Exercice 1 : Question de cours et preuve (6 pt)

On considère l'alphabet $\{0, 1, \square, \$\}$.

Q1. (1 pt) Expliquez la différence entre une MT M_D qui **décide** le langage L et une MT M_R qui **reconnaît** le langage L.

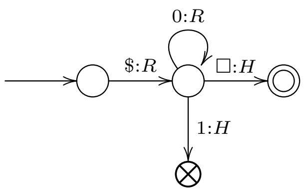
SOLUTION

— *La MT M_D termine pour tout mot de $\omega \in \{0,1\}^*$. Elle répond $\mathbb{V}$ si $\omega \in L$ et $\mathbb{F}$ si $\omega \notin L$. À partir de M_D on peut alors construire un machine qui reconnaît $\overline{L}$ en inversant ses états ◎ et ⊗.*

— *La MT M_R termine dans l'état ◎ pour tout mot $\omega \in L$. Pour certains mots $\omega \notin L$ elle peut ne pas s'arrêter. On ne peut donc pas utiliser M_R pour construire une MT qui reconnaît $\overline{L}$.*

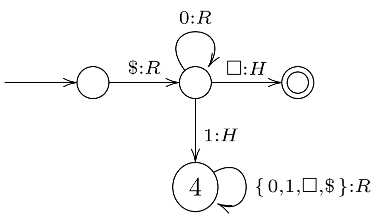
Q2. (0.5 pt) Donnez une MT qui **décide** le langage $\{0^n \mid n \in \mathbb{N}\}$

SOLUTION



Q3. (0.5 pt) Donnez une MT qui **reconnaît** mais ne **décide pas** le langage $\{0^n \mid n \in \mathbb{N}\}$

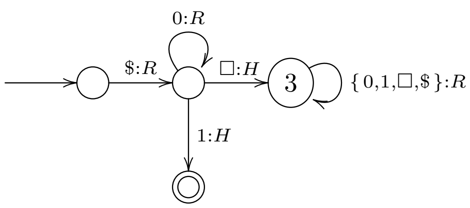
SOLUTION



La boucle sur l'état 4 est optionnelle.

Q4. (0.5 pt) Donnez une MT qui **reconnaît** mais ne **décide pas** le langage $\overline{\{0^n \mid n \in \mathbb{N}\}}$

SOLUTION



La boucle sur l'état 3 est optionnelle.

Q5. (1.5 pt) Considérons un langage L.

Si une MT M_1 décide le langage L alors le langage $\overline{L}$ est décidable.

Démontrez ce théorème :

(a) Commencez par répondre à la question : Que doit-on montrer ?

(b) Rédigez la preuve.

SOLUTION

(a) On doit montrer qu'il existe une MT M_2 qui décide le langage $\overline{L}$, c'est-à-dire que

(i) M_2 termine pour tout mot $\omega \in \{0,1\}^*$,

(ii) M_2 s'arrête sur $\circledcirc$ si $\omega \in \overline{L}$ ie. si $\omega \notin L$, ce qui correspond au cas où M_1 s'arrête en $\otimes$.

(iii) M_2 s'arrête sur $\otimes$ si $\omega \notin \overline{L}$ ie. si $\omega \in L$ ce qui correspond au cas où M_1 s'arrête en $\circledcirc$

(b) On construit la machine de Turing M_2 en prenant une copie de M_1 dans laquelle on remplace l'état $\circledcirc$ par $\otimes$ et $\otimes$ par $\circledcirc$. Ce changement n'affecte pas la terminaison puisqu'il n'y a pas de transitions sortantes de $\circledcirc$ et $\otimes$, donc (i) est satisfait. Ce changement inverse la réponse de M_1 ce qui satisfait (ii) et (iii).

Q6. (2 pt) Considérons un langage L.

Si une MT M_1 reconnaît mais ne décide pas le langage L et si une MT M_2 reconnaît mais ne décide pas le langage $\overline{L}$ alors on peut conclure que le langage L est décidable.

Démontrez ce théorème :

(a) Commencez par répondre à la question : Que doit-on montrer ?

(b) Rédigez la preuve.

SOLUTION

(a) On doit montrer qu'il existe une MT M qui décide le langage $\overline{L}$, c'est-à-dire que

(i) M termine pour tout mot $\omega \in \{0,1\}^*$,

(ii) M s'arrête sur $\circledcirc$ si $\omega \in L$, ce qui correspond au cas où M_1 s'arrête en $\circledcirc$.

