

# Devoir maison n°5

À rendre lundi 02/12

\*\*\*

## Mots synchronisants

Le sujet traite de synchronisation dans des automates, ainsi qu'un jeu à deux joueurs sur la synchronisation.

Les questions de programmation doivent être traitées en langage C uniquement. On supposera que les bibliothèques `stdlib.h`, `stdbool.h` et `string.h` ont été chargées.

### 1 Machines synchronisées

On appelle **machine** un triplet  $(Q, \Sigma, \delta)$  où  $Q$  est un ensemble fini non vide d'états,  $\Sigma$  un alphabet et  $\delta$  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  $q \in Q$ ,  $\delta^*(q, \epsilon) = q$ ;
- 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  $(Q, \sigma, \delta)$  s'il existe  $q_f \in Q$  tel que  $\forall q \in Q$ ,  $\delta^*(q, u) = q_f$ . 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 machine  $M$  est dite **synchronisée** si elle possède un mot synchronisant.

La figure 1 représente une machine  $M_0$ . On pourra remarquer que  $ba$  et  $bb$  sont des mots synchronisants pour  $M_0$ .

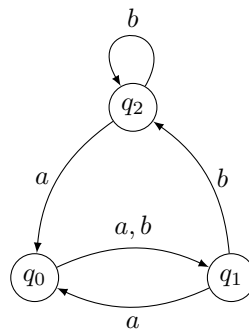


FIGURE 1 – La machine  $M_0$ .

On représente une lettre de  $\Sigma$  en C par un objet de type `char`. Un mot de  $\Sigma^*$  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 Q;
    int Sigma;
    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.Q` représente  $|Q|$ , on suppose  $Q = \llbracket 0, |Q| - 1 \rrbracket$  ;
- `M.Sigma` représente  $|\Sigma|$  ;
- `M.delta` est un tableau de  $|Q|$  tableaux tels que si  $q \in Q$  et  $a \in \Sigma$ , alors `M.delta[q][code(a)]` vaut  $\delta(q, a)$ , où `int code(char a)` est une bijection de  $\Sigma$  dans  $\llbracket 0, |\Sigma| - 1 \rrbracket$ , qu'on suppose déjà définie et qu'on ne demande pas d'explicitier.

**Question 1** Que dire de l'ensemble des mots synchronisant pour une machine ayant un seul état ?

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

**Question 2** On considère la machine  $M_1$  représentée figure 2, sur l'alphabet  $\Sigma = \{a\}$ . Donner, en justifiant, un mot synchronisant pour  $M_1$  s'il en existe un, ou prouver qu'il n'en existe pas.

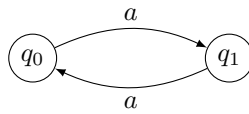


FIGURE 2 – La machine  $M_1$ .

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

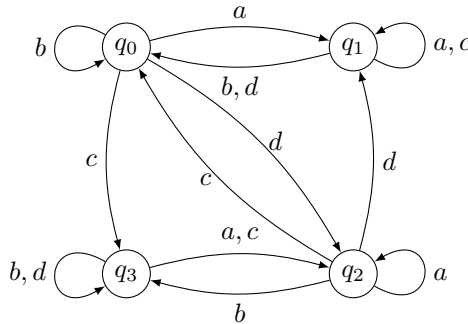


FIGURE 3 – La machine  $M_2$ .

**Question 4** 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  $\delta^*(q, u)$ .

**Question 5** 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.

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 :

$$\forall X \subseteq Q, \forall a \in \Sigma, \widehat{\delta}(X, a) = \{\delta(q, a), q \in X\}$$

**Question 6** Montrer que l'existence d'un mot synchronisant pour  $M$  se ramène à un problème d'accessibilité à certain(s) état(s) depuis certain(s) état(s) de  $\widehat{M}$ .

**Question 7** En déduire que le langage  $LS(M)$  des mots synchronisants de  $M$  est reconnaissable.

**Question 8** Déterminer puis représenter graphiquement un automate fini déterministe (pas nécessairement complet) reconnaissant  $LS(M_0)$ .

## 2 Existence de mot synchronisant

On considère le problème de décision **SYNCHRONISÉE** :

\* **Instance** : une machine  $M = (Q, \Sigma, \delta)$ .

\* **Question** : la machine  $M$  est-elle synchronisée ?

La question 6 montre que ce problème est décidable, mais suggère la création de l'automate des parties, de taille exponentielle en le nombre d'états d'une machine  $M$ . Dans cette partie, on cherche à montrer qu'on peut résoudre ce problème en temps polynomial.

