

Cartouche du document

Année : ING 1 - Matière : Théorie des langages - Activité : Travail dirigé

Objectifs

Machine de Turing et langages contextuels.

Sommaire des exercices

- 1 - Machine de Turing et universalité
- 2 - Machine de Turing et reconnaissance de mots d'un langage
- 3 - JFLAP

Corps des exercices

1 - Machine de Turing et universalité

Énoncé :

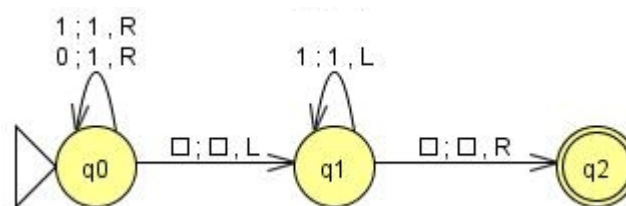
Dans cet exercice, on montre le côté universel de la machine de Turing : la résolution de problèmes quelconques. Dans les 2 prochains exercices, on utilise la machine de Turing comme un **transformateur**. Donc, à la fin de la modification de la chaîne, on doit placer le curseur au début du mot (JFLAP renvoi la chaîne à partir du curseur).

Question 1)

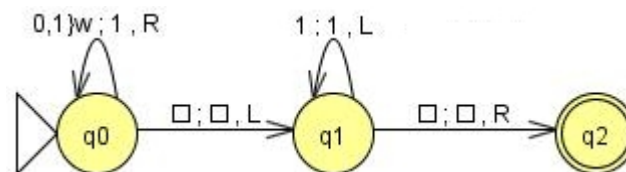
Énoncé de la question

Ecrire une machine de Turing permettant de remplacer tous les 0 d'un nombre binaire par des 1.

Solution de la question



Une autre manière d'écrire cet automate sous JFLAP :



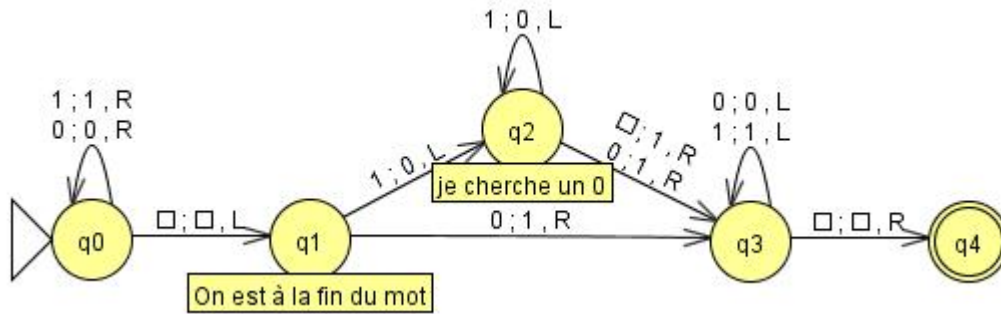
Question 2)

Énoncé de la question

Ecrire une machine de Turing permettant de calculer $X + 1$ (X est un mot binaire).

Solution de la question

Pour cet automate, il suffit de chercher le dernier 0 et de le remplacer par un 1. Pour cela, il faut tout d'abord se placer à la fin du mot puis remonter jusqu'à ce que l'on trouve un 0, en remplaçant les 1 par des 0 au fur et à mesure. Si l'on remonte jusqu'au début du mot, on rajoute un 1 devant le mot.



2 - Machine de Turing et reconnaissance de mots d'un langage

Énoncé :

Dans cet exercice, on utilise la machine de Turing comme machine à reconnaître des langages.

Question 1)

Énoncé de la question

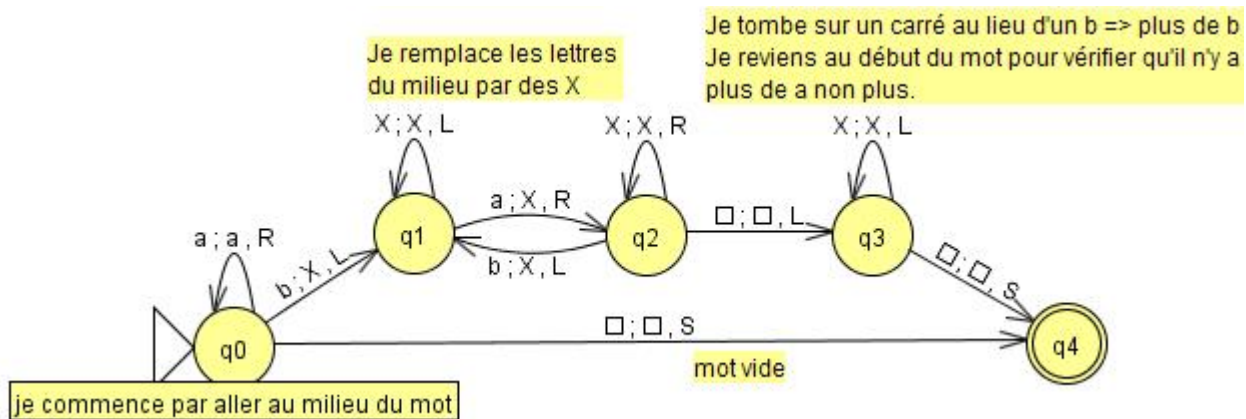
Soit l'alphabet $A = \{a, b\}$. Définir une machine de Turing permettant de reconnaître le langage