(iii) M s'arrête sur $\otimes$ si $\omega \notin \overline{L}$, ce qui correspond au cas où M_2 s'arrête en $\circledcirc$

(b) On construit une machine de Turing M à deux bandes sur lesquelles on recopie ω. On exécutera M_1 sur B_1 et M_2 sur B_2. La MT M agit comme un ordonanceur qui exécute simultanément ou alternativement une transition de M_1 et une transition de M_2. Elle s'arrête lorsqu'une de deux machines de Turing atteint un état accepteur ; ce qui arrivera forcément quel que soit $\omega \in \{0,1\}^*$ puisque soit $\omega \in L$ et alors $M_1 \to^* \circledcirc$, soit $\omega \notin L$ et alors $M_2 \to^* \circledcirc$. Si c'est M_1 qui s'arrête la première en $\circledcirc$ alors M passe en $\circledcirc$; si c'est M_2 qui atteint la première l'état $\circledcirc$ alors M passe en $\otimes$.
On garde pour un futur examen la construction précise de l'ordonanceur ...

Exercice 2 : Opérations sur les booléens (4.5 pt)

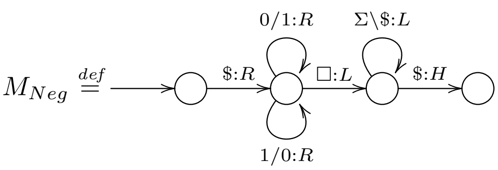
On considère l'alphabet $\{0,1,\Box,\$\}$ et on adopte la convention que 0 représente le booléen *faux* et 1 représente *vrai*. Un mot binaire $\omega \in \{0,1\}^*$ représente une suite de booléens.

Q7. (1.25 pt) Donnez une MT M_{Neg} qui effectue la négation des booléens de ω puis se replace sur le symbole $

Exemple :

$^\infty\Box$	\$	0	1	0	1	1	0	$\Box^\infty$	$\xrightarrow{M_{Neg}}$	$^\infty\Box$	\$	1	0	1	0	0	1	$\Box^\infty$
	$\uparrow$										$\uparrow$							

SOLUTION



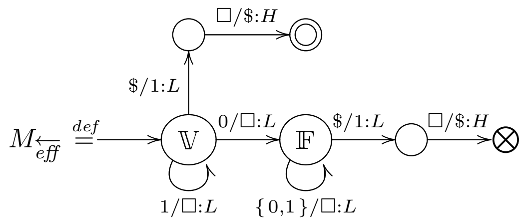
Q8. (2 pt) Donnez une MT M_{Conj} qui effectue la conjonction des booléens de ω, efface les booléens au fur et à mesure, inscrit 1 puis se replace sur le symbole \$ et termine dans un état ◎ si la conjonction vaut $\mathbb{V}$, inscrit 0 puis se replace sur le symbole \$ et termine dans un état ⊗ si la conjonction vaut $\mathbb{F}$.

Exemples :

- $^\infty\square \mid \square \mid \underset{\uparrow}{\$} \mid 1 \mid 0 \mid 0 \mid 1 \mid \square^\infty \xrightarrow{M_{Conj}} {}^\infty\square \mid \underset{\uparrow}{\$} \mid 0 \mid \square \mid \square \mid \square \mid \square \mid \square^\infty$
- $^\infty\square \mid \square \mid \underset{\uparrow}{\$} \mid 1 \mid 1 \mid 1 \mid 1 \mid \square^\infty \xrightarrow{M_{Conj}} {}^\infty\square \mid \underset{\uparrow}{\$} \mid 1 \mid \square \mid \square \mid \square \mid \square \mid \square^\infty$

SOLUTION

$M_{Conj} \overset{def}{=} M_{\overrightarrow{\square}} \,;\, M_{\overleftarrow{eff}}$ où $M_{\overleftarrow{eff}} \overset{def}{=}$



