

Compiler Construction

Lecture 8: Syntactic Analysis III ($LL(k)$ Grammars)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

`noll@cs.rwth-aachen.de`

`http://www-i2.informatik.rwth-aachen.de/i2/cc10/`

Winter semester 2010/11

- 1 Repetition: Nondeterministic Top-Down Parsing
- 2 Adding Lookahead
- 3 $LL(k)$ Grammars
- 4 Follow Sets
- 5 $LL(1)$ Grammars
- 6 Computing Lookahead Sets

Approach:

- ① Given $G \in CFG_\Sigma$, construct a **nondeterministic pushdown automaton** (PDA) which accepts $L(G)$ and which additionally computes corresponding leftmost derivations (similar to the proof of “ $L(CFG_\Sigma) \subseteq L(PDA_\Sigma)$ ”)
 - input alphabet: Σ
 - pushdown alphabet: X
 - output alphabet: $[p]$
 - state set: omitted
- ② **Remove nondeterminism** by allowing **lookahead** on the input:
 $G \in LL(k)$ iff $L(G)$ recognizable by deterministic PDA with lookahead of k symbols

The Nondeterministic Top-Down Automaton I

Definition (Nondeterministic top-down parsing automaton)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_{\Sigma}$. The **nondeterministic top-down parsing automaton** of G , $NTA(G)$, is defined by the following components.

- **Input alphabet:** Σ
- **Pushdown alphabet:** X
- **Output alphabet:** $[p]$
- **Configurations:** $\Sigma^* \times X^* \times [p]^*$ (top of pushdown to the left)
- **Transitions** for $w \in \Sigma^*$, $\alpha \in X^*$, and $z \in [p]^*$:
 - expansion steps: if $\pi_i = A \rightarrow \beta$, then $(w, A\alpha, z) \vdash (w, \beta\alpha, zi)$
 - matching steps: for every $a \in \Sigma$, $(aw, a\alpha, z) \vdash (w, \alpha, z)$
- **Initial configuration** for $w \in \Sigma^*$: (w, S, ε)
- **Final configurations:** $\{\varepsilon\} \times \{\varepsilon\} \times [p]^*$

Remark: $NTA(G)$ is nondeterministic iff G contains $A \rightarrow \beta \mid \gamma$

Theorem (Correctness of $\text{NTA}(G)$)

Let $G = \langle N, \Sigma, P, S \rangle \in \text{CFG}_\Sigma$ and $\text{NTA}(G)$ as before. Then, for every $w \in \Sigma^*$ and $z \in [p]^*$,

$(w, S, \varepsilon) \vdash^* (\varepsilon, \varepsilon, z)$ iff z is a leftmost analysis of w

Proof.

$\implies$ (soundness): see exercises

$\impliedby$ (completeness): on the board



- 1 Repetition: Nondeterministic Top-Down Parsing
- 2 Adding Lookahead
- 3 $LL(k)$ Grammars
- 4 Follow Sets
- 5 $LL(1)$ Grammars
- 6 Computing Lookahead Sets

Adding Lookahead

Goal: resolve nondeterminism of $\text{NTA}(G)$ by supporting **lookahead of $k \in \mathbb{N}$ symbols** on the input

$\implies$ determination of expanding A -production by next k symbols

Definition 8.1 (first_k set)

Let $G = \langle N, \Sigma, P, S \rangle \in \text{CFG}_\Sigma$, $\alpha \in X^*$, and $k \in \mathbb{N}$. Then the **first $_k$ set** of α , $\text{first}_k(\alpha) \subseteq \Sigma^*$, is given by

$$\text{first}_k(\alpha) := \{v \in \Sigma^k \mid \text{ex. } w \in \Sigma^* \text{ such that } \alpha \Rightarrow^* vw\} \cup \{v \in \Sigma^{<k} \mid \alpha \Rightarrow^* v\}$$

Remark: $\text{first}_k(\alpha)$ is effectively computable

Example 8.2 (first_k set)

Let $G : S \rightarrow aSb \mid \varepsilon$.

- ❶ $\text{first}_1(ab) = \{a\} = \text{first}_2(a)$
- ❷ $\text{first}_3(S) = \{\varepsilon, ab, aab, aaa\}$
- ❸ $\text{first}_3(Sa) = \{a, aba, aab, aaa\}$

- 1 Repetition: Nondeterministic Top-Down Parsing
- 2 Adding Lookahead
- 3 $LL(k)$ Grammars
- 4 Follow Sets
- 5 $LL(1)$ Grammars
- 6 Computing Lookahead Sets

$LL(k)$: reading of input from left to right with k -lookahead, computing a leftmost analysis

Definition 8.3 ($LL(k)$ grammar)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_{\Sigma}$ and $k \in \mathbb{N}$. Then G has the $LL(k)$ property (notation: $G \in LL(k)$) if for all leftmost derivations of the form

$$S \Rightarrow_l^* wA\alpha \begin{cases} \Rightarrow_l w\beta\alpha \Rightarrow_l^* wx \\ \Rightarrow_l w\gamma\alpha \Rightarrow_l^* wy \end{cases}$$

such that $\beta \neq \gamma$, it follows that $\text{first}_k(x) \neq \text{first}_k(y)$ (i.e., different productions must not yield the same lookahead).

