TD6 - Machines de Turing et langages contextuels

Objectifs

- Machine de Turing et langages contextuels.
- Aspect universel de la machine de Turing : résolution de problèmes quelconques.

Rappels et notations

Comme pour les autres travaux dirigés, vous utiliserez l'utilitaire JFLAP (http://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. Par exemple, une transition $\sim; \sim, R$ effectue un pas vers la droite indépendemment du caractère lu et sans modifier celui-ci
    ATTENTION ! le carré blanc, $\square$, qui dénote une case vide, fait partie des caractères
  - 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 "$\mathbf{s}_1,\ \ldots\ ,\mathbf{s}_n\ \}\mathbf{w}\ ;\mathbf{w},\mathbf{R}$" signifie que si on rencontre un des symboles $\mathbf{s}_i$ il ne sera pas changé. $\mathbf{w}$ désigne la dernière variable lue et peut être utilisé pour d'autres transitions
- Les noms par défaut des états q_0, ..., 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).

Nous pouvons créer des machines de Turing en JFLAP . Pour cela, on utilise le bouton *Turing Machine* du menu initial. Ensuite, il suffit de créer l'automate.

La machine de Turing peut être utilisée comme un transformateur ou comme un accepteur.

- Pour le premier exercice, 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*.
- Dans le second, 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*.

Exercice 1. Machine de Turing et universalité

Dans cet exercice, 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 renvoie la chaîne à partir du curseur).

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

Question 2. Ecrire une machine de Turing permettant de calculer X + 1 (X est un mot binaire).

Exercice 2. Machine de Turing et reconnaissance de mots d'un langage

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

Question 1. Soit l'alphabet $A = \{a, b\}$.
Définir une machine de Turing permettant de reconnaître le langage $L = \{a^n b^n / n \in N\}$.

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

Question 2. Soit l'alphabet $A = \{a, b, c\}$. Définir une machine de Turing permettant de reconnaître le langage $L = \{a^n b^n c^n / n \in N\}$.

Note : 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 $G =< T, N, S, P >$ avec :

$T = \{a, b, c\}$
$N = \{S, T, Z, C\}$ les états de l'automate
$S = S$

$$P = \left\{ \begin{array}{lll} S \rightarrow & T\ Z & // \ Z \text{ est utilisé pour repérer la fin du mot (puis à droite de } Z \text{ on n'aura que des } c) \\ T \rightarrow & a\ T\ b\ C \mid \varepsilon & // \text{ à chaque fois que l'on ajoute un } a\text{, on ajoute un } b \text{ et un } C \\ & & \quad \text{(Problème : les } b \text{ et les } C \text{ seront alternés)} \\ C\ b \rightarrow & b\ C & // \text{ On désentrelace les } b \text{ et les } C \text{ en faisant passer les } C \text{ à droite des } b \\ C\ z \rightarrow & Z\ c & // \text{ si on a un } C \text{ à la fin du mot, on le remplace par un } c \text{ et on décale} \\ & & \quad \text{le caractère de fin de mot : } Z \\ b\ Z \rightarrow & b & // \text{ si le caractère de fin de mot atteint les } b\text{, on n'a plus de } C \text{ à transformer,} \\ & & \quad \text{on supprime donc } Z \end{array} \right\}$$

Question 3. Définir une machine de Turing permettant de reconnaître le langage $L = \{0^n / n = 2^p\}$ des mots composés d'un nombre de 0 qui soit une puissance de 2.

Question 4. Dans cette question, on utilise l'alphabet binaire, $A = \{0, 1\}$.

Définir une machine de Turing permettant de reconnaître les palindromes, c'est à dire les mots qui se lisent identiquement dans les deux sens. Exemple : 0010100.