Automates d'arbre

TD n°3 : Trees and Logic

Exercise 1 : But first, a bit of homomorphism, with nutts.

A bottom-up tree transducer (NUTT) is a tuple $U = (Q, \mathcal{F}, \mathcal{F}', Q_f, \Delta)$ where Q is a finite set (of states), $\mathcal{F}$ and $\mathcal{F}'$ are finite ranked sets (of input and output), $Q_f \subseteq Q$ (final states) and Δ is a finite set of rules of the form :

- $f(q_1(x_1), ..., q_n(x_n)) \to q(u)$ where $f \in \mathcal{F}$ and $u \in T(\mathcal{F}', \{x_1, ..., x_n\})$
- $q(x_1) \to q'(u)$ where $u \in T(\mathcal{F}', \{x_1\})$.

We say that U is linear when the right side of the rules of Δ are. This defines a rewrite system $\to_U$ on $T(\mathcal{F} \cup \mathcal{F}' \cup Q)$. The relation induced by U is then $\mathcal{R}(U) = \{(t, t') \mid t \in T(\mathcal{F}),\ t' \in T(\mathcal{F}'),\ t \to_U^* q(t'),\ q \in Q_f\}$.

1) Prove that tree morphisms are a special case of NUTT that is if $\mu : T(\mathcal{F}) \longrightarrow T(\mathcal{F}')$ is a morphism, then there exists a NUTT U_μ such that $\mathcal{R}(U_\mu) = \{(t, \mu(t)) \mid t \in T(\mathcal{F})\}$. Be sure that if μ is linear then U_μ is too.
2) Prove that the domain of a NUTT U, that is $\{t \in T(\mathcal{F}) \mid \exists t' \in T(\mathcal{F}'),\ (t, t') \in U\}$, is recognizable.
3) Prove that the image of a recognizable tree language L by a linear NUTT U, that is $\{t' \in T(\mathcal{F}') \mid \exists t \in L,\ (t, t') \in U\}$, is recognizable.

Solution:

1) $Q = \{q\}$, $Q_f = \{Q\}$ and $\Delta =$
   - ⋆ $f(q(x_1), ..., q(x_n)) \longrightarrow q(\mu(f)(x_1, ..., x_n))$ linear when μ is
2) $Q = Q_U$, $F = F_U$ and $\Delta =$
   - ⋆ $f(q_1, ..., q_n) \longrightarrow q$ if there exists u such that $f(q_1(x_1), ..., q_n(x_n)) \longrightarrow q(u) \in \Delta_U$
   - ⋆ $q \longrightarrow q'$ if there exists u such that $q(x_1) \longrightarrow q'(u) \in \Delta_U$
3) Let U a NUTT and $\mathcal{A}$ a NFTA on $\mathcal{F}$. For every pair of rules $r = f(q_1(x_1), ..., q_n(x_n)) \longrightarrow q(u) \in \Delta_U$ and $r' = f(q'_1, ..., q'_n) \longrightarrow q' \in \Delta_{\mathcal{A}}$, we define :
   — $Q^{r,r'} = \{q_p^{r,r'} \mid p \in Pos(u)\}$
   — $\Delta^{r,r'} =$
   - ⋆ $g(q_{p.1}^{r,r'}, ..., q_{p.k}^{r,r'}) \longrightarrow q_p^{r,r'}$ for $p \in Pos(u)$ such that $u(p) = g \in \mathcal{F}'$
   - ⋆ $(q_i, q'_i) \longrightarrow q_p^{r,r'}$ if $u(p) = x_i$ (linearity assure that we only have one of this kind for every i)
   - ⋆ $q_\epsilon^{r,r'} \longrightarrow (q, q')$

