TD2 : Grammaires et analyse syntaxique

Exercice 1

$$S \to aSa \mid bSb \mid cSc \mid a \mid b \mid c \mid \varepsilon$$

Exercice 2

C'est le langage des mots qui ont autant de a que de b.

Exercice 3

1. Premier$(C) = \{c, d\}$, Premier$(B) = \{b\}$, Premier$(A) = \{a, \varepsilon\}$, Premier$(S) = \{a, b, c, d\}$.
   Suivant$(S) = \{\$\}$, Suivant$(C) = \{a\}$, Suivant$(B) = \{\$\}$, Suivant$(A) = \{b\}$.
2.

	a	b	c	d	$\$$
S	$S \to AB$	$S \to AB$	$S \to Ca$	$S \to Ca$	accepter
A	$A \to aAb$	$A \to \varepsilon$	erreur	erreur	erreur
B	erreur	$B \to bB$, $B \to b$	erreur	erreur	erreur
C	erreur	erreur	$C \to cC$	$C \to d$	erreur

3. Non car la case (B, b) contient deux règles.

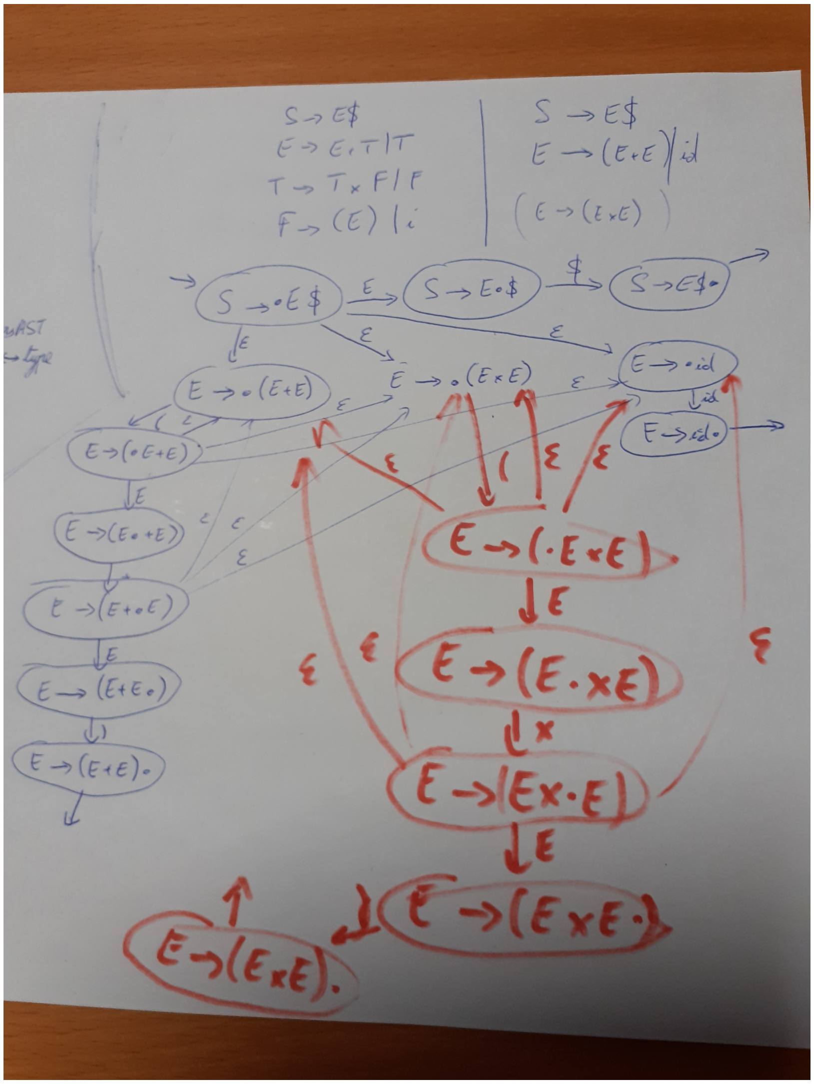
Exercice 4

1. En notation "traditionnelle", on obtient $1 + (2 + 3) \times 4 = 21$.
2. On utilise une pile. On parcourt le mot de gauche à droite. Si on voit un entier, on l'empile. Si on voit un opérateur $\otimes$, on dépile les deux derniers entiers n, p et on empile $n \otimes p$ à la place. (On renvoie une erreur s'il n'y avait pas deux entiers en haut de la pile.)
3. $S \to SS+ \mid SS\times \mid id$.
4. On fait comme en 2, sauf qu'on empile des arbres. Quand on voit un a, on empile l'arbre à deux sommets dont la racine est S et la feuille a. Quand on voit un opérateur $\otimes$, on dépile les deux derniers arbres T_1, T_2 et on empile un arbre dont la racine est S et les trois fils sont T_1, T_2 et une feuille $\oplus$. À la fin, la pile ne contient qu'un arbre, qui est l'arbre de dérivation du mot lu.

