots, q_n\} \models \delta(q_0, a)$.*

Definition 4 *We say that a run is accepting if every leaf of the form $(q, 0)$ satisfies that $q \in Q_f$. Notice that a run may have leaves of the form (q, i) with $i > 0$ if $\varnothing \models \delta(q_0, a)$. Those leaves are considered as 'success' leaves in this semantic. The language of a AWA is the set of words which have an accepting run.*

1. Show how to reduce the emptiness problem for an AWA on a one letter alphabet $\{a\}$ whith formalas that are in positive disjunctive normal form to the emptiness problem of a tree automaton .
2. Show how to reduce the emptiness problem for a tree automaton to the emptiness problem of an AWA on a one letter alphabet $\{a\}$. Conclude on the complexity of the emptiness problem for an AWA on a one letter alphabet.

Solution:

1. Given $\mathcal{A} = (Q, \Sigma, Q_0, Q_f, \delta)$ an AWA we construct an NFTA of the form $(Q, \{f_k(k) \mid 0 \leq k \leq n\}, F, \Delta')$ with $F = Q_0$:

$$\delta(q, a) = \bigvee_{i=1}^{n} \bigwedge_{j=1}^{k_i} (q_{i,j}, i) \Rightarrow \forall i, f_i(q_{i,1}, ..., q_{i,k_i}) \longrightarrow q \in \delta'$$

2. Given $\mathcal{A} = (Q, \mathcal{F}, Q_f, \delta)$ an NFTA, we construct the AWA $\mathcal{A}' = (Q \times \mathcal{F}, \{a\}, I, F, \delta')$ with :

$$F = \{(q, f) \mid f \longrightarrow q \in \Delta\}$$

$$I = \{(q, f) \mid q \in Q_f\}$$

$$\delta((q, f), a) = \bigvee_{f(q_1, \ldots, q_n) \longrightarrow q \in \delta} \bigwedge_{i=1}^{n} \bigvee_{f_j \in \mathcal{F})} ((q_i, f_j), i)$$

.

We deduce that emptiness for AWA on singleton alphabet is P-hard.

Homework for next week : Membership

1. Recall the complexity of the uniform membership problem for DFTAs, NFTAs and NFHAs.
2. Prove that **(HarderUMembership)** :
Instance : an NFHA $\mathcal{A}$ where the horizontal languages are given by AWA (and not finite automata) and a word w
Question : $w \in L(\mathcal{A})$?
is in NP.