Département d'informatique
Faculté des sciences
Universitée de Sherbrooke

DEVOIR 5

Par
Samuel Hamann (19 079 596)
Zachary Déziel (17 022 847)

Travail présenté à
Djemel Ziou

Dans le cadre du cours
IFT313
Introduction aux langages formels

Sherbrooke
August 13, 2020

Exercice 1

Soit $L = \{\lambda, a, ab, a^2b^2, ..., a^nb^n, ...\}$.

1. Donner la grammaire qui engendre L

   Réponse:

   La grammaire qui engendre L est:

$$G : S \to a|A|\lambda, \quad A \to ab|aAb$$

2. Est-ce que G est une grammaire hors contexte? Est-elle régulière? Est-elle normale de Chomsky? Est-elle sous la forme normale de Greibach? Est-elle SLL(k)? Justifier votre réponse.

   Réponse:

   Grammaire hors contexte:

   Selon le théorème vu en classe, G est une grammaire hors contexte si $\forall w \in L(G), \exists$ une dérivation gauche de w.

   Il suffit de donner un contre exemple pour prouver que G n'est pas une GHC. Aucun contre-exemple évident donc preuve par induction pour démontrer que c'est une GHC.

   On doit vérifier tous les mots possibles de L(G) peuvent être obtenu par une dérivation gauche, c'est-à-dire qu'il n'exige que de remplacer le symbole le plus à gauche pour obtenir le mot.

   Cas de base:

   $w = \lambda$: $S \to \lambda$ donc correct pour le théorème.

   $w = a$: $S \to a$ donc correct pour le théorème.

   $w = ab$: $S \to A \to ab$ donc correct pour la dérivation gauche.

   Induction:

   $w = a^nb^n \Rightarrow$w=a$^{n-1}Ab^{n-1}$, ce qui est obtenu avec la dérivation gauche $S \to A \to^{n-1} aAb \to a^nb^n$

   Fermerture:

   On applique la récurrence un nombre fini de fois pour obtenir tout $w \in L(G)/\lambda, a$ Il n'y a aucun autre $w \in L(G)$ pas couverte pas les cas de base où la récurrence.

   Tous les mots de $L(G)$ peuvent être obtenus par une dérivation gauche donc la grammaire est une grammaire hors contexte.

   Grammaire régulière

   La grammaire est régulière si elle décrit un langage régulier.

   On peut utiliser le Lemme de l'Étoile pour démontrer qu'un langage n'est pas régulier:

   Posons $z = aabb$. z peut être décomposé de la sorte que $z = uvw$ où $u = aa$, $v = b$ et $w = b$. Pour que $L(G)$ soit un langage régulier, $uv^nw \in L(G) \forall n \geq 0$. Ceci est faux pour $n = 2$, le langage n'est pas régulier et la grammaire n'est pas régulière.

   Forme normale de Chomsky

   Pour qu'une grammaire soit sous le forme normale de Chomsky, les règles de production doivent être sous l'une de ses formes:

- $A \to BC$
- $A \to a$
- $S \to \lambda$

Le symbole initial ne peut être récursif.

La règle $A \to aAb$ ne respecte pas une de ses formes et elle ne peut être décomposée en plusieurs règles pour respecter les conditions de la forme normale de Chomsky.

Forme normale de Greibach

Pour qu'une grammaire soit sous le forme normale de Greibach, les règles de production doivent être sous l'une de ses formes:

- $A \to aA_1A_2..A_n$
- $S \to \epsilon$

où $\epsilon \in \Sigma*$.

La règle $A \to aAb$ ne répond pas à ces conditions, donc la grammaire n'est pas sous la forme normale de Greibach.

SLL(k)

La grammaire est récursive gauche donc elle n'est pas SLL(k) selon un théorème vu en cours.

3. Démontrer que G engendre L.

   Réponse:

   Cas de base:

   $p = \lambda$: engendré par la règle $S \to \lambda$.

   $p = a$: engendré par la règle $S \to a$.

   $p = ab$: engendré par les règles $S \to A \to ab$

   On pourrait exclure le prochain cas des cas bases, mais montre l'utilisation de la règle $A \to aAb$ qui serait appliqué à la récurrence.

