

Devoir TD de compilation

Parseur

February 9, 2026

Exercice 1 : Grammaire

Écrire une grammaire pour un petit fragment (4-5 opérateurs) de langage de votre choix. Préciser les associativités/priorités. Ci-dessous un exemple de grammaire possible, issu du langage C et de ocaml:

```
 $\langle \text{expr} \rangle ::= \langle \text{INT} \rangle \mid \langle \text{LValue} \rangle \mid \langle \text{LValue} \rangle ++ \mid \langle \text{expr} \rangle \&\& \langle \text{expr} \rangle \mid \langle \text{expr} \rangle + \langle \text{expr} \rangle \mid \langle \text{expr} \rangle \langle \text{expr} \rangle$   
 $\langle \text{LValue} \rangle ::= \langle \text{IDEN} \rangle \mid \langle \text{expr} \rangle [ \langle \text{expr} \rangle ] \mid \langle \text{expr} \rangle *$ 
```

--- * ---

Exercice 2 : LR non-déterministe

Écrire l'automate LR non déterministe associé à la grammaire de l'exercice précédent.

--- * ---

Exercice 3 : LR0

Déterminez l'automate ci-dessus. Attention, les états sont très gros (beaucoup d'états non-déterminists dans un état déterministe), vous pouvez utiliser des noms pour des groupes d'états récurrents.

Résolvez les conflits que vous pouvez résoudre via l'associativité et la priorité.

L'automate shift-action obtenu est-il déterministe ? Si non, que cela signifie-t-il ?

Faites une exécution d'un mot de tokens sur cet automate.

--- * ---