### 2.1 Machine des paires

On considère une machine  $M = (Q, \Sigma, \delta)$ . Pour  $X \subseteq Q$  et  $u \in \Sigma^*$ , on note  $\delta^*(X, u) = \{\delta^*(q, u), q \in X\}$ .

**Question 9** Soit  $u$  un mot synchronisant de  $M$  et  $u_0, u_1, \dots, u_r$  une suite de préfixes de  $u$ , rangés dans l'ordre croissant de leur longueur et telle que  $u_r = u$ . Que peut-on dire de la suite des cardinaux  $|\delta^*(Q, u_i)|$  ?

Soient  $p, q \in Q^2$ ,  $p \neq q$ . La paire d'états  $\{p, q\}$  est dite **synchronisée** s'il existe un mot  $u_{p,q} \in \Sigma^*$  tel que  $\delta^*(p, u_{p,q}) = \delta^*(q, u_{p,q})$ .

**Question 10** Montrer que  $M$  est synchronisée si et seulement si toute paire d'états de  $Q$  est synchronisée.

On veut se servir du critère établi ci-dessus pour déterminer s'il existe un mot synchronisant. Pour cela, on associe à la machine  $M$  la machine **des paires**  $\tilde{M} = (\tilde{Q}, \Sigma, \tilde{\delta})$  définie par :

- $\tilde{Q} = \mathcal{P}_1(Q) \cup \mathcal{P}_2(Q)$  est formé des parties à un ou deux éléments de  $Q$  ;
- $\tilde{\delta}$  est définie par  $\forall (X, a) \in \tilde{Q} \times \Sigma, \tilde{\delta}(X, a) = \{\delta(q, a), q \in X\}$ .

La machine  $\tilde{M}$  est donc une machine à  $\tilde{n} = \frac{n(n+1)}{2}$  états (où  $n = |Q|$ ). On suppose disposer d'une bijection  $\varphi : \tilde{Q} \rightarrow \llbracket 0, \tilde{n} - 1 \rrbracket$  sous la forme d'une fonction `int phi(int p, int q)` telle que pour  $p, q \in Q^2$ , `phi(p, q)` renvoie  $\varphi(\{p, q\})$  (y compris dans le cas où  $p = q$ ).

**Question 11** Écrire une fonction `bool paire_synchronisee(machine M, int p, int q)` qui prend en argument une machine  $M$  et deux états  $p$  et  $q$  dans la machine  $M$  et renvoie un booléen qui indique si la paire  $\{p, q\}$  est synchronisée.

*On ne demande pas nécessairement de construire la machine des paires explicitement, mais on pourra l'utiliser dans le raisonnement.*

**Question 12** En déduire une fonction `bool synchronisee(machine M)` qui détermine si une machine est synchronisée ou non.

**Question 13** Déterminer la complexité temporelle de la fonction précédente en fonction du nombre  $n$  d'états de la machine  $M$ .

**Question 14** Montrer qu'on peut résoudre le problème **SYNCHRONISÉE** en complexité temporelle  $\mathcal{O}(|Q|^2|\Sigma|)$ . On détaillera en français l'algorithme correspondant, mais on ne demande pas de l'implémenter.

## 2.2 Bornes sur les mots synchronisants

Dans cette partie, on considère une machine synchronisée  $M = (Q, \Sigma, \delta)$  avec  $n = |Q|$ . On s'intéresse à des bornes sur la taille d'un mot synchronisant de  $M$  en fonction de  $n$ .

**Question 15** Montrer que  $M$  possède un mot synchronisant de taille au plus  $\frac{n(n-1)^2}{2}$ .

*Indication : on pourra s'intéresser à la taille d'un mot synchronisant une paire d'états.*

Pour  $n \in \mathbb{N}^*$ , on définit la machine de Černý d'ordre  $n$ , notée  $\mathcal{C}_n$ , par  $\mathcal{C}_n = (Q = \llbracket 0, n-1 \rrbracket, \Sigma = \{a, b\}, \delta)$  où, pour  $q \in Q$  :

- $\delta(q, a) = \begin{cases} 1 & \text{si } q = 0 \\ q & \text{sinon} \end{cases}$  ;
- $\delta(q, b) = (q + 1) \bmod n$ .

La figure 4 représente la machine  $\mathcal{C}_3$ .

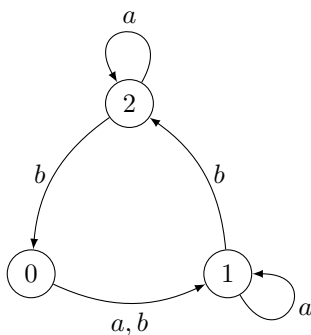


FIGURE 4 – La machine  $\mathcal{C}_3$ .