   $p = aabb$: engendré par les règles $S \to AaAb \to aabb$.

   Récurrence:

   $p = a^{n+1}b^{n+1} \Rightarrow aa^nb^nb$. Le mot peut-être obtenu étant donné que nous avons une règle $A \to aAb$ et $A \to ab$.

   Fermeture:

   On obtient un mot $z \in L(G)$ en appliquant la récurrence un nombre fini de fois.

4. Donner un automate à pile équivalent à G. Vous devez fournir le diagramme d'état et les fonctions de transitions.

   Réponse:

   $PDA = (Q, \Sigma, t, \delta, q_0, F)$

   $\Sigma = \{a, b\}$

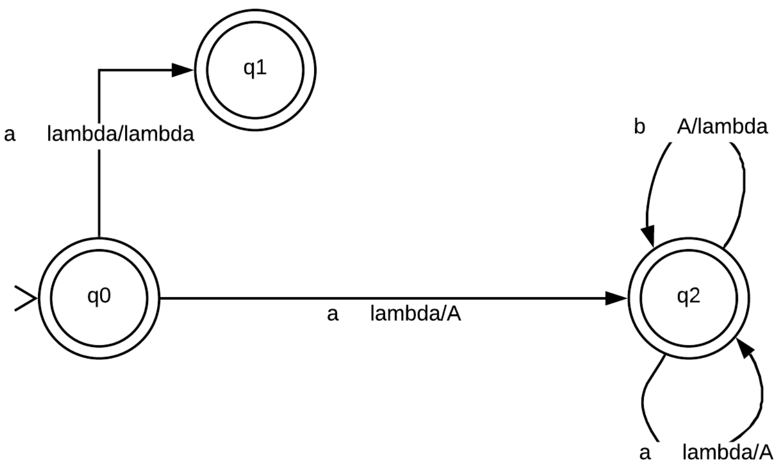
   $Q = \{q_o, q_1\}$

   t est un ensemble fini de symbole (un autre alphabet)

$\delta = Q \times (\Sigma \cup \{\lambda\}) \times (t \cup \{\lambda\}) \to Q \times (t \cup \{\lambda\})$

$F = \{q_0, q_1\}$

Diagramme d'états



Fonctions de transitions

$\delta(q_0, a, \lambda) = \{[q_1, \lambda], [q_2, A]\}$

$\delta(q_2, b, A) = \{[q_2, \lambda]\}$

$\delta(q_2, a, \lambda) = \{[q_2, A]\}$

5. En considérant l'automate à pile qui accepte L, exprimer le calcul du mot p=aabb à l'aide des changements de configuration.

p : aabb Pile : $\begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline\end{array}$

Transition utilisée : $\delta(q_0, a, \lambda) = \{[q_2, A]\}$

p : abb Pile : $\begin{array}{|c|}\hline \\ \hline \\ \hline A \\ \hline\end{array}$

Transition utilisée : $\delta(q_2, a, \lambda) = \{[q_2, A]\}$

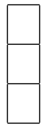
p : bb Pile : $\begin{array}{|c|}\hline \\ \hline A \\ \hline A \\ \hline\end{array}$

Transition utilisée : $\delta(q_2, b, A) = \{[q_2, \lambda]\}$

p : b Pile : $\begin{array}{|c|}\hline \\ \hline \\ \hline A \\ \hline\end{array}$

Transition utilisée : $\delta(q_2, b, A) = \{[q_2, \lambda]\}$

p : λ $\quad$ *Pile* :



Exercice 2

Soient $G1 : S \to A\$, \quad A \to TB, \quad B \to Z|\lambda, \quad T \to b|(A)$ et $G2 : S \to ABC, \quad A \to aA|a, B \to bB|\lambda, \quad C \to cC|a|b|c$. Les majuscules sont les variables. Les minuscules, \$, +, (et) sont les terminaux.

