Théorie des langages : THL

CM 5

Uli Fahrenberg

EPITA Rennes

S5 2023

Aperçu

Programme du cours

1. Langages rationnels, automates finis
3. Langages algébriques, grammaires hors-contexte, automates à pile
5. Parsage LL, partie 1
6. TP 1 : `flex`
7. Parsage LL, partie 2
8. TP 2 : parsage LL
9. Parsage LR
10. TP 3, 4 : `flex` & `bison`

La dernière fois : parsage

Problème de parsage

Pour une grammaire hors contexte G, construire un algorithme de parsage qui

- pour un mot w, decide si $w \in L(G)$,
- et dans le cas $w \in L(G)$, retourne l'arbre de dérivation.

- arbre de dérivation de w $\hat{=}$ sémantique de w

Nos algorithmes de parsage devrait

- pouvoir traiter des grammaires non-ambiguës
- avoir une complexité linéaire en taille d'entrée
- lire w de gauche à droite sans retour arrière

La dernière fois : parsage LL(1)

- approche descendante
- lire le mot w de gauche à droite / Left-to-right
  - sans passer à l'arrière
- construire une dérivation gauche / Leftmost
- en accordant, à chaque pas, le premier symbole de w avec le côté droit d'une production
  - donc avec lookahead 1

- parsage LL(k) : lookahead k / « fenêtre de k lexèmes »
- peu utilisé

La dernière fois : algorithme LL(1)

0. entrée : une grammaire hors contexte $G = (N, \Sigma, P, S)$
   - si-dessous, $V = N \cup \Sigma$
   - éliminer récursion à gauche dans G ; factoriser G à gauche
1. calculer NULL
   - $\text{NULL} = \{A \in N \mid A \Rightarrow^* \varepsilon\}$
2. construire la table FIRST
   - $\text{FIRST}(A) = \{a \in \Sigma \mid \exists w \in V^* : A \Rightarrow^* aw\}$
3. construire la table FOLLOW
   - $\text{FOLLOW}(A) = \{a \in \Sigma \mid \exists B \in N, \alpha, \beta \in V^* : B \Rightarrow^* \alpha A a \beta\}$
4. construire la TABLE de parsage :
   1. pour chaque production $X \to w \quad (n)$:
      1. pour chaque $a \in \text{FIRST}(w)$: $\text{TABLE}(X, a)$ += $\{n\}$
      2. si $w \in \text{NULL}$ ou $w = \varepsilon$:
         – pour chaque $a \in \text{FOLLOW}(X)$: $\text{TABLE}(X, a)$ += $\{n\}$

La dernière fois : LL(1)

Définition (8.5)

G est LL(1) si chaque TABLE(A, a) contient au maximum une production.

La dernière fois : exemple

$$
\begin{aligned}
Z &\to XYZ && (1)\\
&\mid c && (2)\\
X &\to a && (3)\\
&\mid Y && (4)\\
Y &\to b && (5)\\
&\mid \varepsilon && (6)
\end{aligned}
$$

La dernière fois : exemple

$$
\begin{aligned}
Z &\to XYZ && (1)\\
&\mid c && (2)\\
X &\to a && (3)\\
&\mid Y && (4)\\
Y &\to b && (5)\\
&\mid \varepsilon && (6)
\end{aligned}
$$

$$\text{NULL} = \{X, Y\}$$

La dernière fois : exemple

$$
\begin{aligned}
Z &\to XYZ && (1)\\
&\mid c && (2)\\
X &\to a && (3)\\
&\mid Y && (4)\\
Y &\to b && (5)\\
&\mid \varepsilon && (6)
\end{aligned}
$$

$\text{NULL} = \{X, Y\}$

A	FIRST(A)
X	a, b
Y	b
Z	c, a, b

La dernière fois : exemple

$$
\begin{aligned}
Z &\to XYZ && (1)\\
&\mid c && (2)\\
X &\to a && (3)\\
&\mid Y && (4)\\
Y &\to b && (5)\\
&\mid \varepsilon && (6)
\end{aligned}
$$

$$\mathrm{NULL} = \{X, Y\}$$

A	FIRST(A)
X	a, b
Y	b
Z	c, a, b

A	FOLLOW(A)
X	a, b, c
Y	a, b, c
Z	

La dernière fois : exemple

$$
\begin{aligned}
Z &\to XYZ && (1)\\
&\mid c && (2)\\
X &\to a && (3)\\
&\mid Y && (4)\\
Y &\to b && (5)\\
&\mid \varepsilon && (6)
\end{aligned}
$$

