Grammaires & formes normales

Corrigé partiel de la feuille de travaux dirigés n^{o}7

1. On cherche d'abord les variables productives, et on trouve $P = \{X, Z\}$. On doit donc supprimer S et Y. On obtient la grammaire :

$$
\begin{array}{l}
N = \{X, Z\} \\
T = \{a, b, c\} \\
- \\
P \left\{ \begin{array}{l} X \to aX \mid c \\ Z \to aZ \mid c \end{array} \right.
\end{array}
$$

Dans cette grammaire nous n'avons plus d'axiome, c.a.d. qu'aucune variable n'est accessible, donc on ne peut pas dériver aucun mot, donc $L = \emptyset$.

2. a) La grammaire n'a pas l'air d'être linéaire. La première chose à faire est de la nettoyer. Nous supprimons la variable improductive F. On supprime A non-accessible. On obtient :

$$
N = \{S, B, D, E, G\}, \quad T = \{0, 1\}, \quad S, \quad P = \left\{ \begin{array}{l} S \to 1B \mid 0E \\ B \to 0D \mid 1S \mid 0 \mid 1 \\ D \to G \mid 1E \\ E \to 0S \mid 1D \mid 0 \mid 1 \\ G \to 0B \end{array} \right.
$$

La substitution de G rend la grammaire linéaire, d'où l'automate fini :

δ	0	1
$\to q_0$	q_1	q_2
q_1	$\{q_0, q_4\}$	$\{q_3, q_4\}$
q_2	$\{q_3, q_4\}$	$\{q_0, q_4\}$
q_3	q_2	q_1
$\leftarrow q_4$		

b) Pour trouver l'expression régulière on peut résoudre un système associé à l'automate. Cependant ce système s'avère être lourd à résoudre. En effet, il est autrement plus simple de déterminiser et minimiser d'abord l'automate fini, pour obtenir :

δ	0	1
$\to q_0$	q_1	q_1
q_1	q_2	q_2
$\leftarrow q_2$	q_1	q_1

Ainsi il est facile d'obtenir l'expression : $L = ((0 + 1)^2)^+$.

3. a) Cette grammaire n'est pas ambiguë. Il est facile de montrer que le langage engendré est $a^\star b(a + b)^\star$. La seule manière d'obtenir un mot du langage est d'utiliser la première règle de manière à couper le mot lors du premier b. La partie qui précède (et qui contient que des a ne peut être dérivée que d'une seule manière. De même pour la deuxième partie.

b) Cette grammaire est ambiguë. En effet on peut avoir la dérivation : $S \to XaSbY \to XaXaSbYbY \to XaXabYbY$, qui permet d'obtenir par la suite le mot $aaabbb$ de quatre manières (selon si le a manquant est obtenu à partir du premier ou du second X, est de même pour le b manquant.

4. B est non-productive, on peut donc la supprimer. On obtient $\langle N = \{S, A, C\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to CA \\ A \to a \\ C \to b \end{cases}$$ Le langage engendré par cette grammaire est $\{ba\}$.

5.
a) C est non-productive, on peut la supprimer. On obtient $G = \langle N = \{S, A, B\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to aAa \\ A \to Sb \mid bBB \\ B \to abb \end{cases}$$

b) productions sous forme normale de Chomsky :

$$P = \begin{cases} S \to DC & E \to b \\ C \to AD & F \to BB \\ D \to a & B \to DG \\ A \to SE \mid EF & G \to EE \end{cases}$$

c) productions sous forme normale de Greibach :

$$P = \begin{cases} S \to aAa \\ A \to aAab \mid bBB \\ B \to abb \end{cases}$$

6. S_5, S_6, S_7, S_8 ne sont pas productives. Leur suppression donne $\langle N = \{S_1, S_2, S_3, S_4\}, T = \{a, b\}, S_1, P\rangle$:

$$P = \begin{cases} S_1 \to S_1S_2 \mid S_2 \mid S_3 \\ S_2 \to bS_2 \mid \varepsilon \\ S_3 \to bS_3 \mid \varepsilon \\ S_4 \to bS_2 \mid S_3S_4 \end{cases}$$

S_4 n'est pas accessible. Sa suppression donne $\langle N = \{S_1, S_2, S_3\}, T = \{a, b\}, S_1, P\rangle$:

$$P = \begin{cases} S_1 \to S_1S_2 \mid S_2 \mid S_3 \\ S_2 \to bS_2 \mid \varepsilon \\ S_3 \to bS_3 \mid \varepsilon \end{cases}$$

S_2 et S_3 sont identiques. On obtient donc $\langle N = \{S_1, S_2\}, T = \{a, b\}, S_1, P\rangle$:

$$P = \begin{cases} S_1 \to S_1S_2 \mid S_2 \\ S_2 \to bS_2 \mid \varepsilon \end{cases}$$

On peut lire directement le résultat. Le langage qu'on peut dériver à partir de S_2 est $b^\star$ et pour S_1 $(b^\star)^+ = b^\star$ (dans ce cas précis).