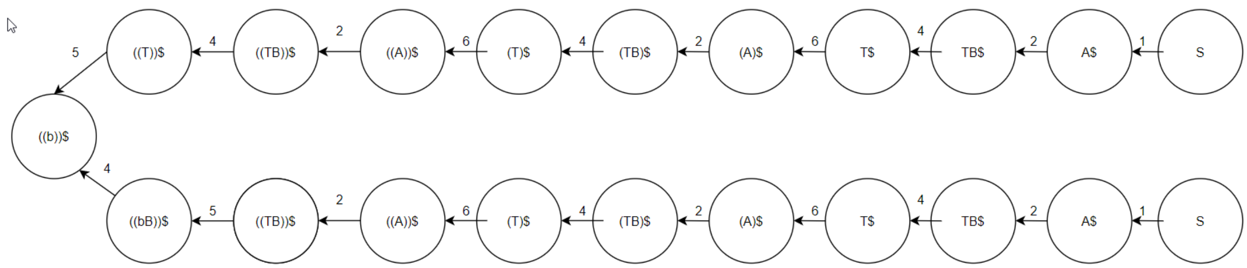
1. Pour $G1$ et le mot $p = ((b))\$$, donner les deux arbres de dérivation (graphes de grammaire) pour les analyses syntaxiques ascendante et descendante.

   Réponse:

   Graphe de grammaire de dérivation ascendante

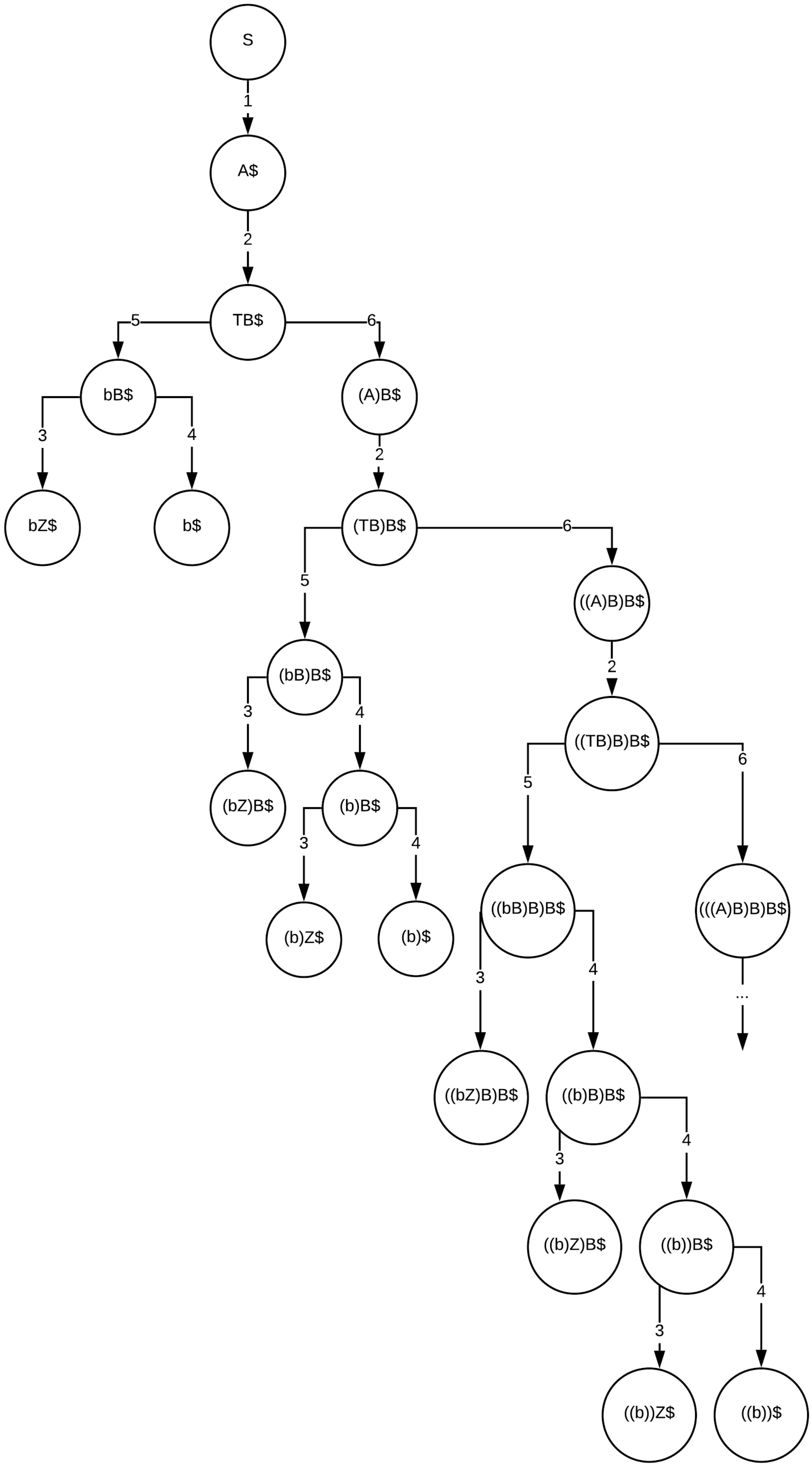
   Bien sûr, il y a toutes les options où on peut insérer un B, mais ces prototypes ne sont pas intéressant pour mener vers des mots potentiels. La règle de $B \to Z$ ne mène pas vers un mot. Il faut quand même souligner qu'il aurait beaucoup plus de branches que les deux présentés.