Exercice 5

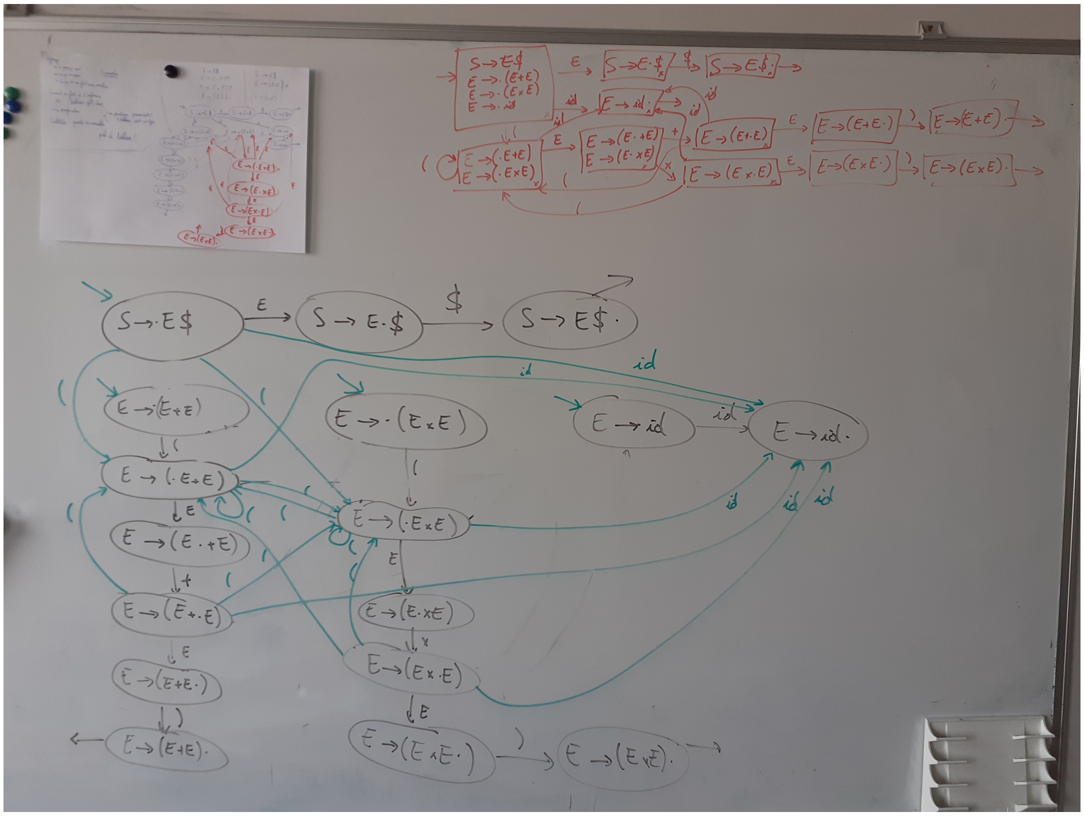
1. Dans l'algorithme précédent, on applique une opération shift à chaque étape. Si on lit un a, on effectue directement un *reduce* de la règle $S \to a$. Si on lit un opérateur $\oplus$, on effectue un *reduce* de la règle $S \to SS\oplus$.
2. La pile contient un arbre dont la racine est T, puis un $+$ et il reste à lire $a \times a$. On ne peut plus finir l'analyse car aucune règle n'a $T+$ en début de son côté droit. Ici on aurait dû réduire T en E avant de shift le $+$.