**Question 16** Montrer que pour  $n \in \mathbb{N}^*$ ,  $\mathcal{C}_n$  est synchronisée et donner l'expression d'un mot synchronisant de taille  $(n-1)^2$ .

Dans la machine  $\mathcal{C}_n$ , pour  $X \subseteq Q$ , on note  $\Delta(X)$  le plus grand « trou » de  $X$ , c'est-à-dire la plus grande suite d'états consécutifs (modulo  $n$ ) de  $Q \setminus X$ . Par exemple, pour  $n = 8$  et  $X = \{2, 4, 5\}$ ,  $\Delta(X) = \{6, 7, 0, 1\}$ . Si plusieurs ensembles  $\Delta(X)$  peuvent convenir, on en choisit un arbitrairement.

**Question 17** Soit  $X \subseteq Q$  et  $\alpha \in \Sigma$  tels que  $|\Delta(X)| = j$  et  $|\Delta(\delta(X, \alpha))| = j + 1$ . Quelles peuvent être les valeurs possibles de  $\alpha$  et  $\Delta(X)$  ?

**Question 18** En déduire qu'il n'existe pas, pour  $n \in \mathbb{N}^*$ , de mot synchronisant pour  $\mathcal{C}_n$  de taille strictement inférieure à  $(n-1)^2$ .

## 2.3 Difficulté du mot synchronisant minimal

On considère le problème de décision **SYNCHRONISANT MINIMAL** :

- \* **Instance** : une machine  $M = (Q, \Sigma, \delta)$  et un entier  $n \in \mathbb{N}$ .
- \* **Question** : existe-t-il un mot synchronisant de  $M$  de longueur  $\leq n$  ?

On cherche à montrer que ce problème est NP-complet par une réduction depuis **3SAT**, qu'on admet comme étant NP-complet. Pour cela, on considère une formule  $\varphi = \bigwedge_{i=1}^n C_i$  où les  $C_i$  sont des clauses disjonctives contenant au plus trois littéraux sur l'ensemble de variables  $\mathcal{V} = \{x_1, \dots, x_m\}$ .

On définit la machine  $M_\varphi = (Q, \Sigma, \delta)$  par :

- $Q = \{f\} \cup \{q_{i,j} \mid i \in \llbracket 1, n \rrbracket, j \in \llbracket 1, m+1 \rrbracket\}$  (où  $n$  est le nombre de clauses et  $m$  le nombre de variables) ;
- $\Sigma = \{0, 1\}$  ;
- $\delta$  est définie par :
  - \*  $\delta(f, 0) = \delta(f, 1) = f$  ;
  - \* pour  $i \in \llbracket 1, n \rrbracket$ ,  $\delta(q_{i,m+1}, 0) = \delta(q_{i,m+1}, 1) = f$  ;
  - \* pour  $i \in \llbracket 1, n \rrbracket$  et  $j \in \llbracket 1, m \rrbracket$  :

$$\delta(q_{i,j}, 1) = \begin{cases} f & \text{si le littéral } x_j \text{ apparaît dans la clause } C_i \\ q_{i,j+1} & \text{sinon} \end{cases}$$

$$\delta(q_{i,j}, 0) = \begin{cases} f & \text{si le littéral } \neg x_j \text{ apparaît dans la clause } C_i \\ q_{i,j+1} & \text{sinon} \end{cases}$$

**Question 19** Représenter la machine  $M_\varphi$  pour  $\varphi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_4}) \wedge (x_2 \vee \overline{x_3} \vee x_4)$ .

*Pour alléger, on pourra choisir de ne pas représenter toutes les transitions, en précisant comment ajouter celles qui manquent.*

**Question 20** Donner un modèle  $\mu \in \{0, 1\}^\mathcal{V}$  de  $\varphi$ . Le mot  $\mu(x_1)\mu(x_2)\mu(x_3)\mu(x_4)$  est-il synchronisant ?

**Question 21** Montrer que formule  $\varphi$  est satisfiable si et seulement si  $M_\varphi$  possède un mot synchronisant de longueur inférieure ou égale à  $m$ .

