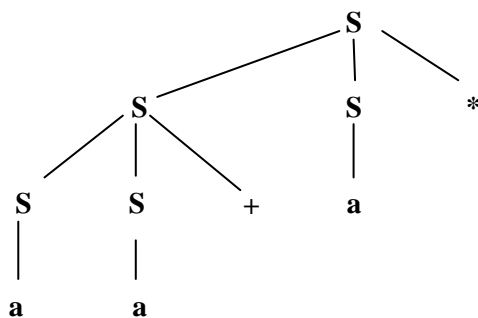


Exercice 1.

- Considérons la grammaire non contextuelle :

$$S \rightarrow SS+ \mid SS^* \mid a$$

- La dérivation à gauche de la chaîne $aa+a^*$:
 $S \Rightarrow SS^* \Rightarrow SS+S^* \Rightarrow aS+S^* \Rightarrow aa+S^* \Rightarrow aa+a^*$
- La dérivation à droite de la chaîne $aa+a^*$:
 $S \Rightarrow SS^* \Rightarrow Sa^* \Rightarrow SS+a^* \Rightarrow Sa+a^* \Rightarrow aa+a^*$
- L'arbre d'analyse de la chaîne $aa+a^*$:



- Le langage engendré par la grammaire :
 Le langage engendré par la grammaire décrit les expressions arithmétiques poste-fixées.

Exercice 2.

- Le langage engendré par les grammaires suivantes:

- $S \rightarrow +SS \mid *SS \mid a$:

Le langage engendré par cette grammaire décrit les expressions arithmétiques préfixées.

- $S \rightarrow S(S)S \mid \varepsilon$:

Le langage engendré par cette grammaire décrit les expressions bien parenthésées.

- $S \rightarrow x \mid y \mid z \mid S+S \mid S-S \mid S*S \mid S/S \mid (S)$:

Le langage engendré par cette grammaire décrit les expressions arithmétiques en trois variables x, y et z, correctement parenthésées.

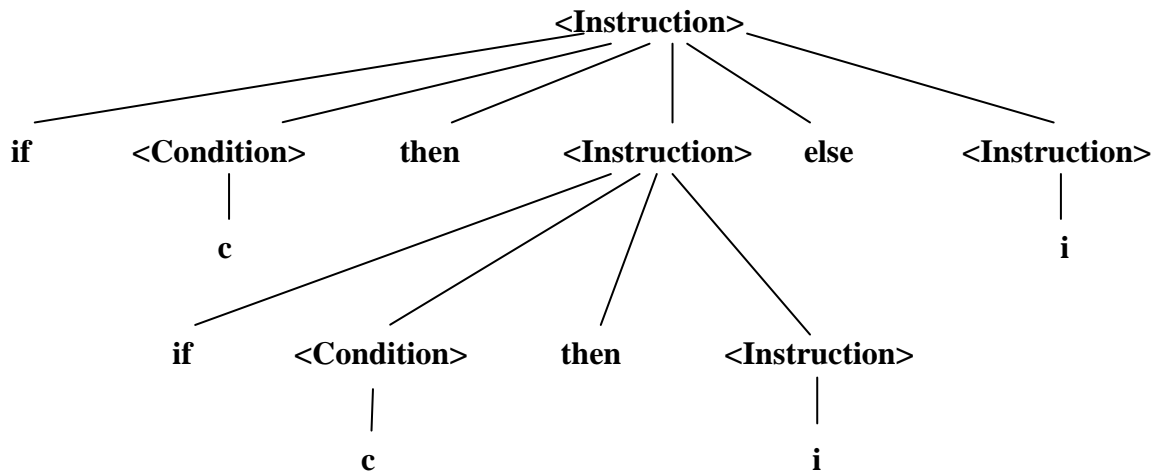
Exercice 3.