   For every rule $r = q(x) \longrightarrow q'(u) \in \Delta_U$, we define :
   — $Q^r = \{q_p^r \mid p \in Pos(u)\} \times Q_{\mathcal{A}}$
   — $\Delta^r =$
   - ⋆ $g((q_{p.1}^r, q''), ..., (q_{p.k}^r, q'')) \longrightarrow (q_p^r, q'')$ for $p \in Pos(u)$ such that $u(p) = g \in \mathcal{F}'$ and $q'' \in Q_{\mathcal{A}}$
   - ⋆ $(q, q'') \longrightarrow (q_p^r, q'')$ if $u(p) = x$ and $q'' \in Q_{\mathcal{A}}$ (linearity assure that we only have one of this kind)
   - ⋆ $(q_\epsilon^r, q'') \longrightarrow (q, q'')$

   Then this NFTA works :

$$\tilde{Q} = Q_U \times Q_{\mathcal{A}} \cup \bigcup_{(r,r')} Q^{r,r'} \cup \bigcup_r Q^r$$

$$\tilde{F} = F_U \times F_{\mathcal{A}}$$

$$\tilde{\Delta} = \bigcup_{(r,r')} \Delta^{r,r'} \cup \bigcup_{r} \Delta^{r}$$

Exercise 2 : And a bit of minimization

Definition 1 *An equivalence relation $\equiv$ on T is a congruence on $T(\mathcal{F})$ if for every $f \in \mathcal{F}_n$:*

$$(\forall i, 1 \leq i \leq n, u_i \equiv v_i) \Rightarrow f(u_1, ..., u_n) \equiv f(v_1, ..., v_n)$$

For a given tree language L, let us define the congruence $\equiv_L$ on $T(\mathcal{F})$ by : $u \equiv_L v$ if for all contexts $C \in C(\mathcal{F})$:

$$C[u] \in L \Leftrightarrow C[v] \in L$$

Prove that the following are equivalent :

1. L is a recognizable tree language
2. L is the union of some equivalence classes of a congruence of finite index
3. the relation $\equiv_L$ is a congruence of finite index.

   Then, show how to obtain the minimal automaton of a language.

Solution:

- (1) $\Rightarrow$ (2). Assume that L is recognized by some complete DFTA $A = (Q, F, Q_f, \delta)$. We consider δ as a transition function. Let us consider the relation $\equiv_A$ defined on $T(\mathcal{F})$ by : $u \equiv_A v$ if $\delta(u) = \delta(v)$. Clearly $\equiv_A$ is a congruence relation and it is of finite index, since the number of equivalence classes is at most the number of states in Q. Furthermore, L is the union of those equivalence classes that include a term u such that $\delta(u)$ is a final state.
- (2) $\Rightarrow$ (3). Let us denote by $\sim$ the congruence of finite index, we assume that $u \sim v$. We can show by induction that $\forall C \in C(\mathcal{F}), C(u) \sim C(v)$. As L is the union of some equivalence classes of $\sim$, we have that $C(u) \in L \Leftrightarrow C(v) \in L$. Finally, we have that $u \equiv_L v$, and the equivalence class of u in $\sim$ is contained inside the one in $\equiv_L$. Consequently, the index of $\equiv_L$ is lower than $\sim$, which is finite.
- (3) $\Rightarrow$ (1) Let Q_{min} be the finite set of equivalence classes of L, we write $[u]$ for the equivalence class of u. Then, we define δ_{min} with :

  $$\delta_{min}(f, [u_1], ..., [u_n]) = [f(u_1), ..., f(u_n)]$$

  . Finally, we let $Q_{minf} = \{[u] | u \in L\}$. The DFTA $A_{min} = (Q_{min}, \mathcal{F},_{minf}, \delta_{min})$ recognizes the language L.

We thus constructed A_{min} which recognizes L. If we consider any automaton A recognizing L, we have with the first proof the relation $\equiv_A$ which has as many classes as the number of states of A. And with the second proof, we have that $\equiv_A$ as more classes than $\equiv_L$. So $\equiv_L$ as less classes than the number of states of A. And finally with the third proof, we have that the number of classes of $\equiv_L$ is the number of states of A_{min}. In conclusion, any automaton A recognizing L has more states (or equal) than A_{min}. Thus, A_{min} can indeed be called the minimal automaton.