$$L = \{ a^n b^n / n \in \mathbb{N} \}.$$

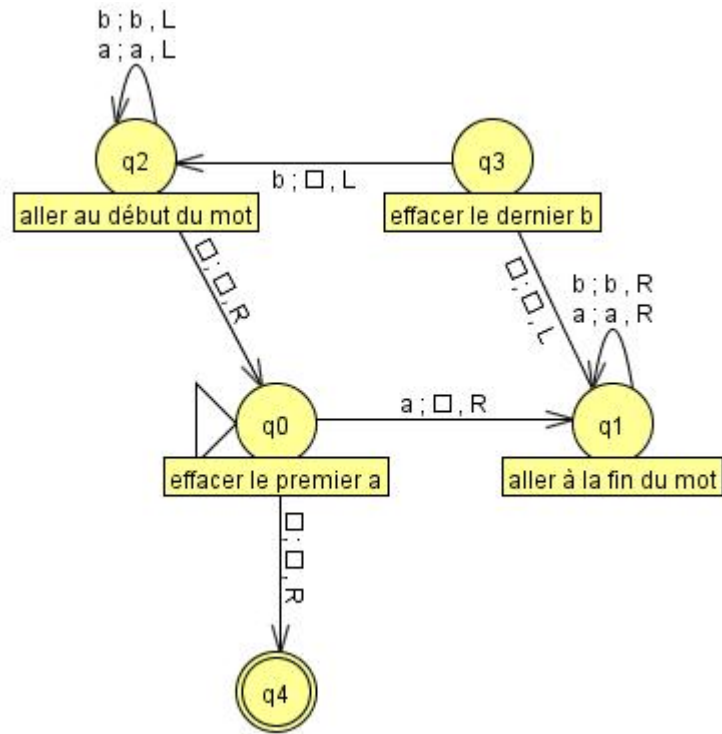
On rappelle que ce langage est de type 2 et qu'il peut donc être reconnu par un automate à pile.

Solution de la question

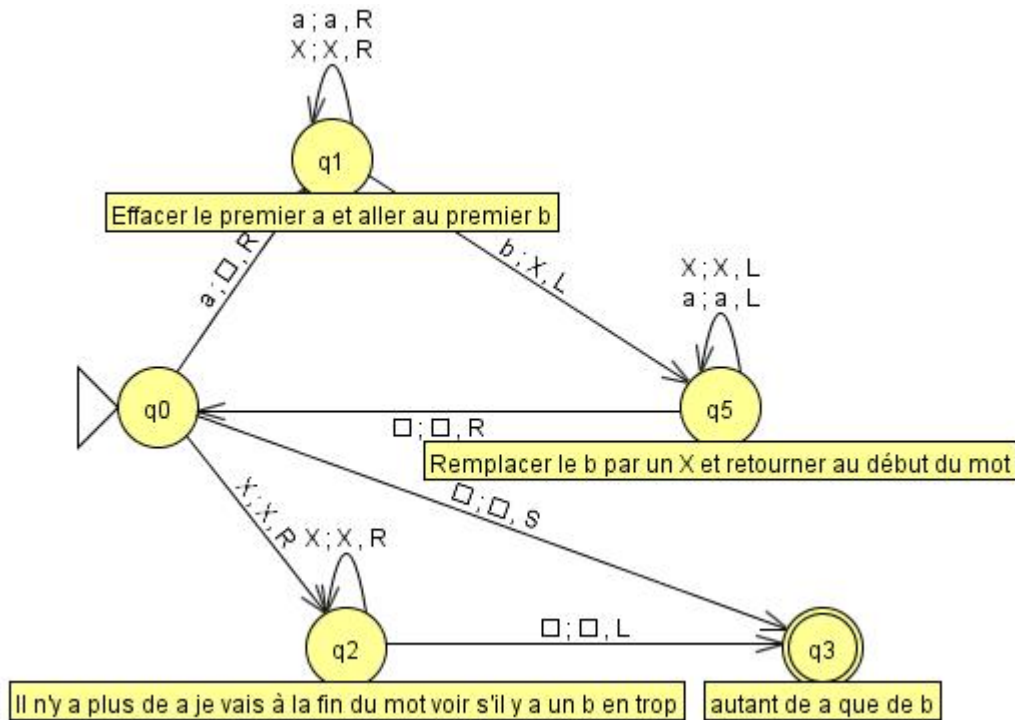
Dans cet automate, je pars du milieu du mot et je remplace le premier b par X, puis le dernier a par X, etc...