Remarks:

- If $G \in LL(k)$, then the leftmost derivation step for $wA\alpha$ in

$$S \Rightarrow_l^* wA\alpha \begin{cases} \Rightarrow_l w\beta\alpha \Rightarrow_l^* wx \\ \Rightarrow_l w\gamma\alpha \Rightarrow_l^* wy \end{cases}$$

is **determined by the next k symbols** following w .

- Corresponding **computations of $NTA(G)$** :

$$\begin{array}{llll} (wx, S, \varepsilon) \vdash^* (x, A\alpha, z) & \overset{(*)}{\vdash} & (x, \beta\alpha, zi) \vdash^* & (\varepsilon, \varepsilon, ziz') \\ (wy, S, \varepsilon) \vdash^* (y, A\alpha, z) & \overset{(*)}{\vdash} & (y, \gamma\alpha, zj) \vdash^* & (\varepsilon, \varepsilon, zjz'') \end{array}$$

where $\pi_i = A \rightarrow \beta$ and $\pi_j = A \rightarrow \gamma$

- **Deterministic decision** in $(*)$ possible if $\text{first}_k(x) \neq \text{first}_k(y)$
- **Problem:** how to **determine the A -production** from the lookahead (potentially infinitely many derivations $\beta\alpha \Rightarrow_l^* x / \gamma\alpha \Rightarrow_l^* y$)?

Lemma 8.4 (Characterization of $LL(k)$)

$G \in LL(k)$ iff for all leftmost derivations of the form

$$S \Rightarrow_l^* wA\alpha \begin{cases} \Rightarrow_l w\beta\alpha \\ \Rightarrow_l w\gamma\alpha \end{cases}$$

such that $\beta \neq \gamma$, it follows that $\text{first}_k(\beta\alpha) \cap \text{first}_k(\gamma\alpha) = \emptyset$.

Proof.

omitted □

Remarks:

- If $G \in LL(k)$, then the A -production is **determined by the lookahead sets $\text{first}_k(\beta\alpha)$** (for every $A \rightarrow \beta \in P$).
- **Problem:** still **infinitely many right contexts** α to be considered (if β [or γ] “too short”, i.e., $\text{first}_k(\beta\alpha) \neq \text{first}_k(\gamma\alpha)$).
- **Idea:** α derives to **“everything that follows A ”**

- 1 Repetition: Nondeterministic Top-Down Parsing
- 2 Adding Lookahead
- 3 $LL(k)$ Grammars
- 4 Follow Sets
- 5 $LL(1)$ Grammars
- 6 Computing Lookahead Sets

Goal: determine all possible lookaheads from production alone
(by combining all possible right contexts)

Definition 8.5 (follow_k set)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_{\Sigma}$, $A \in N$, and $k \in \mathbb{N}$. Then the follow_k set of A , $\text{follow}_k(A) \subseteq \Sigma^*$, is given by

$$\text{follow}_k(A) := \{v \in \text{first}_k(\alpha) \mid \text{ex. } w \in \Sigma^*, \alpha \in X^* \text{ such that } S \Rightarrow_l^* wA\alpha\}.$$

- 1 Repetition: Nondeterministic Top-Down Parsing
- 2 Adding Lookahead
- 3 $LL(k)$ Grammars
- 4 Follow Sets
- 5 $LL(1)$ Grammars**
- 6 Computing Lookahead Sets

Motivation:

- $k = 1$ sufficient to resolve nondeterminism in “most” practical applications
- Implementation of $LL(k)$ parsers for $k > 1$ rather involved (cf. **ANTLR** [ANother Tool for Language Recognition; formerly PCCTS] at [http://www.antlr.org/](http://wwwantlr.org/))

Abbreviations: $\text{fi} := \text{first}_1$, $\text{fo} := \text{follow}_1$, $\Sigma_\varepsilon := \Sigma \cup \{\varepsilon\}$

Corollary 8.6

- 1 For every $\alpha \in X^*$,
$$\text{fi}(\alpha) = \{a \in \Sigma \mid \text{ex. } w \in \Sigma^* : \alpha \Rightarrow^* aw\} \cup \{\varepsilon \mid \alpha \Rightarrow^* \varepsilon\} \subseteq \Sigma_\varepsilon$$
- 2 For every $A \in N$,
$$\text{fo}(A) = \{x \in \text{fi}(\alpha) \mid \text{ex. } w \in \Sigma^*, \alpha \in X^* : S \Rightarrow_l^* wA\alpha\} \subseteq \Sigma_\varepsilon.$$

Definition 8.7 (Lookahead set)

Given $\pi = A \rightarrow \beta \in P$,

$$\text{la}(\pi) := \text{fi}(\beta \cdot \text{fo}(A)) \subseteq \Sigma_\varepsilon$$

is called the **lookahead set** of π (where $\text{fi}(\Gamma) := \bigcup_{\gamma \in \Gamma} \text{fi}(\gamma)$).

