

Grundlegende Parsingalgorithmen

Tabellen-gesteuertes Parsing: LL-Parsing

Kurt Eberle

k.eberle@lingenio.de

(Viele Folien, Teile von Folien, Materialien von **Helmut Schmid's**
Parsing-Kurs WS14 Tübingen, u.a.)

24. Februar 2020

Überblick

LL(k) Parsing und Grammatiken

First- und Follow-Sets

LL(1)-Tabelle and -Tabellen-gesteuerte Erkennung

Kontrolltabellen für TD-Parsing

Problem: “Blinde” Expansion durch den Top-Down Parser

Idee: Wähle Regelanwendung $(A\alpha, aw) \mapsto (\beta\alpha, aw)$ nur falls $A \Rightarrow \beta \overset{*}{\Rightarrow} a\gamma$ gilt.

a ist das *Look-Ahead Symbol* (das nächste Eingabe-Token)

Beispiel: Grammatik und Kontroll-Tabelle

Grammatik

(mit $\Sigma = \{ d, v, a, n \}$)

$S \rightarrow NP VP$ $VP \rightarrow v NP$

$NP \rightarrow d N1$ $N1 \rightarrow a N1$

$NP \rightarrow N1$ $N1 \rightarrow n$

Kontroll-Tabelle: speichert für jedes Nicht-Terminal und jedes mögliche Look-Ahead die Menge anwendbarer Regeln.

Non-terminal	Look Ahead Symbol				
	d	a	n	v	\$
S	$S \rightarrow NP VP$	$S \rightarrow NP VP$	$S \rightarrow NP VP$		
VP				$VP \rightarrow v NP$	
NP	$NP \rightarrow d N1$	$NP \rightarrow N1$	$NP \rightarrow N1$		
N1		$N1 \rightarrow a N1$	$N1 \rightarrow n$		

Tabellen-gesteuertes Parsing mit Look-Ahead

- ⇒ LL(1) Parser
- ⇒ 'Left to right' parser with leftmost derivation and lookahead of 1 symbol
 - ▶ no multiple entries?
- ⇒ allows Top-Down parsing without backtracking
- ⇒ Grammar is called a *LL(1) Grammar*
 - ▶ Generalization ...

Überblick

LL(k) Parsing und Grammatiken

First- und Follow-Sets

LL(1)-Tabelle and -Tabellen-gesteuerte Erkennung

LL(k)-Parsing

- ▶ Extension: LL(k) parsing
- ▶ with lookahead of k symbols
- ▶ $LL(1)(\text{Grammar/language}) \subseteq LL(2) \subseteq \dots LL(k) \dots$
- ▶ Advantage: deterministic processing, linear runtime
- ▶ Disadvantage: many context-free languages are not LL(1) (and not LL(k) for any k)
- ▶ Example of a grammar which is not LL(k)

$S \rightarrow NP VP$

$NP \rightarrow d N1$

$NP \rightarrow N1$

$VP \rightarrow v NP NP$

$VP \rightarrow v NP$

$N1 \rightarrow a N1$

$N1 \rightarrow n$

Überblick

LL(k) Parsing und Grammatiken

First- und Follow-Sets

LL(1)-Tabelle and -Tabellen-gesteuerte Erkennung

First- und Follow-Sets

► First Sets

$First(\alpha)$ is the set of terminal symbols (plus ϵ), which may appear in the initial position of a string derived from α .

$a \in First(\alpha)$, iff $\alpha \xRightarrow{*} aw$

$\epsilon \in First(\alpha)$, iff $\alpha \xRightarrow{*} \epsilon$

► Follow Sets

$Follow(A)$ is the set of terminal symbols (plus \$ indicating the end of the input string), which may follow the non-terminal A in a string produced by an arbitrary derivation.

$a \in Follow(A)$, iff $S \xRightarrow{*} \alpha A a \beta$

$\$ \in Follow(A)$, iff $S \xRightarrow{*} \alpha A$

First-Set Definition

$$\textit{First} : (V \cup \Sigma)^* \longrightarrow \wp(\Sigma \cup \{\varepsilon\})$$

$$\textit{First}(\alpha) = \begin{cases} \{a \in \Sigma \mid \alpha \xRightarrow{*} a\beta, \beta \in \Sigma^*\} \cup \{\varepsilon\}, & \text{if } \alpha \xRightarrow{*} \varepsilon \\ \{a \in \Sigma \mid \alpha \xRightarrow{*} a\beta, \beta \in \Sigma^*\}, & \text{otherwise} \end{cases}$$

Berechnung der First-Sets I

Be $\alpha = X_1 \dots X_n$ a sentential form with $X_i \in (V \cup \Sigma)^*$ for all $1 \leq i \leq n, n \geq 1$

$\text{First}(\alpha)$ is the smallest set including the following:

1. if $\varepsilon \in \text{First}(X_i)$ for all $1 \leq i \leq n$,
then $\varepsilon \in \text{First}(\alpha)$
2. if $a \in \text{First}(X_1)$ and $a \neq \varepsilon$,
then $a \in \text{First}(\alpha)$
3. if $\varepsilon \in \text{First}(X_1)$, $n > 1$, $a \in \text{First}(X_2, \dots, X_n)$,
then $a \in \text{First}(\alpha)$

Berechnung der First-Sets I

Bottom of the recursion ...

1. if $\alpha = a$ with $a \in \Sigma$,
then $\text{First}(\alpha) = \{a\}$
2. if $\alpha = A$ with $A \in V$, $A \rightarrow X_1 \dots X_n \in P$,
then $\text{First}(X_1 \dots X_n)$ is subset of $\text{First}(\alpha)$

Berechnung der Follow-Sets

1. $\$ \in \text{Follow}(S)$
2. If $A \rightarrow \alpha B \beta \in P$, then $\text{First}(\beta) / \{\varepsilon\} \subseteq \text{Follow}(B)$
3. If $A \rightarrow \alpha B \in P$
or $A \rightarrow \alpha B \beta \in P$ and $\varepsilon \in \text{First}(\beta)$,
then $\text{Follow}(A) \subseteq \text{Follow}(B)$

Überblick

LL(k) Parsing und Grammatiken

First- und Follow-Sets

LL(1)-Tabelle and -Tabellen-gesteuerte Erkennung

Erstellen der LL(1)-Tabelle

1. If $a \in \text{First}(\alpha)$ and $a \neq \varepsilon$, then $A \rightarrow \alpha \in \text{Tab}[A, a]$
2. If $\varepsilon \in \text{First}(\alpha)$ and $b \in \text{Follow}(A)$, then $A \rightarrow \alpha \in \text{Tab}[A, b]$ (b could be \$)
3. If $\text{Tab}[A, a]$ is empty, then assign $\text{Tab}[A, a] = \{\text{error}\}$

Keine Mehrfacheinträge $\Rightarrow$ LL(1)-Grammatik

LL(1) Erkenner

Conf := ($S, w\$$)

Repeat

 If Conf = ($a\alpha, ar$) then

 Conf := (α, r)

 else if Conf = ($A\alpha, ar$) and $\text{Tab}[A, a] = A \rightarrow \beta$ then

 Conf := ($\beta\alpha, ar$)

 else if Conf = ($\varepsilon, \$$) then

 accept

 else

 report error