   Note: Vous pouvez trouver l'image attaché à la remise si la résolution n'est pas suffisante.



   Graphe de grammaire de dérivation descendante

   À la prochaine page



S
1
A$
2
TB$
5
6
bB$
(A)B$
3
4
bZ$
b$
2
(TB)B$
5
6
((A)B)B$
(bB)B$
2
3
4
((TB)B)B$
(bZ)B$
(b)B$
5
6
3
4
(b)Z$
(b)$
((bB)B)B$
(((A)B)B)B$
3
4
...
((bZ)B)B$
((b)B)B$
3
4
((b)Z)B$
((b))B$
3
4
((b))Z$
((b))$

2. Pour $G2$, déterminer la valeur de k pour que l'intersection des ensembles suivants liés à la variable A soit vide. Vous devez le faire aussi pour la variable B et pour la variable C.

   Réponse:

   Construction de $LA(S)$

   Plus facile de voir $LA(S)$ avant d'établir $LA(A)$, $LA(B)$ et $LA(C)$.

   $LA(S) = \{a^+b^*(c^* \cup a \cup b \cup c)\} = \{a^+b^*(c^*a \cup c^*b \cup c^+)\}$

   Pour la variable A

   $LA_1(A \to aA) = \{a\}$ et $LA_1(A \to a) = \{a\}$, donc $k = 1$ n'est pas suffisant pour distinguer les ensembles suivants des règles à partir du symbole A.

   $LA_2(A \to aA) = \{aa\}$ et $LA_2(A \to a) = \{aa, ab, ac\}$, $aa \in LA_2(A \to aA) \wedge aa \in LA_2(A \to a)$. Donc, $k = 2$ n'est toujours pas suffisant pour distinguer les ensembles suivant des règles de A.

   $LA_3(A \to aA) = \{aaa\}$ et $LA_3(A \to a) = \{aa, ab, ac, acc, aba, abb, abc\}$, le seul élément de $LA_3(A \to aA)$ n'est pas dans $LA_3(A \to a)$. Donc $k = 3$ est suffisant pour que l'intersection des ensembles suivant de A soient vide.

   Pour la variable B

   Trouver en adaptant $LA(S)$ en fonction de la règle B utiliser.

   $LA(B \to bB) = \{a^+b^+(c^*a \cup c^*b \cup c^+)\}$ et $LA(B \to \lambda) = \{a^+(c^*a \cup c^*b \cup c^+)\}$

   Il y a une différence suivant le a^+. k dépendra donc du nombre de a précédent l'utilisation de la règle B. Posons n qui est égale au nombre de a précédent l'utilisation de la règle B: $n = |a| \wedge a \geq 1$.

