Chapitre 1 Grammaires NON contextuelles

- CM 14h
- TD 14h
- TP 16h

Contenu

- Langages non contextuels
  - Automates à Pile
  - Grammaires non contextuelles
  - Forme normale de Chomsky
- Compilation
  - Analyse lexicale
  - Analyse syntaxique
  - Génération de code

Ressources

- Algo, Sethi, Ullman Compilations : Principes Techniques et Outils (2000)

Lemme de l'étoile

Théorème

Si L est 1 langage (non trivial) régulier alors $\exists$ 1 entier p

$\forall w$ $\exists$ xyz tq xy^iz appartient à L
$|w| >=$ w = wyz avec $|xy| <= p$, $|y| > 0$

Idée de preuve

($q0$) → → → →
<= p

p = taille de l'automate après avoir lu p lettres de w, vous avez forcément revenir à 1 état déjat recentré.

Pour montrer qu'un 1 langage est non réguler, il suffit de montrer qu'un 1 tel p n'existe pas

Exemple

$L = \{0^n 1^n \mid n \in N\}$

Supposons qu'il existe 1 p tq $\forall$ $w < L$, $|w| >= p$, $\exists$ xyz, avec w = xyz
$xy^iz \exists L$, $\forall i$

Prenons $w = 0^p 1^p$
Par le lemme de l'étoile ∃ xyz tq w = xyz, |xy| =< p
→ xy est composé que de 0. Prenons xy^2z. Celui ci par le lemme de l'étoile appartient à L. Or xy^2 a plus de 0 que xyz et le même nombre de 1 que xyz

Par conséquent, le nombre de 0 dans $xy^2z > p$ Contradiction

Automates à pile

Rappel automates pour langages réguliers

Z = alaphabet, L inclut dans Σ*

1 automate c'est 1 tuple (Σ,Q,δ,I,F)

- Q = l'ensemble des états (fini)
- I inclut dans Q = l'ensemble des états initiaux
- F " " Q = " " " " finaux
- $\delta = Q \times \Sigma \to 2^Q$

Une exécution d'1 automate A sur 1 mot w est la séquence (q0, a0, q1, a1, q2, a2, ... , qn) teq que q0 ∈ I, w = a0, a1, ..., an la s"quence et ∀ A <= i <= n, qi ∈ δ (qi-1, ai-1)

Exemple

Si w = a0, a1, a2, a3

1 exécution est par exemple (q0, a0, q1, a1, q2, a3, q4)

Le mot w est dit accepté par A si il existe ue exécution (q0, a0, q1, a1, ..., qn, an, qn+1) de w sur 1 tellque qn+1 appartient) F

Le langage de 1, noté L(A), c'est {w appartient) Σ* | w est accepté par A}

On dit que L est *reconnaissable* si il existe 1 automate A tel que L(A) = L

1 automate A = (Σ, Q, δ, I, F) est dit deterministe ∀ q appartenant à Q, ∀ a appartenant à Σ, |δ (q, a)| = 1

Remarque

Si A est un automate déterministe, alors ∀w, ∃ 1 unique exécution de 1 sur w (preuve simple)

Théorème

Pour tout automate A, il existe 1 automate déterministe A' tel q L(A) = L(A')

De plus, A' peut être calculé en temps f(|A|), pour 1 certaine fonction, par contre |A'| peut être proportionnel à $2^{|A|}$

Corrolaire

Pour 1 automate A, on peut tester en temps O(|w|), si w appartient à L(A).

Automates à pile

Prenons L = {w | nb 0 = nb de 1}

un algorithme pour reconnaitre ce langage

```
Pour i ← 0 à |w| - 1 Faire
    si w[i] != {0, 1} Alors
        return False
    sinon si w[i] = 0 alors cpt++
    sinon cpt--

si (cpt = 0) alors
    return true
else
    return false
```

Algo 2

```
Vérifier que w est sur {0, 1}
while (i < |w|) do {
    si w[i] = 0 alors
        empiler(p, w[i])
    sinon dépiler(p)
}
si pileVide(p) alors
    return True
else return False
```

Les automates à piles sont des machine qui font à peut près le travail de l'algorithme 2

Définition formelle des automates à Pile

Si Σ 1 alphabet, on note $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

1 automate à pile est 1 tuple (Σ, Γ, Q, δ, {q0}, F)

- Σ_ε est l'alphabet fini de mots
- Γ_ε est l'alphabet fini utilisé sur la pile (valeurs manipulées par la pile)
- Q est l'ensemble fini des états
- $\delta : Q \times \Sigma\varepsilon \times \Gamma\varepsilon \rightarrow 2^{(Q \cup \Gamma_\varepsilon)}$ fonction de transition (Si V est un ensemble, 2^V ensemble des sous-ensembles de V)
- q0 appartient à Q est létat initial
- F inclut dans Q l'ensemble des états finaux