

Compilation - Examen Final

(Durée: 01 heure et 30 minutes - Documents et appareils électroniques non autorisés)

Exercice 1 (4 pts)

Considérons la grammaire G suivante:

$$\begin{aligned} S &\rightarrow \text{begin } L \text{ end} \mid \text{if } E \text{ then } S \text{ else } S \mid \text{while } E \text{ do } S \mid \text{if } E \text{ then } S \mid \text{id} := E \mid \text{print}(E) \\ L &\rightarrow S \mid L ; S \\ E &\rightarrow \text{id} \mid \text{num} \mid E + E \mid E * E \mid (E) \end{aligned}$$

1. Écrivez la description formelle de G .
2. G est-elle ambiguë? Justifiez.
3. Éliminez la récursivité à gauche dans G .

Exercice 2 (8 pts)

Considérons la grammaire suivante qui décrit un sous ensemble d'un langage de programmation donné L :

$$\begin{aligned} P &\rightarrow \text{begin } D \text{ S end} \mid A \text{ B} \\ D &\rightarrow C ; D \mid \varepsilon \\ C &\rightarrow \text{id } L \mid \text{num } L \\ L &\rightarrow M \text{ N} \\ M &\rightarrow \text{id } M \mid \text{num } M \mid \varepsilon \\ N &\rightarrow A \text{ B} \mid B \text{ A} \mid \varepsilon \\ S &\rightarrow A \text{ S} \mid B \text{ S} \mid \varepsilon \\ A &\rightarrow \text{if } E \text{ then} \mid \text{while } E \text{ do} \\ B &\rightarrow \text{print } F \mid \text{read id} \\ E &\rightarrow F \text{ G} \mid (E) \\ F &\rightarrow \text{id} \mid \text{num} \\ G &\rightarrow + \text{ F G} \mid \varepsilon \end{aligned}$$

1. Calculez les ensembles FIRST et FOLLOW pour chaque non terminal.
2. Construisez la table prédictive LL(1) pour cette grammaire.
3. Cette grammaire est-elle LL(1)? Justifiez.

Exercice 3 (8 pts)

On voudrais concevoir, dans un analyseur lexical, un module qui reconnait un format de date personnalisé tout en respectant les restrictions ci-dessous.

Le format de la date est $YYYY - MM - DD$, où :

- $YYYY$ est une année à 4 chiffres. Elle doit être comprise entre 1000 et 2099.
- MM est un mois à 2 chiffres.
- DD est un jour à 2 chiffres.
- Si la date est en **Février** et que le jour est **29**, l'année doit être **bissextil**¹.

Exemples de dates valides:

2023-07-15 (Date valide)

1999-12-31 (Date valide)

2020-02-29 (Année bissextil, date valide)

Exemples de dates invalides:

2021-04-31 (Avril a seulement 30 jours)

2023-13-01 (Mois invalide)

2022-02-30 (Jour invalide pour février)

2023-02-29 (Pas une année bissextil)

0000-01-01 (Année hors de la plage autorisée)

1. Définissez la spécification lexicale de ce langage à l'aide d'expressions régulières.
2. Construisez un DFA qui reconnait ce langage en traitant les composants dans l'ordre correct.
3. Implémentez ce module d'analyseur lexical en *Flex*, en respectant les restrictions mentionnées ci-dessus.

¹ **Note** : Une année est **bissextil** si elle est divisible par 4, mais pas par 100. Si l'année est divisible par 4 et divisible par 100, elle n'est pas bissextil, sauf si elle est également divisible par 400.

Par exemple :

- ✓ 2024 est une année bissextil car elle est divisible par 4 mais pas par 100.
- ✓ 2000 est une année bissextil car elle est divisible par 4 et 100, mais également par 400.