Corollary 8.8

❶ For all $a \in \Sigma$,

$$a \in \text{la}(A \rightarrow \beta) \text{ iff } a \in \text{fi}(\beta) \text{ or } (\beta \Rightarrow^* \varepsilon \text{ and } a \in \text{fo}(A))$$

❷ $\varepsilon \in \text{la}(A \rightarrow \beta)$ iff $\beta \Rightarrow^* \varepsilon$ and $\varepsilon \in \text{fo}(A)$

Characterization of $LL(1)$

Theorem 8.9 (Characterization of $LL(1)$)

$G \in LL(1)$ iff for all pairs of rules $A \rightarrow \beta \mid \gamma \in P$ (where $\beta \neq \gamma$):

$$\text{la}(A \rightarrow \beta) \cap \text{la}(A \rightarrow \gamma) = \emptyset.$$

Proof.

on the board



Remark: the above theorem generally does not hold if $k > 1$
(cf. exercises)

- 1 Repetition: Nondeterministic Top-Down Parsing
- 2 Adding Lookahead
- 3 $LL(k)$ Grammars
- 4 Follow Sets
- 5 $LL(1)$ Grammars
- 6 Computing Lookahead Sets

Computing Lookahead Sets I

(see Waite/Goos: *Compiler Construction*, p. 164f)

Lemma 8.10 (Computation of fi/fo)

The sets $\text{fi}(\alpha) \subseteq \Sigma_\varepsilon$ (for $\alpha \in X^$) and $\text{fo}(A) \subseteq \Sigma_\varepsilon$ (for $A \in N$) are the least sets such that:*

- ❶ $\text{fi}(Y)$ for $Y \in X$:
 - $Y \in \Sigma \implies \text{fi}(Y) = \{Y\}$
 - $Y \rightarrow A_1 \dots A_k Z \alpha \in P, k \in \mathbb{N}, Z \in X, \varepsilon \in \text{fi}(A_1) \cap \dots \cap \text{fi}(A_k), a \in \text{fi}(Z) \implies a \in \text{fi}(Y)$
 - $Y \rightarrow A_1 \dots A_k \in P, k \in \mathbb{N}, \varepsilon \in \text{fi}(A_1) \cap \dots \cap \text{fi}(A_k) \implies \varepsilon \in \text{fi}(Y)$
- ❷ $\text{fi}(Y_1 \dots Y_n)$ for $n \in \mathbb{N}, Y_i \in X$:
 - $\varepsilon \in \text{fi}(Y_1 \dots Y_{k-1}), a \in \text{fi}(Y_k), k \in [n] \implies a \in \text{fi}(Y_1 \dots Y_n)$
 - $\varepsilon \in \text{fi}(Y_1) \cap \dots \cap \text{fi}(Y_n) \implies \varepsilon \in \text{fi}(Y_1 \dots Y_n)$
- ❸ $\text{fo}(A)$ for $A \in N$:
 - $\varepsilon \in \text{fo}(S)$
 - $A \rightarrow \alpha B \beta \in P, a \in \text{fi}(\beta) \implies a \in \text{fo}(B)$
 - $A \rightarrow \alpha B \beta \in P, \varepsilon \in \text{fi}(\beta), x \in \text{fo}(A) \implies x \in \text{fo}(B)$

Computing Lookahead Sets II

Corollary 8.11

- ① $A \rightarrow a\alpha \in P \implies a \in \text{fi}(A)$
- ② $A \rightarrow B\alpha \in P, a \in \text{fi}(B) \implies a \in \text{fi}(A)$
- ③ $A \rightarrow \varepsilon \in P \implies \varepsilon \in \text{fi}(A)$
- ④ $\text{fi}(\varepsilon) = \{\varepsilon\}$
- ⑤ $a \in \text{fi}(A) \implies a \in \text{fi}(A\alpha)$
- ⑥ $A \rightarrow \alpha B \in P, x \in \text{fo}(A) \implies x \in \text{fo}(B)$

Example 8.12

Grammar for
arithmetic
expressions

(cf. Example 7.3):

$$\begin{array}{l} G_{AE} : \quad E \rightarrow E+T \mid T \\ \quad \quad T \rightarrow T * F \mid F \\ \quad \quad F \rightarrow (E) \mid a \mid b \end{array}$$

- $F \rightarrow a \in P \implies a \in \text{fi}(F)$
- $T \rightarrow F \in P, a \in \text{fi}(F) \implies a \in \text{fi}(T)$
- $a \in \text{fi}(T)$
 $\implies \text{la}(T \rightarrow T * F) = \text{fi}(T * F \cdot \text{fo}(T)) \ni a$
- $a \in \text{fi}(F)$
 $\implies \text{la}(T \rightarrow F) = \text{fi}(F \cdot \text{fo}(T)) \ni a$
- $\implies a \in \text{la}(T \rightarrow T * F) \cap \text{la}(T \rightarrow F) \neq \emptyset$
- $\implies G_{AE} \notin LL(1)$