- Considérer la grammaire $G = \langle N, T, P, S \rangle$ dont les productions sont données ci-après :

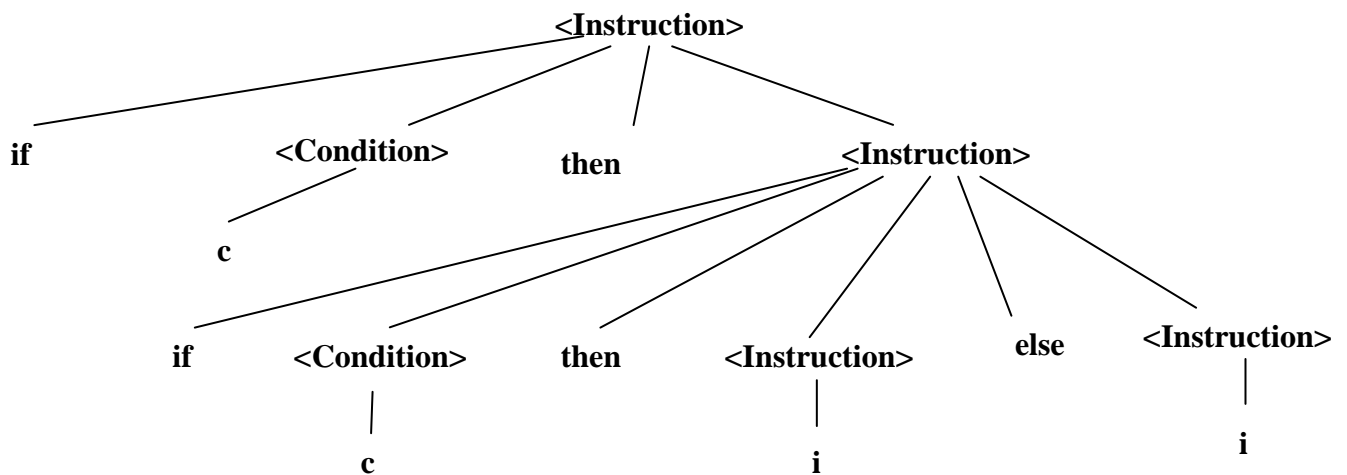
$$\begin{aligned}
 P : \langle \text{Instruction} \rangle &\rightarrow \text{if } \langle \text{Condition} \rangle \text{ then } \langle \text{Instruction} \rangle \text{ else } \langle \text{Instruction} \rangle \\
 &\mid \text{if } \langle \text{Condition} \rangle \text{ then } \langle \text{Instruction} \rangle \\
 &\mid \text{begin } \langle \text{LI} \rangle \text{ end} \\
 &\mid \text{i}
 \end{aligned}$$

$\langle \text{Condition} \rangle \rightarrow c$
 $\langle \text{LI} \rangle \rightarrow \langle \text{LI} \rangle ; i \mid i$

a) Démonstration que la grammaire G précédente est ambiguë en donnant deux arbres syntaxiques pour le fragment de programme suivant : **if c then if c then i else i**



Arbre 1



Arbre 2

La grammaire G est ambiguë car elle a deux arbres syntaxiques différents pour le fragment de programme suivant : **if c then if c then i else i**

b) Il est possible d'imbriquer les instructions **if**. Pour lever l'ambiguïté, en l'absence de parenthèses bien placées, le langage C par exemple associe toujours le **else** au **if** sans **else** le plus proche.

Exercice 4.

▪ Soit la grammaire $G = \langle \{ A, B, C \}, \{ a, b \}, P, A \rangle$

$A \rightarrow aB$

$B \rightarrow bC$

$C \rightarrow aC \mid bC \mid \epsilon$

a) La dérivation la plus à gauche de la chaîne : **abbbaab**

$A \Rightarrow aB \Rightarrow abC \Rightarrow abbC \Rightarrow abbbC \Rightarrow abbbaC \Rightarrow abbbaaC \Rightarrow abbbaabC \Rightarrow abbbaab$

b) La dérivation la plus à droite de la chaîne : **abbbaab**

$A \Rightarrow aB \Rightarrow abC \Rightarrow abbC \Rightarrow abbbC \Rightarrow abbbaC \Rightarrow abbbaaC \Rightarrow abbbaabC \Rightarrow abbbaab$

c) Le langage engendré par la grammaire :

$L(G) : ab(a|b)^*$