Calculabilité - Décidabilité (ICC)

Interrogation n°2 du 4 novembre 2019 - Automates finis

Seul le polycopié de cours est autorisé - Durée : 40 min

Le barème est donné à titre indicatif

Le sujet se décompose en 4 exercices indépendants. La qualité de la rédaction, la clarté et la précision des raisonnements entreront pour une part importante dans l'évaluation de la copie. Il conviendra de bien détailler les étapes d'un algorithme et non pas de donner directement le résultat.

Exercice 1 (6 pts).

1. Construire une grammaire hors-contexte générant le langage

$$L_1 = \{w \in \{a,b\}^* \mid w \text{ contient autant de } a \text{ que de } b\}.$$

2. Construire une grammaire hors-contexte générant le langage

$$L_2 = \{w \in \{a,b\}^* \mid w \text{ contient au moins autant de } a \text{ que de } b\}.$$

3. Construire une grammaire hors-contexte générant le langage

$$L_3 = \{w \in \{a,b\}^* \mid w \text{ contient strictement plus de } a \text{ que de } b\}.$$

Solution :

1. $S \to SS \mid aSb \mid bSa \mid \varepsilon$
2. $S \to SS \mid aSb \mid bSa \mid a \mid \varepsilon$
3. $S \to TaT,\ T \to TT \mid aTb \mid bTa \mid a \mid \varepsilon$

Exercice 2 (4 pts). Soit la grammaire $G = (\{S, A\}, \{a, b\}, P, S)$ où P est donné par

$$\begin{array}{lcl} S & \to & aAA \\ A & \to & aS \mid bS \mid \varepsilon \end{array}$$

Convertir la grammaire ci-dessus en un automate à pile reconnaissant le même langage par état final.

Solution : On utilise l'algorithme vu en TD.

Soit $G = (V, T, Q, S)$ une grammaire hors-contexte. Soit $P = (\{q_i, q, q_f\}, T, V \cup T, \delta, Z, q_i, \{q_f\})$ un AP avec δ définie par

— On a : $\delta(q_i, \varepsilon, Z) = (q, SZ)$

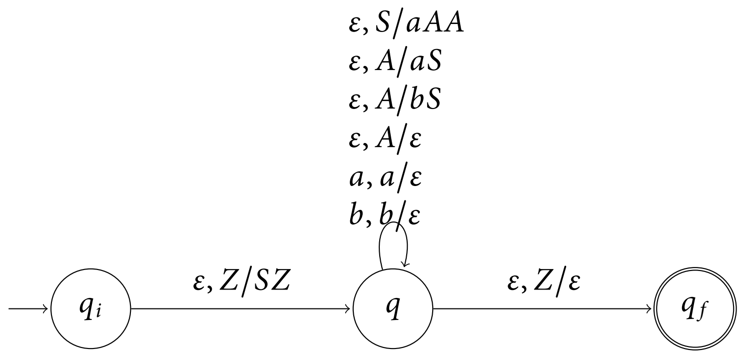
— Pour toute variable A

$$\delta(q, \varepsilon, A) = \{(q, \beta) \mid A \to \beta \text{ est une production de } G\}$$

— Pour tout symbole terminal a, $\delta(q, a, a) = (q, \varepsilon)$

— On a : $\delta(q, \varepsilon, Z) = (q_f, \varepsilon)$

Alors, P reconnait $L(G)$ par état final.



Exercice 3 (4 pts)**.** On considère l'alphabet $\Sigma = \{a, b, c\}$.

1. Construire un automate à pile reconnaissant le langage

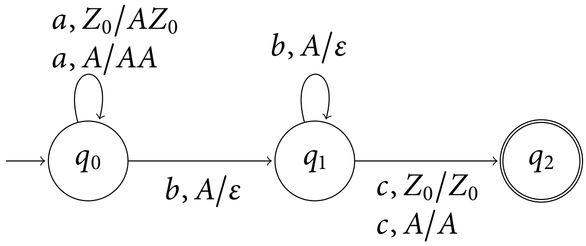
$$\{a^n b^m c : n \geq m \geq 1\}.$$

   Expliquer *rapidement* le principe de cet automate. Préciser son mode d'acceptation.