Une autre possibilité est de supprimer le premier a et le dernier b du mot, jusqu'à n'avoir plus que des carrés :



Une autre possibilité : je supprime les a en début de mot et je remplace les premiers b trouvés par X :



Question 2)

Énoncé de la question

Soit l'alphabet $A = \{a, b, c\}$. Définir une machine de Turing permettant de reconnaître le langage

$$L = \{ a^n b^n c^n / n \in \mathbb{N} \}.$$

On admettra que cet exemple est l'un des plus "simples" qui soit un "vrai" langage de type 1 (i.e : non reconnaissable par un automate à pile).

Pour rappel, une grammaire possible de ce langage (contextuelle) est la suivante :

$G = \{$

$T = \{a, b, c\}$

$N = \{S, T, Z, C\}$

$S = S$

$P = \{$

$S \rightarrow TZ // Z$ est utilisé pour repérer la fin du mot (puis à droite de Z on n'aura que des c)

$T \rightarrow aTbC \mid \epsilon //$ à chaque fois que l'on ajoute un a , on ajoute un b et un C (Problème : les b et les C seront alternés)

$Cb \rightarrow bC //$ On désentrelace les b et les C en faisant passer les C à droite des b

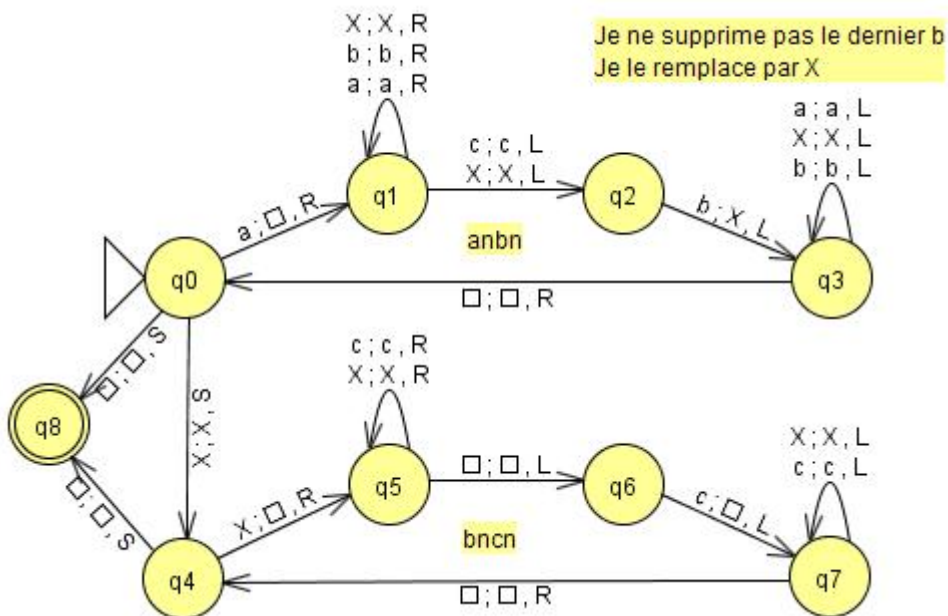
$CZ \rightarrow Zc //$ si on a un C à la fin du mot, on le remplace par un c et on décale le caractère de fin de mot : Z

$bZ \rightarrow b //$ si le caractère de fin de mot atteint les b , on n'a plus de C à transformer, on supprime donc Z

