

TD 6 – Grammaires

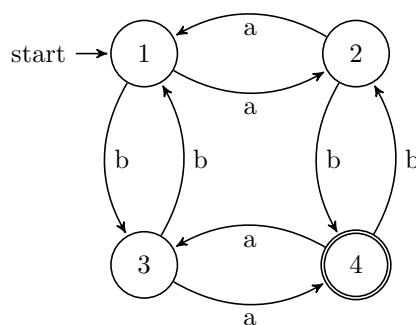
1. Soit $\Sigma = \{a, b\}$. Construisez des grammaires correspondant aux langages :

- a^*
- $(a \cup b)(a \cup b)^*$
- $(a \cup b)^*$
- $(a \cup b)^* aa (a \cup b)^*$

2. Soit $\Sigma = \{a, b\}$. Soit le langage $(a \cup b)^* aa (a \cup b)^*$.

- Construisez une grammaire régulière correspondant à ce langage.
- Déduisez l'automate à états finis de la grammaire construite.

3. Construisez la grammaire régulière engendrant le langage reconnu par l'automate suivant.



4. Construisez l'automate à pile acceptant le langage $\{a^{2n} b^n \mid n \geq 0\}$.

5. Soit la grammaire LR(1)

$E \rightarrow E \vee T \mid T$

$T \rightarrow T \wedge F \mid F$

$F \rightarrow \neg F \mid (E) \mid 0 \mid 1$

- Effectuez l'analyse ascendante de l'expression $0 \vee 1$ puis de l'expression $(0 \vee \neg 0) \wedge (\neg 1 \vee 0)$.
- Ajoutez des actions à la grammaire pour évaluer une expression.
- Évaluez les expressions précédentes.

6. Soit la grammaire LR(1)

$I \rightarrow \text{if } B \text{ then } I \mid \text{if } B \text{ then } I \text{ else } I \mid o$

$B \rightarrow b$

- Montrez que cette grammaire est ambiguë.
- Effectuez l'analyse ascendante de l'instruction *if b then if b then o else o*. À quelle étape a-t-on le choix entre une lecture et une réduction ? Donnez les arbres de dérivation dans chacun des cas.
- Construisez une grammaire non ambiguë équivalente qui associe chaque *else* avec le *then* le plus proche qui ne correspond pas déjà à un *else*.