2. Exécuter votre automate sur l'entrée $aaabc$ (en explicitant les différentes configurations de l'automate).

Solution :

1. On construit l'automate à pile acceptant par état final suivant : $\mathcal{A} = (Q, \Sigma, Z, T, q_0, Z_0, \{q_2\})$ avec



   Dans l'état initial, l'automate lit les a du mot, et empile un symbole A sur la pile à chaque fois. Lorsque l'automate lit le premier b (il y a alors nécessairement un A au sommet de la pile, car il faut avoir lu au moins un a, l'automate passe dans l'état q_1, et efface le symbole A en sommet de pile pour chaque b lu. Ainsi, on s'assure que le mot contient au plus autant de b que de a, car lorsque la pile ne contient plus de A on ne peut plus lire de b. Lorsque l'automate lit c, on passe alors dans l'état acceptant dans lequel on ne peut plus lire aucune lettre.

2. $(q_0, aaabc, Z_0) \vdash (q_0, aabc, AZ_0) \vdash (q_0, abc, AAZ_0) \vdash (q_0, bc, AAAZ_0) \vdash (q_1, c, AAZ_0)$
   $\vdash (q_2, \varepsilon, AAZ_0)$

Exercice 4 (6 pts)**.** Soit la grammaire $G = (\{S, W, X, Y, Z\}, \{a, b\}, P, S)$ où P est donné par

$$\begin{array}{lcl} S & \rightarrow & aXb \mid aYbb \mid aZbbb \\ W & \rightarrow & aXb \mid aWbb \mid \varepsilon \\ X & \rightarrow & aXb \mid aX \mid \varepsilon \\ Y & \rightarrow & aYbb \mid Yb \mid \varepsilon \\ Z & \rightarrow & aZb \mid aZbb \end{array}$$

1. Mettre la grammaire G sous forme normale de Chomsky. Pour cela, vous expliquerez les différentes étapes de la mise sous forme normale de Chomsky et vous les détaillerez.

2. En utilisant l'algorithme CYK vu en cours, tester si le mot $abbb$ appartient à $L(G)$.

3. Quel est donc le langage engendré par cette grammaire ?

Solution :

1. On voit clairement que les variables W et Z sont inutiles. Cela donne la grammaire

$$\begin{array}{lcl} S & \rightarrow & aXb \mid aYbb \\ X & \rightarrow & aXb \mid aX \mid \varepsilon \\ Y & \rightarrow & aYbb \mid Yb \mid \varepsilon \end{array}$$

   En éliminant les ε-productions (X et Y sont les seules variables annulables), on obtient la grammaire

$$\begin{array}{lcl} S & \rightarrow & aXb \mid aYbb \mid ab \mid abb \\ X & \rightarrow & aXb \mid aX \mid ab \mid a \\ Y & \rightarrow & aYbb \mid Yb \mid abb \mid b \end{array}$$

   qui est sous forme simplifiée. Sa forme normale de Chomsky est donc

$$\begin{array}{lcl} S & \rightarrow & AC \mid AE \mid AB \mid AD \\ X & \rightarrow & AC \mid AX \mid AB \mid a \\ Y & \rightarrow & AE \mid YB \mid AD \mid b \\ A & \rightarrow & a \\ B & \rightarrow & b \\ C & \rightarrow & XB \\ D & \rightarrow & BB \\ E & \rightarrow & YD \end{array}$$

2. En appliquant l'algorithme CYK vu en cours, on contruit le tableau suivant :

$$\begin{array}{cccc} \{S,Y\} & & & \\ \{S,Y,C\} & \{E,Y\} & & \\ \{S,X,C\} & \{Y,D\} & \{Y,D\} & \\ \{X,A\} & \{Y,B\} & \{Y,B\} & \{Y,B\} \\ \hline a & b & b & b \end{array}$$

   On en déduit alors que $abbb \in L(G)$

3. Le langage engendré par cette grammaire est $L = \{a^n b^m \mid m \leq n \text{ ou } n \leq 2m,\ \text{avec } m, n \geq 1\}$.