TD grammaires

17 novembre 2023

On supposera dans ce problème que toutes les grammaires considérées ne font jamais apparaitre leur symbole initial dans un membre droit d'une règle.

1. Expliquer comment on peut se ramener à cela sans perte de généralité.

On supposera de plus qu'elles ne font intervenir que des non terminaux utiles c'est-à-dire qui apparaissent dans une dérivation du symbole initial à un mot généré.

1 Grammaire $\mathcal{G}_2$

On définit $\mathcal{G}_2 = (\mathcal{V}_2, \Sigma_2, S_2, R_2)$ avec $\mathcal{V}_2 = \{S_2, E, T\}$, $\Sigma_2 = \{a, +, *\}$ et R_2 :

$$S_2 \to E$$
$$E \to E + T | T$$
$$T \to T * a | a$$

2. Est ce que le mot $a * a + a * a$ est généré par $\mathcal{G}_2$? Si oui, donner un arbre de dérivation pour ce mot et une dérivation gauche.
3. Est ce que le mot $a + a * a + a$ est généré par $\mathcal{G}_2$? Si oui, donner un arbre de dérivation pour ce mot et une dérivation droite.
4. On considère la grammaire $\mathcal{G}_2'$ ayant les mêmes caractéristiques que $\mathcal{G}_2$ mais de symbole initial T. Donner (sans justification) une expression régulière décrivant le langage engendré par $\mathcal{G}_2'$.
5. On considère la grammaire $\mathcal{G}_2''$ ayant les mêmes caractéristiques que $\mathcal{G}_2$ mais de symbole initial E. Donner (sans justification) une expression régulière décrivant le langage engendré par $\mathcal{G}_2''$.
6. En déduire une description du langage engendré par $\mathcal{G}_2$ et justifier que cette grammaire n'est pas ambigüe. Montrer, à l'aide d'un shéma, la forme d'un arbre de dérivation de cette grammaire.
7. Ecrire un programme en C de signature `bool genere_2(char* m)` qui prend en entrée une chaîne de caractères quelconque et qui renvoie `true` si et seulement si celle-ci correspond à un mot généré par $\mathcal{G}_2$. On attend une complexité linéaire en la longueur de `m`. Il n'est pas attendu de s'appuyer sur les règles de la grammaire mais uniquement sur le langage engendré qui a été identifié.

2 Grammaire $\mathcal{G}_1$ et réductions

Considérons la grammaire suivante : $\mathcal{G}_1 = (\mathcal{V}_1, \Sigma_1, S, R_1)$ avec $\mathcal{V}_1 = \{S, R, T\}$, $\Sigma_1 = \{a, b\}$ et R_1 :

$$S \to R | T$$
$$R \to aRb | ab$$
$$T \to aTbb | abb$$

8. Montrer proprement que le langage engendré par $\mathcal{G}_1$ est inclus dans $\{a^n b^n : n \in \mathbb{N}^*\} \cup \{a^n b^{2n} : n \in \mathbb{N}^*\}$.
9. Démontrer l'inclusion réciproque afin de caractériser le langage engendré par $\mathcal{G}_1$.
10. Donner une dérivation gauche depuis le symbole initial pour le mot $aaabbb$.
11. Donner une dérivation droite depuis le symbole initial pour le mot $aabbbb$.
12. Ecrire un programme en C de signature `bool genere_1(char* m)` qui prend en entrée une chaîne de caractères quelconque et qui renvoie `true` si et seulement si celle-ci correspond à un mot généré par $\mathcal{G}_1$. On attend une complexité linéaire en la longueur de `m`. Il n'est pas attendu de s'appuyer sur les règles de la grammaire mais uniquement sur le langage engendré qui a été identifié.

    On définit maintenant la notion de réduction comme l'opération contraire à la dérivation. On dit que pour $\mathcal{G} = (\mathcal{V}, \Sigma, S, R)$, $u \in (\mathcal{V} \cup \Sigma)^*$ se réduit en $v \in (\mathcal{V} \cup \Sigma)^*$ ssi v se dérive en u. On note cette réduction $u \hookrightarrow^* v$. Une réduction directe se notera $u \hookrightarrow v$ et correspondra à $v \Rightarrow u$.

Une réduction de u est une réduction de u jusqu'au symbole initial.

On dit qu'une réduction de u est gauche ssi, à chaque étape, on réduit le facteur le plus à gauche possible.

13. Donner une réduction gauche pour le mot $aabbbb$ dans $\mathcal{G}_1$.
14. Montrer qu'une réduction gauche correspond à une dérivation droite pour toute grammaire.
15. Montrer que si une grammaire est non ambigüe alors tout mot engendré par celle-ci admet une unique réduction gauche.
16. Montrer que si on a $u \hookrightarrow v$ alors il existe $x, h, y \in (\mathcal{V} \cup \Sigma)^*$ et une règle $T \to h \in R$ tels que $u = xhy$ et $v = uTy$. On dit que h est un facteur réductible de u.
17. Soit $w = u_1 \hookrightarrow u_2 \hookrightarrow u_3 \ldots \hookrightarrow u_k$ une réduction gauche avec $w \in \Sigma^*$. On a alors pour chaque u_i l'existence d'une factorisation sous la forme $u_i = xhy$ et d'une règle de la forme $T \to h$ tels que $u_{i+1} = xTy$. Justifier que y est composé uniquement de symboles terminaux.

   On appelle mot valide un mot $m \in (\mathcal{V} \cup \Sigma)^*$ pour lequel il existe $u \in \mathcal{L}(\mathcal{G})$ tel que m apparait dans une réduction gauche de u.
18. Dire (et justifier) si les mots suivants sont valides pour la grammaire $\mathcal{G}_1$:
    (a) $aaRbb$
    (b) $aTbb$
    (c) $aaRbbbbb$
    (d) $aaabbbbbb$
19. Pour tout mot valide m, on appelle facteur réductible gauche, un facteur réductible de m situé le plus à gauche possible tel que si on lui applique la réduction on obtient un mot valide.

    Pour chacun des mots valides identifiés dans la question précédente, donnez son facteur réductible gauche.
20. Une grammaire est dite déterministe si pour tout mot m valide de $(\mathcal{V} \cup \Sigma)^*$, il existe une unique réduction directe gauche vers un mot valide de règle $T \to h$ avec $m = xhy$ et que pour tout $y' \in \Sigma^*$, cette même réduction est l'unique réduction gauche de $m' = xhy'$ vers un mot valide quand m' est valide.

    Justifier à l'aide des mots $aaabbb$ et $aaabbbbbb$ que $\mathcal{G}_1$ n'est pas déterministe.