Mots synchronisants (en C)

30 septembre 2024

On appelle **machine** un triplet (Q, Σ, δ) où Q est un ensemble fini non vide d'**états**, Σ un alphabet et δ une application de $Q \times \Sigma$ dans Q appelée **fonction de transition**. Une machine peut donc être vue comme un automate fini déterministe complet sans notion d'état initial ou final. Comme pour les automates finis, on utilisera la notion de fonction de transition étendue définie par :

— pour tout $q \in Q$, $\delta^*(q, \varepsilon) = q$;

— pour tous $q \in Q$, $u \in \Sigma^*$ et $a \in \Sigma$, $\delta^*(q, ua) = \delta(\delta^*(q, u), a)$.

Un mot $u \in \Sigma^*$ est dit **synchronisant** pour une machine (Q, σ, δ) s'il existe $q_0 \in Q$ tel que $\forall q \in Q$, $\delta^*(q, u) = q_0$. L'existence de tels mots permet de ramener une machine dans un état particulier connu en lisant un mot donné, donc en pratique de réinitialiser une machine réelle. La figure 1 représente une machine M_0. On pourra remarquer que ba et bb sont des mots synchronisants pour M_0.

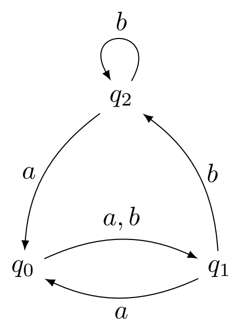

FIGURE 1 – La machine M_0.

On représente une lettre de Σ en C par un objet de type `char`. On supposera que Σ est constitué de lettres minuscules consécutives de l'alphabet courant commençant par la lettre `'a'`. Ainsi, si `a` est un objet de type `char`, alors `a - 97` est un entier compris entre 0 et 25. Un mot de Σ^* sera alors représenté par une chaîne de caractères de type `char*`. On rappelle que le caractère de fin de chaîne est `'\0'`.

Une machine sera représentée par un objet de type `machine` défini par :

```
struct Machine{
    int tQ;
    int tSigma;
    int** delta;
};

typedef struct Machine machine;
```

tel que si $M = (Q, \Sigma, \delta)$ est une machine représenté par un objet `M` de type `machine*`, alors :

— `M->tQ` représente $|Q|$, on suppose $Q = [\![0, |Q| - 1]\!]$;

— `M->tSigma` représente $|\Sigma|$, on suppose $\Sigma = [\![0, |\Sigma| - 1]\!]$;

— `M->delta` correspond à un tableau des transitions, c'est-à-dire tel que si $q \in Q$ et $a \in \Sigma$, alors `M->delta[q][a]` vaut $\delta(q, a)$.

1. Écrire une fonction `machine* init_machine(int tQ, int tSigma)` qui crée une machine telle que $|Q|$ et $|\Sigma|$ sont donnés en arguments et, pour tout $q \in Q$ et $a \in \Sigma$, $\delta(q, a) = q$.
2. Écrire une fonction `void liberer_machine(machine* M)` qui libère l'espace mémoire occupé par une machine.

3. Que dire de l'ensemble des mots synchronisants pour une machine ayant un seul état ?

   Dans toute la suite du problème, on supposera que les machines ont au moins deux états.

4. On considère la machine M_1 représentée figure 2, sur l'alphabet $\Sigma = \{a\}$. Donner un mot synchronisant pour M_1 s'il en existe un. Justifier la réponse.

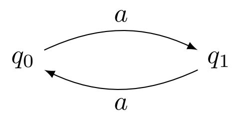

FIGURE 2 – La machine M_1.

5. Donner un mot synchronisant de trois lettres pour la machine M_2 représentée figure 3. On ne demande pas de justifier la réponse.

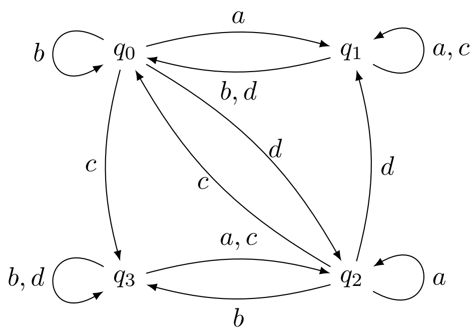

FIGURE 3 – La machine M_2.

6. Écrire une fonction `int delta_etoile(machine* M, int q, char* u)` qui prend en arguments une machine $M = (Q, \Sigma, \delta)$, un état q et un mot u et renvoie $\delta^*(q, u)$. On prendra garde que `u` est une chaîne de caractères, et qu'un caractère a une valeur comprise entre 97 et 122 et non entre 0 et 25.
7. Écrire une fonction `bool synchronisant(machine* M, char* u)` qui prend en argument une machine M et un mot u et renvoie le booléen `true` si u est synchronisant pour M et `false` sinon.
8. Montrer que si une machine admet un mot synchronisant alors il existe $a \in \Sigma$ et $q \neq q' \in Q$ tels que $\delta(q, a) = \delta(q', a)$.

   Soit $LS(M)$ le langage des mots synchronisants d'une machine $M = (Q, \Sigma, \delta)$. On introduit la machine des parties $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta})$ où $\widehat{Q} = \mathcal{P}(Q)$ et $\widehat{\delta}$ est définie par :

$$\forall P \subset Q, \forall a \in \Sigma, \widehat{\delta}(P, a) = \{\delta(p, a), p \in P\}$$

9. Montrer que l'existence d'un mot synchronisant pour M se ramène à un problème d'accessibilité de certain(s) état(s) depuis certain(s) état(s) de $\widehat{M}$.
10. En déduire que le langage $LS(M)$ des mots synchronisants de M est reconnaissable.
11. Déterminer puis représenter graphiquement un automate fini déterministe (pas nécessairement complet) reconnaissant $LS(M_0)$.
12. Montrer que si l'on sait résoudre le problème de l'existence d'un mot synchronisant, on sait dire, pour une machine M et un état q_0 de M choisi, s'il existe un mot u tel que pour tout état q de Q, le chemin menant de q à $\delta^*(q, u)$ passe forcément par q_0.