Automates d'arbre

TD n°3 : Minimization and Logic

Exercise 1 : Minimization results

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 2 : Let's try to minimize

We consider the complete DFTA on $\mathcal{F} = \{f/2, g/2, a/0, b/0\}$ with states $\{q_a, q_b, q_f, q_g, \top, \bot\}$, finale state $\top$ and transitions :

— $a \longrightarrow q_a$
— $b \longrightarrow q_b$
— $f(q_a, q_b) \longrightarrow q_f$
— $f(q_f, q_b) \longrightarrow \top$
— $g(q_a, q_a) \longrightarrow q_q$
— $f(q_g, q_b) \longrightarrow \top$
— $h(q, q') \longrightarrow \top$ if $h \in \{f, g\}$, and $q = \top$ or $q' = \top$.
— $h(q, q') \longrightarrow \perp$ in all other cases.

Give the corresponding minimized algorithm obtained through the partition refinement algorithm.

Solution:
The initial partitioning is $P = \{(\top), (q_a, q_b, q_g, q_f, \perp)\}$.
Then, we can for instance distinguish q_a with q_f as $f(q_f, q_b) \longrightarrow \top$ but $Tf(q_a, q_b) \longrightarrow \perp$ and we do not have $(\perp\ P\ \top)$. This argument is also valid for (q_b, q_f),(q_b, q_g), (q_a, q_b),(q_a, q_g), and $(\perp,_))$. However, we have that $f(q_f, q_b) \longrightarrow \top$ and $Tf(q_g, q_b) \longrightarrow \top$, so we have $(q_f P' q_g)$. Finally, at the end of the first loop, $P' = \{(\top), (q_b), (q_a), (\perp), (q_f, q_g)\}$. If we try once more, it is stable, so we have the minimal automaton by merging q_f and q_g.

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

Solution:

1. $\mathit{closed}(X) := \forall y \forall z (y \in X \wedge (z \downarrow_1 y) \vee z \downarrow_2 y)) \Rightarrow z \in X)$
2. $x \subseteq y := \forall X (y \in X \wedge \mathit{closed}(X) \Rightarrow X(x))$
3. $\exists x \exists y (\neg(x = y) \wedge x \subseteq y \wedge P_a(x) \wedge P_a(y))$