**Question 22** En déduire que SYNCHRONISANT MINIMAL est NP-complet.

### 3 Jeu de synchronisation

On considère le jeu à deux joueurs suivant dans une machine  $M = (Q, \Sigma, \delta)$ , appelé **jeu de synchronisation** :

- initialement, il y a une pièce sur chaque état de  $M$  ;
- les joueurs jouent tour à tour, en commençant par le joueur 1. Le tour d'un joueur se déroule comme suit :
  - \* le joueur choisit une lettre  $a$  ;
  - \* pour chaque état  $q$  qui contient une pièce, le joueur déplace la pièce de l'état  $q$  vers  $\delta(q, a)$  ;
  - \* si après les déplacements un état contient plusieurs pièces, le joueur les enlève toutes sauf une (de telle sorte qu'à la fin de son tour, un état contient au plus une pièce) ;
- la partie se termine s'il ne reste qu'un seul état qui contient une pièce, auquel cas c'est le joueur 1 qui remporte la partie. Si le joueur 2 peut empêcher la partie de se terminer indéfiniment, c'est lui qui remporte la partie.

Pour  $M$  une machine fixée, une **configuration** du jeu est un ensemble  $X \subseteq Q$  qui représente les états de  $M$  qui contiennent une pièce. Une **partie** peut alors être vue soit comme une suite de lettres  $(a_i)_{i \in \mathbb{N}^*}$  (correspondant aux lettres choisies par les joueurs), soit comme une suite de parties  $(X_i)_{i \in \mathbb{N}}$  (correspondant aux configurations de la partie), avec  $X_0 = Q$ . On remarque qu'on peut faire le lien entre ces deux représentations : pour  $i \in \mathbb{N}$ ,  $X_{i+1} = \delta(X_i, a_{i+1})$ . La partie est gagnée par le joueur 1 s'il existe  $i \in \mathbb{N}$  tel que  $|X_i| = 1$ .

Une **stratégie** pour un joueur est alors une fonction  $f : \mathcal{P}(Q) \rightarrow \Sigma$ . Si  $f$  et  $g$  sont des stratégies pour les joueurs 1 et 2 respectivement, on appelle  $(f, g)$ -**partie** depuis une configuration  $X \subseteq Q$  la partie  $(X_i)_{i \in \mathbb{N}}$  telle que  $X_0 = X$  et :

- pour  $i \in 2\mathbb{N}$ ,  $X_{i+1} = \delta(X_i, f(X_i))$  ;
- pour  $i \in 2\mathbb{N} + 1$ ,  $X_{i+1} = \delta(X_i, g(X_i))$ .

Une stratégie  $f$  est **gagnante** depuis une configuration  $X$  pour le joueur 1 si quelle que soit la stratégie  $g$  pour le joueur 2, la  $(f, g)$ -partie depuis  $X$  est gagnante pour 1. On définit de même une stratégie gagnante pour le joueur 2. Lorsqu'on parle de stratégie gagnante sans préciser la configuration  $X$ , il sera sous-entendu qu'il s'agit d'une stratégie gagnante depuis la configuration  $X = Q$ .

**Question 23** Montrer que s'il existe une stratégie gagnante pour 1 dans une machine  $M$ , alors  $M$  est synchronisée.

**Question 24** Montrer que pour une machine  $M$ , il existe une stratégie gagnante pour le joueur 1 si et seulement si il existe une stratégie gagnante pour le joueur 1 depuis toute configuration  $X$  de cardinal 2.

La réciproque à la question 24 est fausse, et on peut utiliser à nouveau les machines de Černý.

**Question 25** Montrer que pour  $n \geq 4$ , il existe une stratégie gagnante pour le joueur 2 dans le jeu de synchronisation dans la machine  $\mathcal{C}_n$ .

Soient  $p, q \in Q^2$ , on dit que  $q$  est **accessible** depuis  $p$ , noté  $p \preceq q$ , s'il existe  $u \in \Sigma^*$  tel que  $\delta^*(p, u) = q$ . Un état **puits** est un état maximal pour la relation  $\preceq$ , c'est-à-dire un état  $q_m \in Q$  tel que pour tout  $q \in Q \setminus \{q_m\}$ ,  $q_m \not\preceq q$ .

$M$  est dite **faiblement acyclique** si la relation  $\preceq$  est antisymétrique.

**Question 26** Soit  $M = (Q, \Sigma, \delta)$  une machine faiblement acyclique. Montrer qu'il y a équivalence entre les trois propriétés suivantes :

- (a)  $M$  a un unique état puits.
- (b)  $M$  est synchronisée.
- (c) Le joueur 1 a une stratégie gagnante dans  $M$ .

## Partie bonus

**Question 27** Soit  $M = (Q, \Sigma, \delta)$  une machine synchronisée à  $n$  états. Montrer que  $M$  admet un mot synchronisant de longueur  $(n-1)^2$ .

\*\*\*