Exercise 3 : MSO on finite trees

We consider trees with maximum arity 2. Give MSO formulae which express the following :

1. X is closed under predecessors
2. $x \subseteq y$ (with $\subseteq$ the prefix relation on positions)
3. 'a' occurs twice on the same path
4. 'a' occurs twice not on the same path
5. There exists a sub tree with only a's
6. The frontier word contains the chain 'ab'

Solution:

1. $closed(X) := \forall y \forall z(y \in X \wedge (z \downarrow_1 y) \vee z \downarrow_2 y)) \Rightarrow z \in X)$
2. $x \subseteq y := \forall X(y \in X \wedge closed(X) \Rightarrow X(x))$
3. $\exists x \exists y(\neg(x = y) \wedge x \subseteq y \wedge P_a(x) \wedge P_a(y))$
4. $\exists x \exists y(\neg(y \subseteq x) \wedge \neg(x \subseteq y) \wedge Pa(x) \wedge P_a(y))$
5. $\exists x \forall y(x \subseteq y \Rightarrow P_a(y))$
6. We first implement a way to say that a leaf is next to another one :

$$x \prec y := \exists x_0 \exists y_0 \exists z(z \downarrow_1 x_0) \wedge (z \downarrow_2 y_0) \wedge x_0 \subseteq x \wedge y_0 \subseteq y)$$

   And with this :

$$\exists x \exists y(Fr(x) \wedge Fr(y) \wedge P_a(x) \wedge P_b(y) \wedge x \prec y \wedge \neg \exists z(Fr(z) \wedge x \prec z \wedge z \prec y))$$

Exercise 4 : From formulaes to automaton

Give tree automatons recognizing the languages on trees of maximum arity 2 defined by the formulae :

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

Solution:

1. We construct an NFTA $\mathcal{A}_1$ on $\Sigma \times \{0,1\}^2$, which recognizes $x \in S$. The idea is to reject if we can witness a $x \notin S$, and we accept otherwise. So, for all $f \in \mathcal{F}$:
   — $(f, 1, 0)(q_1, q_2) \longrightarrow \perp \;\; if\ \forall i, q_i \neq \perp$
   — $(f, \_, \_)(q_1, q_2) \longrightarrow \top\ if\ \forall i, q_i \neq \perp$
   We construct an NFTA $\mathcal{A}_2$ on $\Sigma \times \{0,1\}^3$, which recognizes $(x \downarrow_1 y \Rightarrow y \in S))$. If we witness a $y \notin S$, we go into a specific state to check if it is not the son of x, thus failing the formula.
   — $(f,, 1, 0)(q_1, q_2) \longrightarrow q_{y \notin S}\ if\ \forall i, q_i \neq \perp$
   — $(f,, 1, 0)(q_{y \notin S}, q_2) \longrightarrow \perp$
   — $(f, \_, \_)(q_1, q_2) \longrightarrow \top\ if\ \forall i, q_i \neq \perp$
   We construct an NFTA $\mathcal{A}_3$ on $\Sigma \times \{0,1\}^2$, which recognizes $(z \in S \Rightarrow P_h(z))$.
   — $(f, 1, 1)(q_1, q_2) \longrightarrow \perp \;\; if\ \forall i, q_i \neq \perp, \forall f \neq h \in \mathcal{F}$
   — $(f, \_, \_)(q_1, q_2) \longrightarrow \top\ if\ \forall i, q_i \neq \perp, \forall f \in \mathcal{F}$
   Then, with the correct inversed projections, we can transform A_i into A'_i on $\Sigma \times \{0,1\}^4$ with ordering (x, y, z, S), and $\bigcap A'_i$ is the desired automaton.
2. We project $\bigcap A'_i$ on $\Sigma \times \{0,1\}^3$, and we obtain the result.