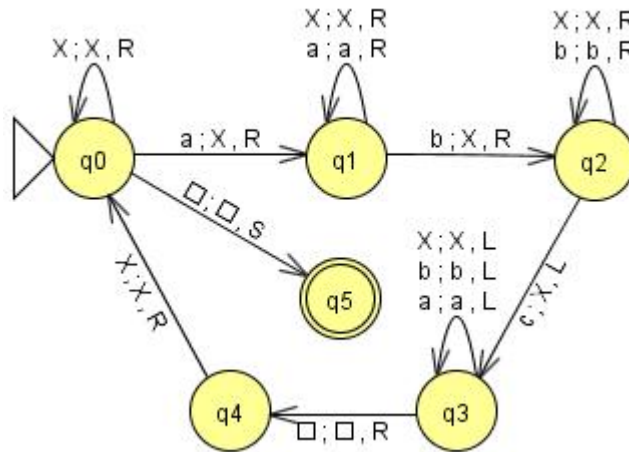
$\}$
 $\}$

Solution de la question

Pour cet exercice, il suffit de reprendre l'automate précédent et d'exécuter 2 fois la procédure pour tester $a^n b^n$ puis $b^n c^n$



Une autre possibilité est de faire les 3 en un seul passage, en remplaçant tout par X :



Question 3)

Énoncé de la question

Définir une machine de Turing permettant de reconnaître le langage

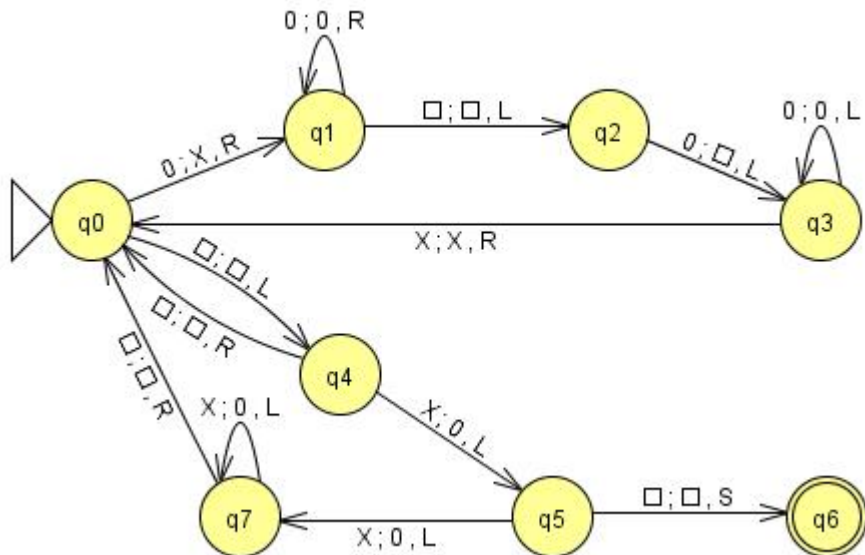
$$L = \{ 0^n / n = 2^p \}.$$

C'est à dire les mots composés d'un nombre de 0 qui soit une puissance de 2.

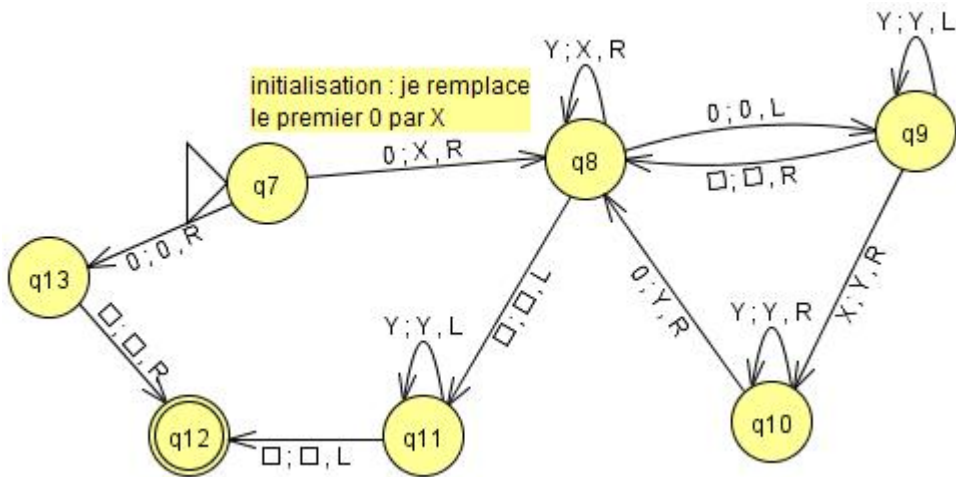
Solution de la question

Pour cet exercice assez difficile, une fois l'astuce trouvée, l'automate est très simple. Il suffit en effet de diviser par 2 le nombre de 0 à chaque étape. Si on finit par n'avoir plus qu'un seul 0 alors le mot est accepté.

Pour diviser les 0 par 2, je remplace le premier par un X et je supprime le dernier.



