Tree Automata and their Applications

TD n°4 : Extensions and PDL

2021-2022

From last TD : Alternating Word Automata

Definition 1 *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 $\phi ::= \bot \mid \top \mid \phi \vee \phi \mid \phi \wedge \phi$.*

Definition 2 *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), $Q_0 \subset Q$ (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 such that :*
— *if $w = \varepsilon$, then $t = q_0$ with $q_0 \in Q_0$.*
— *if $w = a.w'$, then $t = q_0(t_1, \ldots, t_n)$ $q_0 \in Q_0$ and such that for all i, t_i is a run of w' on $(Q, \Sigma, q_i, Q_f, \delta)$ and $\{q_1, \ldots, q_n\} \models \delta(q_0, a)$.*

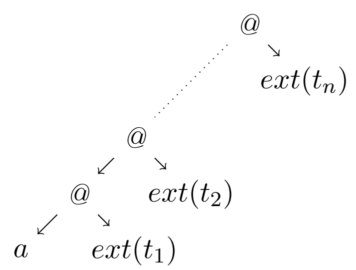
Definition 3 *We say that a run is accepting if every leaf of the form q satisfies that $q \in Q_f$.*

1. Show how to reduce the emptiness problem for an AWA on a one letter alphabet $\{a\}$ with formulas 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.

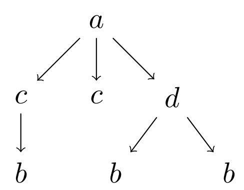
From last TD : Extensions

Definition 4 (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 :



1. Give the extension encoding of :



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

Exercise 3 : Warm up PDL

Definition 5 (PDL)
The syntax is the following :

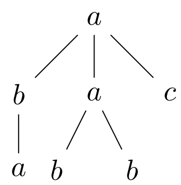
$$\phi ::= a \mid \top \mid \neg\phi \mid \phi \vee \phi \mid \langle\pi\rangle\,\phi \qquad \textit{(position formulae)}$$
$$\pi ::= \downarrow \mid \rightarrow \mid \pi^{-1} \mid \pi;\pi \mid \pi+\pi \mid \pi^{*} \mid \phi? \qquad \textit{(path formulae)}$$

The semantic is defined this way : let t be a tree, we define $[\![\phi]\!]_t$ (resp. $[\![\pi]\!]_t$) as a set of positions of t (resp. a relation on positions of t) by induction on the size of ϕ (resp. π) :

$$[\![a]\!]_t = \{w \in Pos(t) \mid t(w) = a\} \qquad [\![\downarrow]\!]_t = \{(w, w.i) \mid w, w.i \in Pos(t)\}$$
$$[\![\top]\!]_t = Pos(t) \qquad [\![\rightarrow]\!]_t = \{(w.i, w.(i+1)) \mid w.i, w.(i+1) \in Pos(t)\}$$
$$[\![\neg\phi]\!]_t = Pos(t) \setminus [\![\phi]\!]_t \qquad [\![\pi^{-1}]\!]_t = [\![\pi]\!]_t^{-1}$$
$$[\![\phi_1 \vee \phi_2]\!]_t = [\![\phi_1]\!]_t \cup [\![\phi_2]\!]_t \qquad [\![\pi_1;\pi_2]\!]_t = [\![\pi_2]\!]_t \circ [\![\pi_1]\!]_t$$
$$[\![\langle\pi\rangle\,\phi]\!]_t = [\![\pi]\!]_t^{-1}([\![\phi]\!]_t) \qquad [\![\pi_1+\pi_2]\!]_t = [\![\pi_1]\!]_t \cup [\![\pi_2]\!]_t$$
$$[\![\pi^{*}]\!]_t = [\![\pi]\!]_t^{*} \qquad [\![\phi?]\!]_t = \Delta_{[\![\phi]\!]_t} = \{(w,w) \mid w \in [\![\phi]\!]_t\}$$

Let t be a tree and w, $w' \in Pos(t)$. We note :
— *$t, w \models \phi$ if $w \in [\![\phi]\!]_t$*
— *$t \models \phi$ if $t, \epsilon \models \phi$ and we say that t satisfies ϕ*
— *$t, w, w' \models \pi$ if $(w, w') \in [\![\pi]\!]_t$*

Let t be the tree :



Which formulae are satisfied by t ?

1. $\phi_1 = \neg a \vee \langle\downarrow\rangle \left(\neg\langle\leftarrow\rangle\top \wedge b \wedge \langle\rightarrow^{*}\rangle (c \wedge \neg\langle\rightarrow\rangle\top)\right)$
2. $\phi_2 = \neg a \vee \langle\downarrow\rangle \left(\neg\langle\leftarrow\rangle\top \wedge b \wedge \langle(\rightarrow; c?)^{*}\rangle (\neg\langle\rightarrow\rangle\top)\right)$
3. $\phi_3 = \langle(a?;\downarrow)^{*}\rangle (a \wedge \neg\langle\downarrow\rangle\top)$

Exercise 4 : The power of PDL ?

Give a translation of PDL in MSO which preserves models. That is, given a position formula ϕ (resp. a path formula π), construct a MSO formula $\tilde{\phi}$ (resp. $\tilde{\pi}$) whose set of free variable is $\{X_a \mid a \in \mathcal{F}\} \cup \{x\}$ (resp. $\{X_a \mid a \in \mathcal{F}\} \cup \{x, y\}$) such that $t, w \models \phi$ iff $(P_a(t))_{a\in\mathcal{F}}, w \models \tilde{\phi}$ (resp. $t, w, w' \models \pi$ iff $(P_a(t))_{a\in\mathcal{F}}, w, w' \models \tilde{\pi}$) where $P_a(t) = \{w \in Pos(t) \mid t(w) = a\}$.

Homework for final week : From formulaes to automaton

Give tree automatons recognizing the languages on trees of maximum arity 2 defined by the formulae. (**To remind you, $P_f(z)$ means "at position z there is an f"**) (*you do not have to prove that your construction is correct, just the automata will suffice*) :

1. $(x \in S \wedge (x \downarrow_1 y \Rightarrow y \in S)) \wedge (z \in S \Rightarrow P_f(z))$
2. $\exists S.(x \in S \wedge (x \downarrow_1 y \Rightarrow y \in S)) \wedge (z \in S \Rightarrow P_f(z))$

Note : You can send the homework by mail to asuresh@lsv.fr by 13th January, 2022. You can send me pictures of your handwritten answers, if you are not comfortable with typing it up.