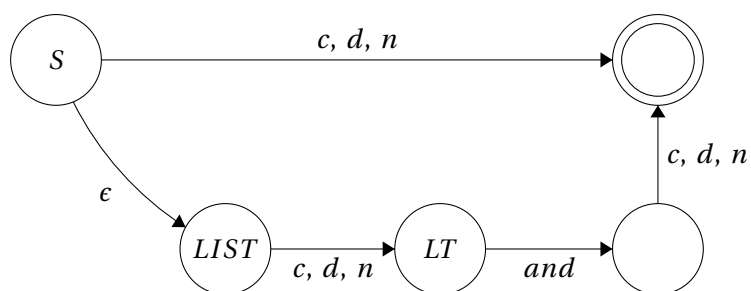


Exemple d'un langage Context Free

- $S \rightarrow "c" \mid "d" \mid "n" \mid \text{LIST}$
- $\text{LIST} \rightarrow "c" \text{LISTTAIL} \mid "d" \text{LISTTAIL} \mid "n" \text{LISTTAIL}$
- $\text{LISTTAIL} \rightarrow ", \text{LIST} \mid \text{"and"} "n" \mid \text{"and"} "c" \mid \text{"and"} "d"$

Il y a chaque fois un seul element non terminaux dans la partie droite. On remarque donc qu'il sagit d'un langage regulier.



Hierarchie de chomsky

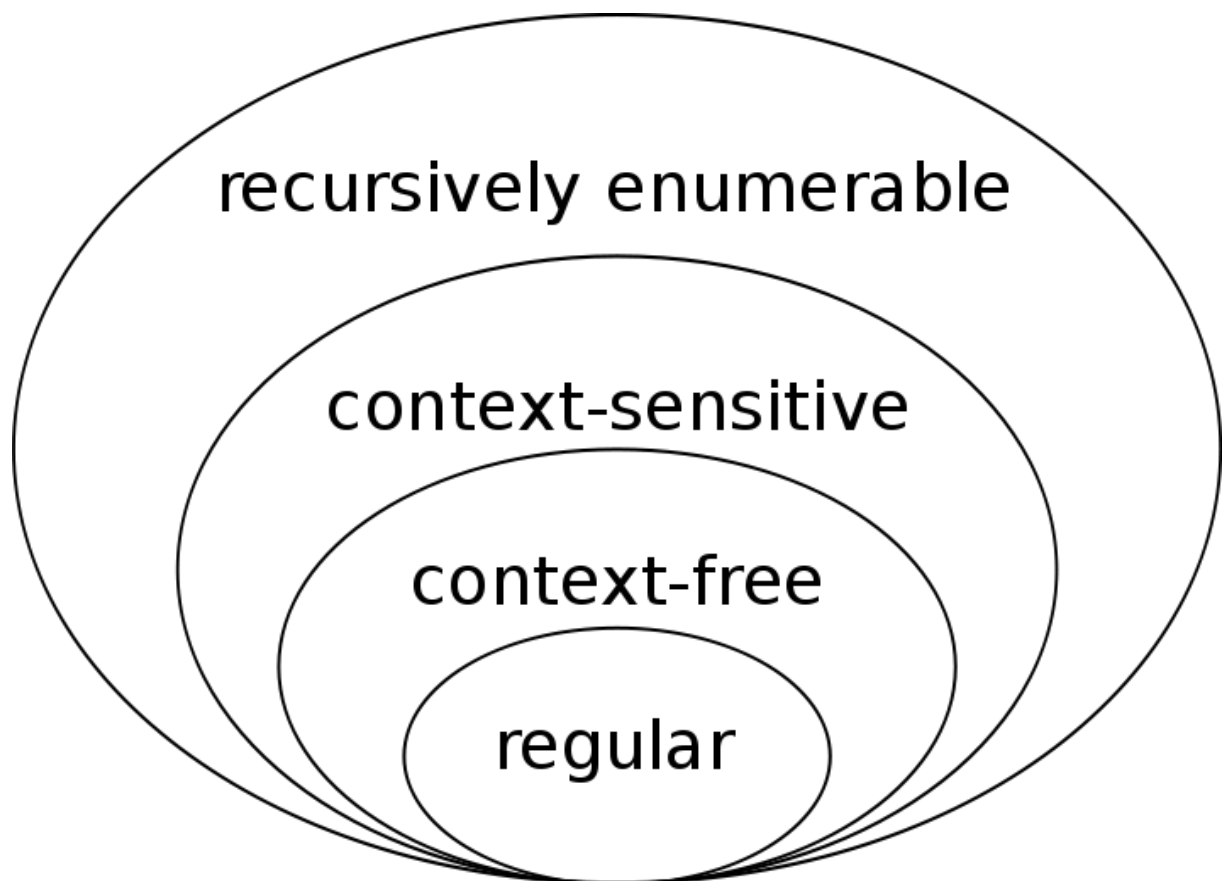


Figure 1: Hierarchie de chomsky

- Type 0: → Machine de Turing
- Type 1:
 - Monotonie
 - Contexte Sensitive $a^n b^n c^n$ → Automate borne lineaire
- Type 2:
 - Context Free $a^n b^n$ → Automate a pile
- Type 3:
 - Grammaire reguliere (langage rationnel) $a^* b^*$ → Automate
- Type 4:
 - Contexte Fini
- P: Temps polynomial sur une machine de turing deterministe.
- NP: Temps polynomial sur une machine de turing non deterministe.

Compilateur

Analyse lexical (Type 3) → Analyse syntaxique (Type 2) → Analyse Semantique (Type 1)

Simplification:

- $A \rightarrow \varepsilon$ On ajoute epsilon au context Free (Ca ne change pas son expressivite).

Grammaire ambiguë

1. $S \rightarrow S "+" S$
2. $S \rightarrow "n"$

On peut faire deux arbres de derivation different.

Dans ce cas on dit que la grammaire est ambiguë. On veut donc modifier la grammaire pour regler ce probleme.

1. $S \rightarrow S + S_n \mid S_n$
2. $S_n \rightarrow n$

Ici il n'y a plus d'ambiguite.