7. Les seules réponses possibles sont le **a)** et le **d)**. Il reste à montrer, que seule la réponse **a)** est correcte.
La question qu'on se pose est : *peut-on créer des symboles non-productifs lors de la suppression des non-accessibles ?*.
Supposons que le symbole X devienne non-productif suite à la suppression d'un symbole Y non-accessible. Dans ce cas, X est accessible (autrement le problème ne se pose pas), et donc Y doit aussi être accessible à travers X ! Ce qui règle le problème.

8. Il faut dans un premier temps supprimer les ε-productions de $\langle N = \{S, A, B\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to Sa \mid Sb \mid Aa \\ A \to Bb \mid b \mid Aa \\ B \to Bb \mid Ba \mid b \mid a \end{cases}$$

Ensuite on supprime la récursivité gauche en B, $\langle N = \{S, A, B\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to Sa \mid Sb \mid Aa \\ A \to Bb \mid b \mid Aa \\ B \to bC \mid aC \mid b \mid a \\ C \to bC \mid aC \mid b \mid a \end{cases}$$

B et C sont identiques $\langle N = \{S, A, B\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to Sa \mid Sb \mid Aa \\ A \to Bb \mid b \mid Aa \\ B \to bB \mid aB \mid b \mid a \end{cases}$$

Suppression de la récursivité gauche en A, $\langle N = \{S, A, B, C\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to Sa \mid Sb \mid Aa \\ A \to Bb \mid b \mid BbC \mid bC \\ C \to a \mid aC \\ B \to bB \mid aB \mid b \mid a \end{cases}$$

Suppression de la récursivité gauche en S , $\langle N = \{S, A, B, C, D\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to Aa \mid AaD \\ D \to a \mid b \mid aD \mid bD \\ A \to Bb \mid b \mid BbC \mid bC \\ C \to a \mid aC \\ B \to bB \mid aB \mid b \mid a \end{cases}$$

Il suffit de faire les substitutions (dans l'ordre B dans A puis A dans S) pour obtenir,
$\langle N = \{S, A, B, C, D\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to aBba \mid bBba \mid aba \mid bba \mid ba \mid aBbCa \mid bBbCa \mid abCa \mid bbCa \mid bCa \\ S \to aBbaD \mid bBbaD \mid abaD \mid bbaD \mid baD \mid aBbCaD \mid bBbCaD \mid abCaD \mid bbCaD \mid bCaD \\ D \to a \mid b \mid aD \mid bD \\ A \to aBb \mid bBb \mid ab \mid bb \mid b \mid aBbC \mid bBbC \mid abC \mid bbC \mid bC \\ C \to a \mid aC \\ B \to aB \mid bB \mid a \mid b \end{cases}$$

A ce stade on remarque que A n'est plus accessible, donc on la supprime,
$\langle N = \{S, B, C, D\}, T = \{a, b\}, S, P\rangle$:

$$P = \begin{cases} S \to aBba \mid bBba \mid aba \mid bba \mid ba \mid aBbCa \mid bBbCa \mid abCa \mid bbCa \mid bCa \\ S \to aBbaD \mid bBbaD \mid abaD \mid bbaD \mid baD \mid aBbCaD \mid bBbCaD \mid abCaD \mid bbCaD \mid bCaD \\ D \to a \mid b \mid aD \mid bD \\ C \to a \mid aC \\ B \to aB \mid bB \mid a \mid b \end{cases}$$

Remarque 1 : Selon la méthode vue en cours il fallait effectuer les substitutions au fur et mesure. En effet, dans le cas général, cela est nécessaire car la substitution elle même peut créer d'autres récursions gauches. Dans cet exercice nous avons pu supprimer les récursions gauches puis effectuer les substitutions car, hormis les récursions gauches (les boucles), le *graphe des dépendances* était sans circuit, donc les substitutions ne pouvaient créer de nouvelles boucles.

Remarque 2 : une autre manière de traiter ce problème - plus simple ici, mais pas généralisable consiste à **deviner** d'abord le langage. En effet, il est assez facile dans ce cas de réaliser que B permet de dériver les mots de $(a + b)^\star$, que A permet de dériver les mots de la forme $Bba^\star = (a + b)ba^\star$. Enfin, de S on peut dériver $Aa(a + b)^\star = (a + b)^\star ba^\star a(a + b)^\star$, ce qui se simplifie en $a + b)^\star ba(a + b)^\star$. Il est facile de donner la grammaire sous forme normale de Greibach qui correspond à l'automate non-déterministe qui reconnaît ce langage :

$N = \{S, X, Y\}$
$T = \{a, b\}$
S

$$P = \begin{cases} S \to aS \mid bS \mid bX \\ X \to aY \mid a \\ Y \to a \mid b \mid aY \mid bY \end{cases}$$

9. On commence par constater que dans cette grammaire toutes les variables sont productifs et accessibles et qu'il n'y a pas de ε-production, ni de renommage.

$$P = \begin{cases} A \to Aa \mid Ab \mid Ca \mid a \\ B \to Aa \mid Bb \\ C \to Ba \mid Cb \end{cases}$$

