

TD 2 : Minimisation

Exercice 1 (Calcul de l'équivalence de NERODE).

Définition 1 (Congruence). Soit $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe complet, et $\sim$ une relation d'équivalence sur Q . Cette relation est une *congruence* si elle est compatible avec les transitions de $\mathcal{A}$:

$$q \sim q' \text{ implique } \forall a \in \Sigma, \delta(q, a) \sim \delta(q', a) \quad (1)$$

et si elle sature $\mathcal{A}$:

$$q \sim q' \text{ implique } q \in F \text{ ssi } q' \in F \quad (2)$$

pour tous états q, q' de Q .

Notons $[q]$ pour la classe d'équivalence de l'état q .

Définition 2 (Quotient). Soit $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe complet et $\sim$ une équivalence sur Q . L'*automate quotient* de $\mathcal{A}$ par $\sim$ est l'automate $\mathcal{A}/\sim = \langle \Sigma, Q/\sim, [q_0], \{[q_f] \mid q_f \in F\}, \delta/\sim \rangle$ avec

$$\delta/\sim = \{([q], a, [\delta(q, a)]) \mid q \in Q, a \in \Sigma\}.$$

Définition 3 (Congruence de NERODE). Soit $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe complet. On appelle la relation d'équivalence $\cong$ suivante la *congruence de NERODE* :

$$q \cong q' \text{ ssi } L(\langle Q, \Sigma, \delta, q, F \rangle) = L(\langle Q, \Sigma, \delta, q', F \rangle)$$

pour tous états q, q' de Q .

Définition 4 (Algorithme de MOORE). Soit $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe complet. On définit $\cong_i$ sur Q par

$$q \cong_0 q' \text{ ssi } (q \in F \text{ ssi } q' \in F) \quad (3)$$

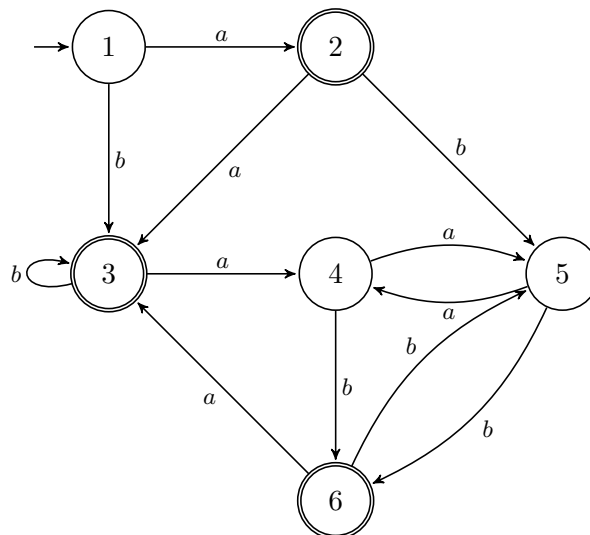
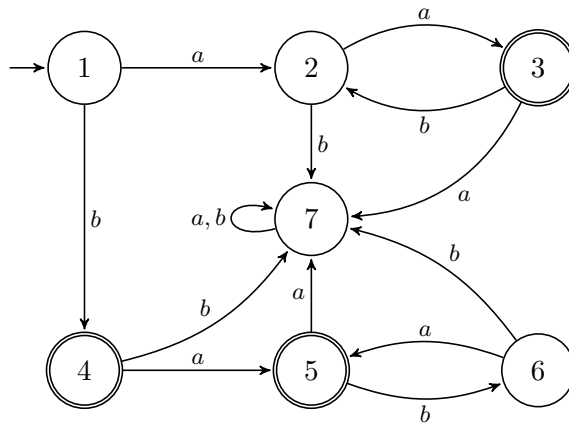
$$q \cong_{i+1} q' \text{ ssi } q \cong_i q' \text{ et } \forall a \in \Sigma, \delta(q, a) \cong_i \delta(q', a) \quad (4)$$

pour tous états q, q' de Q et pour tout $i \geq 0$.