$\mathrm{NULL} = \{X, Y\}$

A	FIRST(A)
X	a, b
Y	b
Z	c, a, b

A	FOLLOW(A)
X	a, b, c
Y	a, b, c
Z	

	a	b	c
X	3, 4	4	4
Y	6	5, 6	6
Z	1	1	1, 2

Exemples

Exemple (tableau)

$$
\begin{aligned}
S &\to FS && (1)\\
&\mid Q && (2)\\
&\mid \text{'('}\, S \,\text{')'}\, S && (3)\\
F &\to \text{'!'} && (4)\\
Q &\to \text{'?'} && (5)
\end{aligned}
$$

« Une session est une séquence de faits suivi par une question ; sous-sessions sont permis »

Exemple (tableau)

$$
\begin{aligned}
Z &\to S\$ && (1)\\
S &\to LQ && (2)\\
&\quad \mid \text{'('}\, S\, \text{')'}\, S && (3)\\
L &\to FL && (4)\\
&\quad \mid \varepsilon && (5)\\
F &\to \text{'!'} && (6)\\
Q &\to \text{'?'} && (7)
\end{aligned}
$$

Plus ça change, ...

Implémentation

Grammaire :

$$S \to F \tag{1}$$

$$\quad \mid \text{'('}\; S \;\text{'+'}\; F \;\text{')'} \tag{2}$$

$$F \to \text{'}\mathtt{a}\text{'} \tag{3}$$

Simple parseur en Python :

`parse.py`

Bonus : langages non-rationnels

Motifs répétitifs

- Existent-ils des langages non-rationnels ?
- Le langage $\{a^n b^n \mid n \geq 0\}$ est-il rationnel ?
- Le langage des expressions arithmétiques est-il rationnel ?

Lemme de l'étoile

1. Soit $A = (\Sigma, Q, Q_0, F, \delta)$ un automate fini avec k états.
2. Soit $x \in L(A)$ un mot de longueur $|x| = k$ (si il existe) ; écrivons $x = a_1 \dots a_k$.
3. Alors on a un calcul réussi $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots \xrightarrow{a_k} s_{k+1}$ dans A.
4. Ce calcul utilise $k+1$ états, alors un état de A a été utilisé deux fois. (Principe des tiroirs.)
5. Soient donc $i < j$ tel que $s_i = s_j$: la chaîne $s_i \rightsquigarrow s_j$ est une boucle.
6. Alors $s_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{i-1}} s_i \xrightarrow{a_j} s_{j+1} \to \cdots \to s_{k+1}$ est aussi un calcul réussi, avec étiquette $a_1 \dots a_{i-1} a_j \dots a_k$.
7. En écrivant $u = a_1 \dots a_{i-1}$, $v = a_i \dots a_{j-1}$ et $w = a_j \dots a_k$ on trouve que $L(uv^*w) \subseteq L(A)$.

Lemme de l'étoile

Théorème (4.25)

Soit L un langage rationnel. Il existe $k \geq 0$ tel que tout $x \in L$ avec longueur $|x| \geq k$ peut s'écrire $x = uvw$ avec $|uv| \leq k$, $|v| \geq 1$ et $L(uv^*w) \subseteq L$.

- aussi lemme de pompage
- note $\exists k : \forall x : \exists u, v, w$
- démonstration par quelques petites modifications de l'argument précédent

Corollaire

Théorème (rappel)

Soit L un langage rationnel. Il existe $k \geq 0$ tel que tout $x \in L$ avec longueur $|x| \geq k$ peut s'écrire $x = uvw$ avec $|uv| \leq k$, $|v| \geq 1$ et $L(uv^*w) \subseteq L$.

Corollaire

Le langage $\{a^n b^n \mid n \geq 0\}$ n'est pas rationnel.

Démonstration.

1. Supposons par l'absurde que L soit rationnel.
2. Soit k comme fourni par la lemme d'étoile.
3. Soit $x = a^k b^k$, alors $x = uvw$ avec $|uv| \leq k$ et $|v| \geq 1$.
4. Donc $u = a^i$, $v = a^j$ et $w = a^{k-i-j} b^k$ pour un $j \geq 1$.
5. On a $uw \in L(uv^*w)$ mais $uw \notin L$, contradiction !

Exercice

Théorème (rappel)

Soit L un langage rationnel. Il existe $k \geq 0$ tel que tout $x \in L$ avec longueur $|x| \geq k$ peut s'écrire $x = uvw$ avec $|uv| \leq k$, $|v| \geq 1$ et $L(uv^*w) \subseteq L$.

Montrer que le langage $\{ww \mid w \in \{a, b\}^*\}$ n'est pas rationnel.

Les automates finis sont décidables

Théorème (4.27)

Il existe un algorithme qui, pour A un automate fini, décide si $L(A)$ est vide, fini ou infini.

Démonstration.

Soit k le nombre d'états de A.

1. $L(A)$ est non-vide ssi il existe $w \in L(A)$ avec longueur $|w| < k$.
2. $L(A)$ est infini ssi il existe $w \in L(A)$ avec $k \leq |w| < 2k$.

(le reste sur tableau)


That's all Folks!