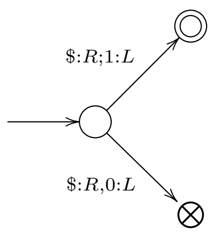
Q9. (1.25 pt) Exploitez le loi de *DeMorgan* $\neg(b_1 \vee b_2 \vee \ldots \vee b_n) = \neg b_1 \;\wedge\; \neg b_2 \;\wedge\; \ldots \;\wedge\; \neg b_n$ pour construire à une MT M_{Disj} telle que

$$M_{Disj}(\underbrace{b_1 b_2 \ldots b_n}_{\omega}) = b_1 \vee b_2 \vee \ldots \vee b_n$$

M_{Disj} doit effacer le ruban, inscrire le résultat sur le ruban, se replacer sur \$ et terminer dans un état correspond au résultat ◎ pour $\mathbb{V}$, ⊗ pour $\mathbb{F}$

SOLUTION

$M_{Disj} \overset{def}{=} M_{Neg} \,;\, M_{Conj} \,;\, M_{Neg} \,;$



Exercice 3 : Tableaux et tri en Gamma (3.5 pt)

En Gamma un tableau

i	0	1	2	3	4	5
$\mathtt{t}[i]$	2	521	3	42	4	2

est représenté par le multi-ensemble

$$\mathscr{M} = \{\, \text{T}(0,2), \text{T}(1,521), \text{T}(2,3), \text{T}(3,42), \text{T}(4,4), \text{T}(5,2) \,\}$$

où les éléments T*(indice,valeur)* flottent dans la solution chimique.

Q10. (0.5 pt) Donnez la propriété logique qui caractérise le fait que le tableau $\mathtt{t}[0..N]$ est trié dans l'ordre croissant sous la forme d'une formule commençant par $\forall$.

SOLUTION

$$\forall i,j \in [0..N],\; i \le j \Rightarrow \mathtt{t}[i] \le \mathtt{t}[j]$$

Q11. (0.5 pt) Donnez le multi-ensemble qui correspond au tableau T trié dans l'ordre croissant.

SOLUTION

$$\{\, \text{T}(0,2), \text{T}(1,2), \text{T}(2,3), \text{T}(3,4), \text{T}(4,42), \text{T}(5,521) \,\}$$

Q12. (1 pt) Donnez la/les règle(s) Gamma qui permettent d'obtenir un multi-ensemble d'éléments T*(indice,valeur)* trié dans l'ordre croissant.

SOLUTION

$$\mathrm{T}(i,v),\mathrm{T}(i',v') \xrightarrow{i \leq i' \wedge v > v'} \mathrm{T}(i,v'),\mathrm{T}(i',v)$$

Q13. (0.5 pt) Donnez la propriété logique qui caractérise le fait que le multi-ensemble $\mathscr{M}$ d'élements T*(indice,valeur)* est trié dans l'ordre croissant sous la forme d'une formule commençant par $\forall$.

SOLUTION

$$\forall\, \mathrm{T}(i,v),\mathrm{T}(i',v') \in \mathscr{M},\ i \leq i' \Rightarrow v \leq v'$$

Q14. (1 pt)

(a) Donnez l'exécution la plus efficace de cet algorithme sur le multi-ensemble $\mathscr{M}$ de l'exemple.

(b) Combien d'applications des règles sont nécessaires pour obtenir un multi-ensemble trié ;

(c) Combien d'étapes sont nécessaires pour obtenir le multi-ensemble trié.

SOLUTION

(a)
$$\mathrm{T}(0,2),\underbrace{\mathrm{T}(1,521),\mathrm{T}(5,2)}_{\downarrow},\mathrm{T}(2,3),\underbrace{\mathrm{T}(3,42),\mathrm{T}(4,4)}_{\downarrow}$$
$$\mathrm{T}(0,2),\overbrace{\mathrm{T}(1,2),\mathrm{T}(5,421)},\mathrm{T}(2,3),\overbrace{\mathrm{T}(3,4),\mathrm{T}(4,42)}$$

(b) 2 applications de la règle.

(c) 1 étape puisque la règle s'applique en parallèle.

Exercice 4 : Codage des Automates à une pile en machines de Turing

(8 pt)

L'objectif de l'exercice est de simuler un automate à une pile (AUP) par une machine de Turing à deux bandes. Le mot ω à reconnaître sera inscrit sur la bande B_1 et la pile de l'automate sera représentée par la bande B_2.

Q15. (0.75 pt) Rappelez la définition de **l'acceptation sur pile vide**. Autrement dit, donnez les conditions à satisfaire pour qu'un mot ω soit accepté par un AUP.