Le quotient d'un automate déterministe complet $\mathcal{A}$ par la congruence de NERODE est l'automate déterministe complet minimal équivalent à $\mathcal{A}$.

1. Soit $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe complet et $\sim$ une congruence sur Q . Montrez que l'automate quotient $\mathcal{A}/\sim$ est déterministe complet et reconnaît $L(\mathcal{A})$.
2. Montrez que la congruence de NERODE est bien une congruence au sens de la définition 1.
3. Montrez qu'il existe k tel que $\cong_k = \cong_{k+j}$ pour tout $j \geq 0$, et tel que $\cong_k = \cong$.

Exercice 2 (Minimisations). Minimiser les automates suivants :



Exercice 3 (Minimisation par renversement). Montrer que l'automate déterminisé d'un automate co-déterministe et co-accessible est minimal. Quelle est la complexité de cette méthode de minimisation ?

Exercice 4 (Algorithme d'HOPCROFT).

1. On considère l'algorithme de raffinement de partition suivant :

Algorithme 1.

PARTITIONNE ($A = \langle Q, \Sigma, \delta, i, F \rangle$)

1. $Q/\sim \leftarrow \{F, Q \setminus F\}$
2. **tant que** $\exists B, C \in Q/\sim, a \in \Sigma. (\delta(B, a) \cap C \neq \emptyset \wedge \delta(B, a) \not\subseteq C)$ **faire**
3. RAFFINE ($Q/\sim, B, C, a$)
4. **fin tant que**
5. **retourner** $Q/\sim$

RAFFINE ($Q/\sim, B, C, a$)

1. $B_1 \leftarrow B \cap \delta^{-1}(C, a)$
2. $B_2 \leftarrow B \setminus B_1$
3. $Q/\sim \leftarrow (Q/\sim \setminus \{B\}) \cup \{B_1, B_2\}$
4. **retourner** $\{B_1, B_2\}$

Dans cet algorithme, l'ensemble C sert à raffiner le bloc B par rapport à la lettre a .

- (a) Montrer que la procédure de l'algorithme 1 termine après au plus $|Q|$ itérations.
- (b) Montrer que l'algorithme 1 calcule la congruence de NERODE sur Q .
- (c) Donner une famille d'automates avec n états telle que cet algorithme effectue $\Omega(n^2)$ opérations.
2. Une première étape en direction de l'algorithme d'HOPCROFT consiste à itérer sur un ensemble L de *raffineurs potentiels* (C, a) dans $2^Q \times \Sigma$.
 - (a) Soit (C, a) un raffineur potentiel. Comment calculer l'ensemble des blocs B qui seraient *effectivement* raffinés par (C, a) ? On note $R_\sim(C, a)$ cet ensemble. Montrer qu'il peut être calculé en temps $O(\max(|C|, |\delta^{-1}(C, a)|))$.
 - (b) Supposons que B soit raffiné par (C, a) en B_1 et B_2 , et que D soit un sous-ensemble de B_1 ou de B_2 . Le bloc D peut-il être effectivement raffiné par (C, a) ?
 - (c) Soit B un bloc raffiné en $B_1 \uplus B_2$. Montrer que, pour tout a de Σ , raffiner par rapport à (B, a) , (B_1, a) et (B_2, a) revient au même que raffiner par rapport à (B_1, a) et (B_2, a) .
 - (d) En déduire une reformulation simple de la procédure PARTITIONNE qui itère sur un ensemble de raffineurs potentiels (et met cet ensemble à jour au fur et à mesure).
3. Pour obtenir une complexité en $O(n \log n)$, l'algorithme d'HOPCROFT évite l'ajout de certains raffineurs redondants dans L .
 - (a) Soit B un bloc raffiné en $B_1 \uplus B_2$. Montrer que, pour tout a de Σ , raffiner par rapport à (B, a) et (B_1, a) revient au même que raffiner par rapport à (B, a) et (B_2, a) .
 - (b) Donner une nouvelle version de l'algorithme qui travaille en $O(n \log n)$.