Compiler Construction

Lecture 6: Syntax Analysis II ($LL(k)$ Grammars)

Winter Semester 2018/19

Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University

`https://moves.rwth-aachen.de/teaching/ws-1819/cc/`

Software Modeling and Verification Chair

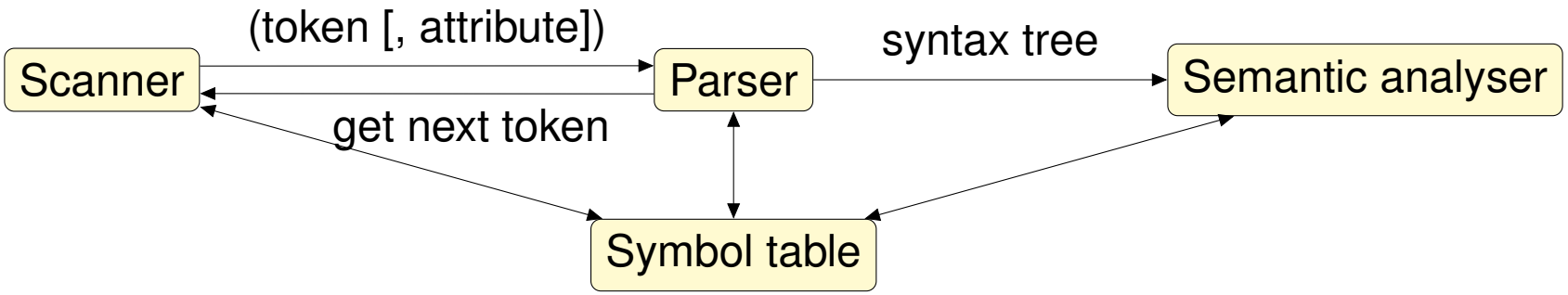
RWTH AACHEN UNIVERSITY

Syntax Analysis

Definition

The goal of syntax analysis is to determine the syntactic structure of a program, given by a token sequence, according to a context-free grammar.

The corresponding program is called a parser:

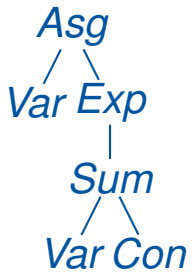


Example: $\ldots\,␣\texttt{x1}␣\texttt{:=y2+}␣\texttt{1}␣\texttt{;}␣\ldots$

$\downarrow$ Scanner

$\ldots (\text{id}, p_1)(\text{gets},)(\text{id}, p_2)(\text{plus},)(\text{int}, 1)(\text{sem},)\ldots$

Parser $\longrightarrow$



Parsing Context-Free Languages

Goal: exploit the special syntactic structures as present in programming languages (usually: no ambiguities) to devise parsing methods which are based on deterministic pushdown automata with linear space and time complexity

Two approaches:

Top-down parsing: construction of syntax tree from the root towards the leaves, representation as leftmost derivation

Bottom-up parsing: construction of syntax tree from the leaves towards the root, representation as (reversed) rightmost derivation

Top-Down Parsing

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: $X \ (= N \cup \Sigma)$
   - output alphabet: $[p]$
   - state set: not required
2. Remove nondeterminism by supporting lookahead on the input:
   $G \in LL(k)$ iff $L(G)$ recognisable by deterministic PDA with lookahead of k symbols

The Nondeterministic Top-Down Automaton

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 $\pi_i = A \to \beta$, 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 $A \to \beta \mid \gamma$

Correctness of $NTA(G)$

Theorem 6.1 (Correctness of $NTA(G)$)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_\Sigma$ and $NTA(G)$ as in Definition 5.13. Then, for every $w \in \Sigma^*$ and $z \in [p]^*$,

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

Proof.

"$\Longrightarrow$" (soundness): see exercises

"$\Longleftarrow$" (completeness): on the board

□

Adding Lookahead

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

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

Definition 6.2 (first_k set)

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

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

Remark: $\mathrm{first}_k(\alpha)$ is effectively computable. If $\alpha \in \Sigma^*$, then $|\mathrm{first}_k(\alpha)| = 1$.

Example 6.3 (first_k set)

For $G : S \to aSb \mid \varepsilon$:

$$\mathrm{first}_1(ab) = \{a\} = \mathrm{first}_2(a)$$
$$\mathrm{first}_3(S) = \{\varepsilon, ab, aab, aaa\}$$
$$\mathrm{first}_3(Sa) = \{a, aba, aab, aaa\}$$

$LL(k)$ Grammars I

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

Definition 6.4 ($LL(k)$ 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 $\mathrm{first}_k(x) \neq \mathrm{first}_k(y)$.

Thus: different productions must not yield the same lookahead.

$LL(k)$ Grammars II

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 $NTA(G)$:

$$(wx, S, \varepsilon) \vdash^{|w|+|z|} (x, A\alpha, z) \overset{(*)}{\vdash} (x, \beta\alpha, zi) \vdash^{|x|+|z'|} (\varepsilon, \varepsilon, ziz')$$
$$(wy, S, \varepsilon) \vdash^{|w|+|z|} (y, A\alpha, z) \overset{(*)}{\vdash} (y, \gamma\alpha, zj) \vdash^{|y|+|z''|} (\varepsilon, \varepsilon, zjz'')$$

  where $\pi_i = A \to \beta$ and $\pi_j = A \to \gamma$
- 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 $\beta\alpha \Rightarrow_l^* x$ and $\gamma\alpha \Rightarrow_l^* y$)?

$LL(k)$ Grammars III

Lemma 6.5 (Characterisation 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 $\mathrm{first}_k(\beta\alpha) \cap \mathrm{first}_k(\gamma\alpha) = \emptyset$.

Proof.

omitted □

Remarks:

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

Follow Sets

The follow_k Sets

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

Definition 6.6 (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, $\mathrm{follow}_k(A) \subseteq \Sigma^*$, is given by

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

$LL(1)$ Grammars

The Case $k = 1$

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/`)

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

Corollary 6.7

1. *For every* $\alpha \in X^*$,

$$\mathrm{fi}(\alpha) = \{a \in \Sigma \mid \text{ex. } w \in \Sigma^* : \alpha \Rightarrow^* aw\} \cup \{\varepsilon \mid \alpha \Rightarrow^* \varepsilon\} \subseteq \Sigma_\varepsilon$$

2. *For every* $A \in N$,

$$\mathrm{fo}(A) = \{x \in \mathrm{fi}(\alpha) \mid \text{ex. } w \in \Sigma^*, \alpha \in X^* : S \Rightarrow_l^* wA\alpha\} \subseteq \Sigma_\varepsilon.$$

Lookahead Sets

Definition 6.8 (Lookahead set)

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

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

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

Corollary 6.9

1. *For all* $a \in \Sigma$,
$$a \in \mathrm{la}(A \to \beta) \quad \textit{iff} \quad a \in \mathrm{fi}(\beta) \textit{ or } (\beta \Rightarrow^* \varepsilon \textit{ and } a \in \mathrm{fo}(A))$$
2. $\varepsilon \in \mathrm{la}(A \to \beta) \quad \textit{iff} \quad \beta \Rightarrow^* \varepsilon \textit{ and } \varepsilon \in \mathrm{fo}(A)$

Characterisation of $LL(1)$

Theorem 6.10 (Characterisation of $LL(1)$)

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

$$\mathrm{la}(A \to \beta) \cap \mathrm{la}(A \to \gamma) = \emptyset.$$

Proof.

on the board □

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