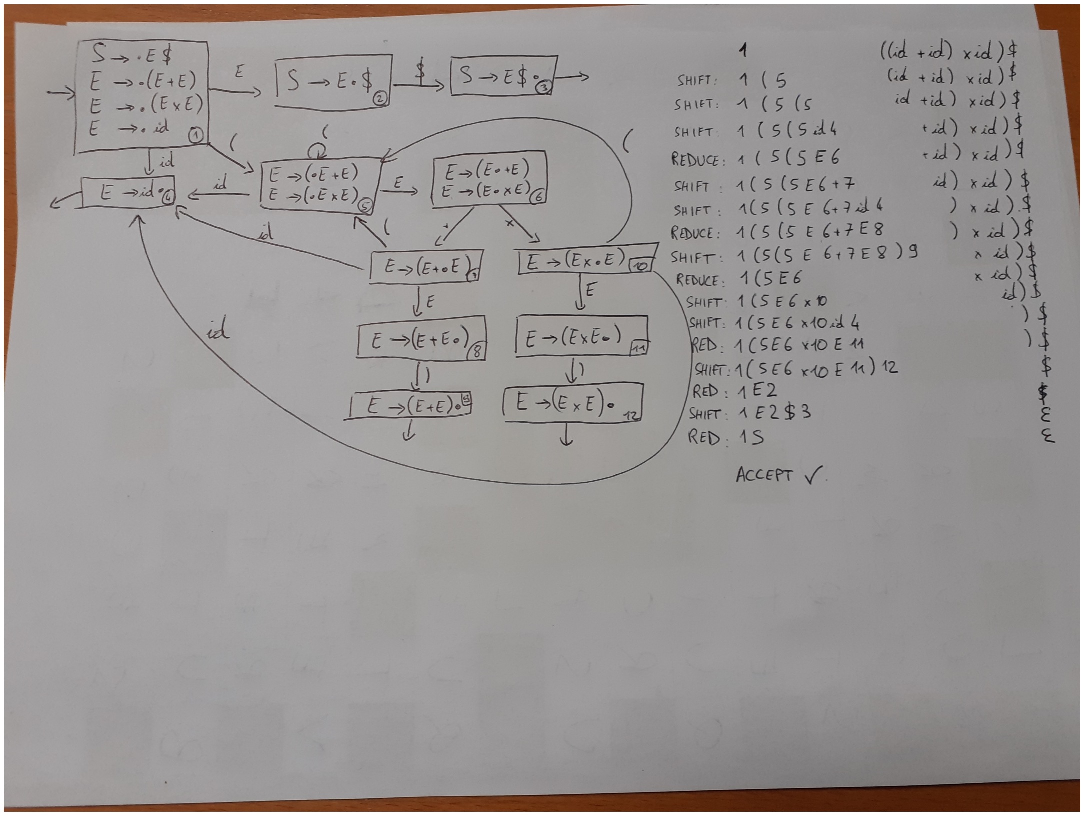
3. La pile contient $E + E$, ou bien E selon la règle utilisée par le dernier reduce (et il reste à lire $\times a$). Dans les deux cas, on a un problème car on est forcés de shift, et on ne pourra jamais réduire quelque chose contenant $E\times$.

4. 

5.

6.

7. 

8. 

9. Non, il y a un (en fait deux) conflits shift-reduce car les items $E \to T\bullet$ et $T \to T \bullet \times F$ se retrouvent dans le même état.

Exercice 6

1. Dans l'arbre de dérivation, chaque noeud a au plus d fils. Si l'arbre a une hauteur h, il y a donc au plus d^h feuilles. En lisant les feuilles de gauche à droite, on obtient α donc, $|w| \leqslant d^h$.

2. D'après la question précédente, l'arbre de dérivation de α doit avoir une hauteur au moins $N+1$. Il y a donc une branche qui utilise au moins $N+1$ non-terminaux. Comme il y a N non-terminaux, elle doit donc utiliser deux fois le même.

3. On applique la question précédente à un arbre de dérivation de α ayant le moins de noeuds possibles. En regardant les sous-mots de α dérivés à partir des noeuds trouvés à la question précédente, on trouve la factorisation $uvwxy$ recherchée. Si $vx = \varnothing$, on peut raccourcir la branche et obtenir un arbre de dérivation de α qui utilise moins de noeuds, ce qui contredit la minimalité de l'arbre choisi initialement. Ainsi $vx \neq \varnothing$.

4. Pour que uv^2wx^2y appartienne à L, v et x ne peuvent pas contenir deux lettres différentes. Ainsi, dans tous les cas, le nombre de a, b, c dans uv^2wx^2y ne peuvent être les mêmes. On obtient donc une contradiction, et L n'était pas reconnu par une grammaire.

5. L est l'intersection de $\{a^n b^n c^m, n, m > 0\}$ et de $\{a^m b^n c^n, n, m > 0\}$, qui sont reconnus par les grammaires

$$S \to aSbT \mid \varepsilon \qquad T \to cT \mid \varepsilon$$

$$S \to TbSc \mid \varepsilon \qquad T \to aT \mid \varepsilon$$

Pourtant L n'est pas reconnu par une grammaire.