On commence de rendre la grammaire montante (l'ordre sur les variables sera l'ordre alphabétique).
On transforme la récursion gauche en A en recursion droite :

$$P = \begin{cases} A \to Ca \mid a \mid CaD \mid aD \\ B \to Aa \mid Bb \\ C \to Ba \mid Cb \\ D \to aD \mid bD \mid a \mid b \end{cases}$$

A est maintenant montante, on substitue le résultat dans les règles de B
$B \to Bb \mid Caa \mid aa \mid CaDa \mid aDa$
puis on transforme la récursion gauche en B en recursion droite :

$$P = \begin{cases} A \to Ca \mid a \mid CaD \mid aD \\ B \to Caa \mid aa \mid CaDa \mid aDa \mid CaaE \mid aaE \mid CaDaE \mid aDaE \\ C \to Ba \mid Cb \\ D \to aD \mid bD \mid a \mid b \\ E \to bE \mid b \end{cases}$$

B est maintenant montante, on substitue le résultat dans les règles de C
$C \to Caaa \mid aaa \mid CaDaa \mid aDaa \mid CaaEa \mid aaEa \mid CaDaEa \mid aDaEa \mid Cb$
puis on transforme la récursion gauche en C en recursion droite :

$$P = \begin{cases} A \to Ca \mid a \mid CaD \mid aD \\ B \to Caa \mid aa \mid CaDa \mid aDa \mid CaaE \mid aaE \mid CaDaE \mid aDaE \\ C \to aaa \mid aDaa \mid aaEa \mid aDaEa \mid aaaF \mid aDaaF \mid aaEaF \mid aDaEaF \\ D \to aD \mid bD \mid a \mid b \\ E \to bE \mid b \\ F \to aaa \mid aDaa \mid aaEa \mid aDaEa \mid b \mid aaaF \mid aDaaF \mid aaEaF \mid aDaEaF \mid bF \end{cases}$$

La grammaire est maintenant montante, il suffit de faire les substitutions de C dans les règles de B et A, pour obtenir :

$$\begin{cases} A \to & aaaa \mid aDaaa \mid aaEaa \mid aDaEaa \mid aaaFa \mid aDaaFa \mid aaEaFa \mid aDaEaFa \mid aaaaD \mid \\ & aDaaaD \mid aaEaaD \mid aDaEaaD \mid aaaFaD \mid aDaaFaD \mid aaEaFaD \mid aDaEaFaD \mid a \mid aD \\ B \to & aaaaa \mid aDaaaa \mid aaEaaa \mid aDaEaaa \mid aaaFaa \mid aDaaFaa \mid aaEaFaa \mid aDaEaFaa \mid aa \mid \\ & aaaaDa \mid aDaaaDa \mid aaEaaDa \mid aDaEaaDa \mid aaaFaDa \mid aDaaFaDa \mid aaEaFaDa \mid \\ & aDaEaFaDa \mid aDa \mid aaaaaE \mid aDaaaaE \mid aaEaaaE \mid aDaEaaaE \mid aaaFaaE \mid aDaaFaaE \mid \\ & aaEaFaaE \mid aDaEaFaaE \mid aaE \mid aaaaDaE \mid aDaaaDaE \mid aaEaaDaE \mid aDaEaaDaE \mid \\ & aaaFaDaE \mid aDaaFaDaE \mid aaEaFaDaE \mid aDaEaFaDaE \mid aDaE \\ C \to & aaa \mid aDaa \mid aaEa \mid aDaEa \mid aaaF \mid aDaaF \mid aaEaF \mid aDaEaF \\ D \to & aD \mid bD \mid a \mid b \\ E \to & bE \mid b \\ F \to & aaa \mid aDaa \mid aaEa \mid aDaEa \mid b \mid aaaF \mid aDaaF \mid aaEaF \mid aDaEaF \mid bF \end{cases}$$

ce qui est sous forme normale de Greibach.

Une petite remarque aurait permis d'obtenir une autre grammaire beaucoup plu simple sous formale de Greibach !

En effet, nous pouvons remarquer, que si on prend l'image mirroir de toutes les règles, nous aurons une grammaire linéaire droite :

$$\langle N = \{A, B, C\}, T = \{a, b\}, A, P\rangle \quad P = \begin{cases} A \to aA \mid bA \mid aC \mid a \\ B \to aA \mid bB \\ C \to aB \mid bC \end{cases}$$

Ceci correspond à l'automate nondéterministe

		a	b
$\to$	A	A, C, D	A
	B	A	B
	C	B	C
$\leftarrow$	D	$-$	$-$

On déterminise et minimise, pour obtenir :

		a	b
$\to$	A	D	A
$\leftarrow$	D	D	A

Ce qui permet de conclure que le langage reconnu est $(a + b)^\star a$.

Ainsi, le langage engendré par la grammaire a était le mirroir, c.à.d. $a(a+b)^\star$, ce qui peut être engendré par la grammaire suivant sous forme normale de Greibach :

$\langle N = \{S, X\}, T = \{a, b\}, S, P\rangle$

$$P = \begin{cases} S \to aX \mid a \\ X \to aX \mid bX \mid a \mid b \end{cases}$$