   Considérons $k = n + 1$ pour voir si un seul caractère est suffisant après le nombre de a. $LA_n + 1(B \to bB) = \{a^nb\}$ et $LA_n(B \to \lambda) = \{a^n(a \cup b \cup c)\}$. L'intersection des deux ensembles n'est pas vide. En fait, $a^nb \in LA_n + 1(B \to bB) \wedge a^nb \in LA_n + 1(B \to \lambda)$.

   Considérons $k = n+2$. $LA_n+2(B \to bB) = \{a^nbb\}$ et $LA_n(B \to \lambda) = \{a^n(a \cup ca \cup b \cup cb \cup c \cup cc)\}$. Dans ce cas-ci $n + 2$ est suffisant pour que l'intersection des ensembles suivants de la variable B soit vide.

   Pour la variable C

   Trouver en adaptant $LA(S)$ en fonction de la règle C utiliser.

   $LA(C \to cC) = \{a^+b^*(c^+a \cup c^+b \cup c^+)\}$

   $LA(C \to a) = \{a^+b^*a\}$

   $LA(C \to b) = \{a^+b^*b\}$

   $LA(C \to c) = \{a^+b^*c\}$

   Similaire que pour la variable B, posons n le nombre de a de a^+ $(n \geq 1)$ et m le nombre de b $(m \geq 0)$ précédent l'utilisation d'une règle de B.

   Considérons $k = n + m + 1$. L'intersection des règles $LA_{n+m+1}(C \to a)$ et $LA_{n+m+1}(C \to b)$ et l'intersection de ses règles avec les deux autres produisent des ensembles vides. Par contre, l'intersection des règles $LA_{n+m+1}(C \to cC)$ et $LA_{n+m+1}(C \to c)$ n'est pas vide. En fait l'intersection est $\{a^+b^*c\}$.

   Considérons $k = n + m + 2$.
   $LA_{n+m+2}(C \to cC) = \{a^*b^+aa, a^*b^+ab, a^*b^+ac, a^*b^+ba, a^*b^+bb, a^*b^+bc, a^*b^+ca, a^*b^+cb, a^*b^+cc\}$

$LA_{n+m+2}(C \to c) = \{a^*b^+a, a^*b^+b, a^*b^+c\}$
Donc, $LA_{n+m+2}(C \to cC) \cap LA_{n+m+2}(C \to c) = \{\}$

L'intersection des règles $LA_{n+m+2}(C \to cC)$ et $LA_{n+m+2}(C \to c)$ est vide. $k = n + m + 2$ est suffisant pour que l'intersection des ensembles suivants de C soit vide.

3. Soit G une grammaire hors contexte dont l'intersection des ensembles suivants pour chaque variable est vide, montrer par l'absurde que G n'est pas ambiguë. Est-ce que l'inverse est vrai? Justifier votre réponse en fournissant un contre-exemple ou une démonstration.

   Réponse:

   Preuve initiale

   Lemme: Si G est ambiguë, il existe deux règles de production dela grammaire (G) A et B de la sorte que $p = AB \land p = BA$.

   p: G est une GHC dont l'intersection des ensembles suivant des règles est l'ensemble vide.

   q: G est ambiguë.

   Si $p = AB$, $p \in LA(A)$ et, si $p = BA$, $p \in LA(B)$. Donc p est dans l'ensemble suivant des deux règles de production. Ceci est une contradiction avec le fait que notre grammaire est une grammaire hors contexte.

   Preuve de la réciproque

   p: G est une grammaire ambiguë.

   q: G est une GHC dont l'intersection des ensembles suivant des règles est l'ensemble vide.

   G est une GHC dont l'intersection des ensembles suivant des règles est l'ensemble vide implique qu'il n'existe pas un mot $p \in L(G)$ qui peut être obtenu par deux dérivations différentes tel que $p = AB$ ou $p = BA$. Ceci est une contradiction avec le lemme qui défini une grammaire ambiguë.

4. Implanter la fonction $First_k$ en utilisant C++ ou Python. Pour éviter de manipuler des cas particulier, vous pouver transformer une règle $S \to w$ en une règle $S \to w^k$ avec un marqueur qui se répète k fois. La grammaire en entrée est ni ambiguë, ni récursive à gauche.