Une autre possibilité est de raisonner par multiplication plutôt que par division. On part du mot 0 qu'on transforme en X, puis on crée autant de Y à la place des 0 restant qu'il y a de X. Si on ne peut pas en créer assez, c'est qu'il n'y a pas suffisamment de 0.



Question 4)

Énoncé de la question

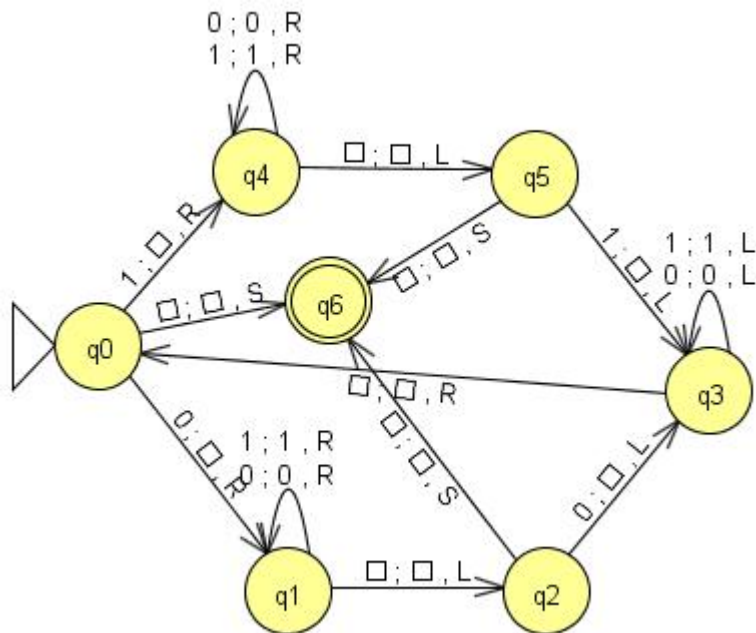
Dans cette question, on utilise l'alphabet binaire.

Définir une machine de Turing permettant de reconnaître les palindromes (un mot qui se lit identiquement dans les deux sens).

Exemple : 0010100.

Solution de la question

A l'instar de la reconnaissance du langage $a^n b^n$, on vérifie si la première lettre du mot correspond bien à la dernière, puis on les supprime et on réitère jusqu'à n'avoir plus de lettres.



3 - JFLAP

Énoncé :

Comme pour les autres travaux dirigés, vous utiliserez l'utilitaire JFLAP (<http://www.jflap.org/jflaptmp>).

Remarques sur JFLAP :

- si on considère la machine de Turing comme un transformateur, la chaîne de sortie est lue à partir du curseur. Il faut donc revenir au début du mot avant d'arrêter la machine ;
- dans une machine de Turing :
 - le symbole \sim désigne n'importe quel caractère (ATTENTION ! le carré blanc qui dénote une case vide fait partie des caractères). Par exemple, une transition $\sim;\sim,R$ effectue un pas vers la droite indépendamment du caractère lu et sans modifier celui-ci ;
 - dans une transition $!x$ désigne n'importe quel symbole sauf x ;
 - dans une transition pour désigner tout sauf le symbole carré blanc, on écrit $!$;
 - La transition " $s_1, \dots, s_n \} w; w, R$ " signifie que si on rencontre un des symboles s_i il ne sera pas changé. w désigne la dernière variable lue et peut être utilisé pour d'autres transitions ;
- Les noms par défaut des états q_0, \dots, q_i peuvent être changés (clic droit sur l'état). On peut également ajouter des étiquettes en dessous des états (clic droit sur l'état).

Question 1)

Énoncé de la question

Nous pouvons créer des Machines de Turing en JFLAP (<http://jflap.org/jflaptmp>). Pour cela, on utilise le bouton *Turing Machine* du menu initial. Ensuite, il suffit de créer l'automate.

Solution de la question

Se conférer à la réponse globale de l'exercice

Question 2)

Énoncé de la question

La machine de Turing peut être utilisée comme un transformateur ou comme un accepteur. Pour les 2 premiers exercices, elle est utilisée comme transformateur. Pour la tester on peut lui soumettre un ensemble de chaînes d'entrées dans l'item *Multiple Run (Transducer)* du menu *Input*.

Solution de la question

Se conférer à la réponse globale de l'exercice

Question 3)

Énoncé de la question

Dans les exercices suivants, la Machine de Turing est utilisée comme un accepteur afin de reconnaître des langages. Pour la tester on peut lui soumettre un ensemble de chaînes d'entrées dans l'item *Multiple Run* du menu *Input*.

Solution de la question

Se conférer à la réponse globale de l'exercice