SOLUTION

*Un AUP A accepte un mot ω si **il existe** une **exécution** de A qui (i) commence dans l'**état initial** de A avec une **pile vide** ; (ii) **consomme toutes** les lettres du mot ; (iii) s'arrête dans un **état accepteur** avec une **pile vide**.*

Autant de 0 que de 1 On considère le langage L formé des mots binaires qui ont autant de 0 que de 1 (sans tenir compte de l'ordre de 0 et des 1).

Q16. (0.5 pt) Donnez trois mots binaires qui appartiennent au langage L et trois mots binaires qui n'appartiennent pas à L. En particulier, que dire du mot ϵ ?

SOLUTION

$$\epsilon, 01, 10 \in L$$
$$0, 1, 011 \notin L$$

Q17. (2 pt) Donnez un AUP qui reconnaît le langage L. Respectez les notations indiquées ci-après.

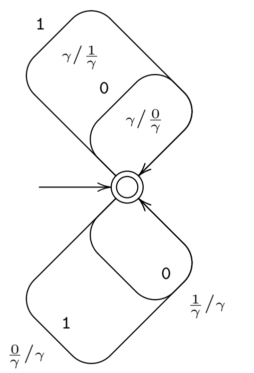
Indication : On notera $\mathbf{q} \xrightarrow[\gamma/\gamma']{\ell} \mathbf{q}'$ une transition d'AUP où ℓ est le symbole lu, γ l'état de la pile avant la transition et γ' l'état de la pile après la transition. Une transition peut effectuer l'une opérations suivantes sur la pile :

1. empiler un symbole : $\gamma / \frac{s}{\gamma}$
2. lire le sommet de la pile sans modifier la pile : $\frac{s}{\gamma} / \frac{s}{\gamma}$
3. dépiler le sommet : $\frac{s}{\gamma} / \gamma$
4. lire le sommet et empiler un symbole $\frac{s}{\gamma} / \frac{s'}{\frac{s}{\gamma}}$

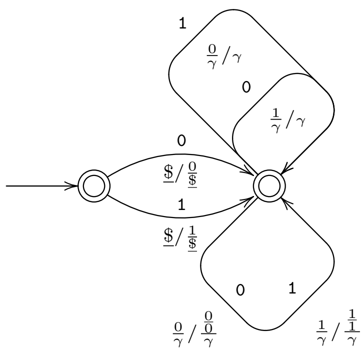
Exemples :

- La transition $\mathbf{q} \xrightarrow[\gamma/\frac{s}{\gamma}]{\ell} \mathbf{q}'$ lit le symbole ℓ et empile le symbole s
- La transition $\mathbf{q} \xrightarrow[\frac{s}{\gamma}/\gamma]{\ell} \mathbf{q}'$ lit le symbole ℓ et dépile le symbole s à condition que le symbole s soit en sommet de pile.
- La transition $\mathbf{q} \xrightarrow[\frac{\ell}{\gamma}/\frac{\ell}{\gamma}]{\ell} \mathbf{q}'$ lit le symbole ℓ et vérifie que le symbole en sommet de pile est le même que le symbole lu.

VERSION NON-DÉTERMINISTE



VERSION DÉTERMINISTE



Q18. (0.25 pt) Dessinez la bande B_2 correspondant à une pile vide.

SOLUTION

$^\infty\square$	\$	$\square^\infty$

$\uparrow$

Q19. (1.5 pt) Donnez la traduction en MT de chacune des trois transitions d'AUP suivantes : 1. $\mathbf{q} \xrightarrow[\gamma/\gamma]{\ell} \mathbf{q}'$ 2. $\mathbf{q} \xrightarrow[\frac{s}{\gamma}/\gamma]{\ell} \mathbf{q}'$ 3. $\mathbf{q} \xrightarrow[\gamma/\frac{\ell}{\gamma}]{\ell} \mathbf{q}''$. Respectez les notations indiquées ci-après.

> **Indication :** On notera les transitions de la MT à deux bandes de la manière suivante $\mathbf{q} \xrightarrow[\ell_2/e_2:d_2]{\ell_1/e_1:d_1} \mathbf{q}'$, c'est-à-dire au dessus de la flêche la *lecture/écriture* :*déplacement* sur la bande 1 et sous la flêche la *lecture/écriture* :*déplacement* sur la bande 2.

