

Théorie des langages

①

- mot: est défini sur l'alphabet, suite fini de symboles de V .
 - Alphabet V : est un ensemble non vide constitué d'éléments (lettres, symboles)
 - V^* : c'est l'ensemble de tous les mots qu'on peut former sur l'alphabet V .
 - reflet miroir R : mot à l'envers.
 - concaténation: mettre les mots bout à bout (collés).
 - $|\text{mot}|$: longueur du mot (nbr de symboles du mot).
 - préfixe: un mot qui contient les symboles de début du mot.
 - suffixe: un mot qui contient les symboles de fin du mot.
 - langage formel: un sous ensemble de V^* .
- ### Types de langages:
- langage fini: c'est un langage où je peux énumérer les éléments.
 - langage infini: c'est un langage où je ne peux pas énumérer les éléments.

ILD WOLF

- un élément existant on le rendra pas.

- ~~une grammaire~~:

$$L, L^n = L^{n+1}$$

Type 0: aucune restriction.

Type 1: $\alpha \rightarrow \beta$ avec $|\alpha| \leq |\beta|$ sauf $S \rightarrow \epsilon$.

- α est une suite de symboles terminaux et non terminaux mais obligatoirement avoir ^{au} un non terminal.

- β est une suite non vide constituée de symboles terminaux et non terminaux. *grammaire monotone.*

Type 2: $A \rightarrow u$: A (non terminal peut être au milieu)

- Partie gauche: constituer d'un seul non terminal.

- // droite: suite de symboles terminaux et de non terminaux. *grammaire à contexte libre.*

Type 3: $A \rightarrow uB$ ou $A \rightarrow u$

- Partie gauche: constituer d'un seul non terminal.

- ~~droite~~

②

Partie droite : soit u^b soit u avec b en étant un seul non terminal et u une suite de terminaux.
~~gram~~ (le non terminal n'est pas au milieu) ~~grammaire~~ régulière.

* Les Automates :

généralisé : transition $ab, abc, +, \epsilon$.

partiellement généralisé : transition ϵ seulement.

Simple : Sans ϵ

* Automate qui accepte le Puffet Miracore : $()^R$:

* on rend l'automate avec un seul état final.

- on ajoute un Nouveau état final.

- on ajoute des ϵ -transitions des anciens état finaux vers le Nouveau état qu'on vient de créer.

- PL89
- chaque transition (u, s_i, s_j) devient (u^R, s_j, s_i)
(cela revient à inverser le sens des flèches
aussi inverser l'état initial et le final).

* Automate qui accepte l'itération: $()^*$

- ajouter des ϵ -transitions des états finaux vers
l'état initial.

* Complémentaire:

- on prend l'automate déterministe obtenu.
- on ajoute un état puis - non final - q_p -
aux états qui ont des transitions manquantes
on les raccorde.
- on inverse les états finaux et non-finaux