SOLUTION

1. $\mathbf{q} \xrightarrow[\gamma/\gamma]{\ell} \mathbf{q}' \quad \rightsquigarrow \quad \mathbf{q} \xrightarrow[\Sigma:H]{\ell:R} \mathbf{q}'$
2. $\mathbf{q} \xrightarrow[\frac{s}{\gamma}/\gamma]{\ell} \mathbf{q}' \quad \rightsquigarrow \quad \mathbf{q} \xrightarrow[s/\square:L]{\ell:R} \mathbf{q}'$
3. $\mathbf{q} \xrightarrow[\gamma/\frac{\ell}{\gamma}]{\ell} \mathbf{q}'' \quad \rightsquigarrow \quad \mathbf{q} \xrightarrow[\Sigma:R]{\ell:H} \mathbf{q}_\ell \xrightarrow[\square/\ell:H]{\ell:R} \mathbf{q}''$

Q20. (1.5 pt) Donnez les traductions en MT : (1) d'un état accepteur $\mathbf{q}$ de l'AUP, et (2) d'un état non-accepteur $\mathbf{q}$ de l'AUP de manière à simuler l'acceptation sur pile vide.

SOLUTION

- *Un état accepteur* $\mathbf{q}$ *de l'AUP est traduit en un état* $\mathbf{q}$ *dans la MT à laquelle on ajoute la transition suivante* $\mathbf{q} \xrightarrow[\$:H]{\square:H} \circledcirc$ *qui vérifie qu'on a consommé toutes les lettres de* ω *(partie* $\xrightarrow{\square:H}$*) et que la pile est vide (partie* $\xrightarrow[\$:H]{}$*). En réalité on ne vérifie pas que la pile est vide mais seulement que la tête de la bande 2 pointe sur le fond de la pile, symbolisé par le symbole \$.*
- *Les états non-accepteurs* $\mathbf{q}$ *sont laissés tel que et on ajoute la transition :* $\mathbf{q} \xrightarrow[\Sigma:H]{\square:H} \otimes$
- ***Bonus :*** *La transition précédente ne suffit pas à capturer le cas où l'AUP se trouve coincé dans l'état* $\mathbf{q}$ *avec encore des lettres de* ω *à lire et sans pouvoir effectuer aucune transition. Pour capturer ce cas il faut ajouter à la MT une transition sur* $\mathbf{q}$ *qui consomme toutes les lettres du mot avant de prendre la transition vers* $\otimes$.

$$\begin{tikzcd} \mathbf{q} \arrow[r, "\square:H", "\Sigma:H"'] \arrow[loop below, "\Sigma:R \,/\, \Sigma:H"] & \otimes \end{tikzcd}$$

La MT ainsi produite est alors non-déterministe. C'est la partie délicate de la traduction en MT.

Q21. (1.5 pt) Expliquez comment une MT peut simuler l'exécution d'un AUP A sur un mot ω de manière à accepter ω si A reconnaît ω.

> **Indication :** Vous pouvez faire référence aux questions précédentes dans vos explications.

SOLUTION

Pour que la MT M_A *simule l'exécution de l'AUP* A *sur le mot* ω*, on procède comme suit :*

1. *On copie le mot* ω *sur* B_1 *et on place* T_1 *sur le symbole \$. La bande 1 représente le mot à examiner.*
2. *On inscrit \$ sur* B_2 *et on place* T_2 *sur le symbole \$. La bande 2 représente la pile, qui est vide au départ puisque* $B_2 = \$$
3. *On traduit chacune des transitions de* A *en transitions de* M_A *comme indiquée dans les questions précédentes.*
4. *On ajoute les transitions vers* $\circledcirc$ *comme indiqué dans la question précédente. Les états non-accepteurs de l'AUP on les reproduit tels quels dans la MT.*

5. *On exécute la* MT *à deux bandes* M_A.
6. *Le mot* ω *est accepté si l'exécution* $M_A(\omega, \epsilon)$ *termine dans l'état* ◎.
7. ***Bonus :*** *Pour un mot* $\omega \notin \mathscr{L}(A)$ *la* MT *pourra se trouver bloquée dans un état* **q** *et ne rendra pas de résultat. Elle sera donc capable de reconnaître* $\mathscr{L}(A)$ *mais pas de décider* $\mathscr{L}(A)$.