



Compiler Construction

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

Summer Semester 2017

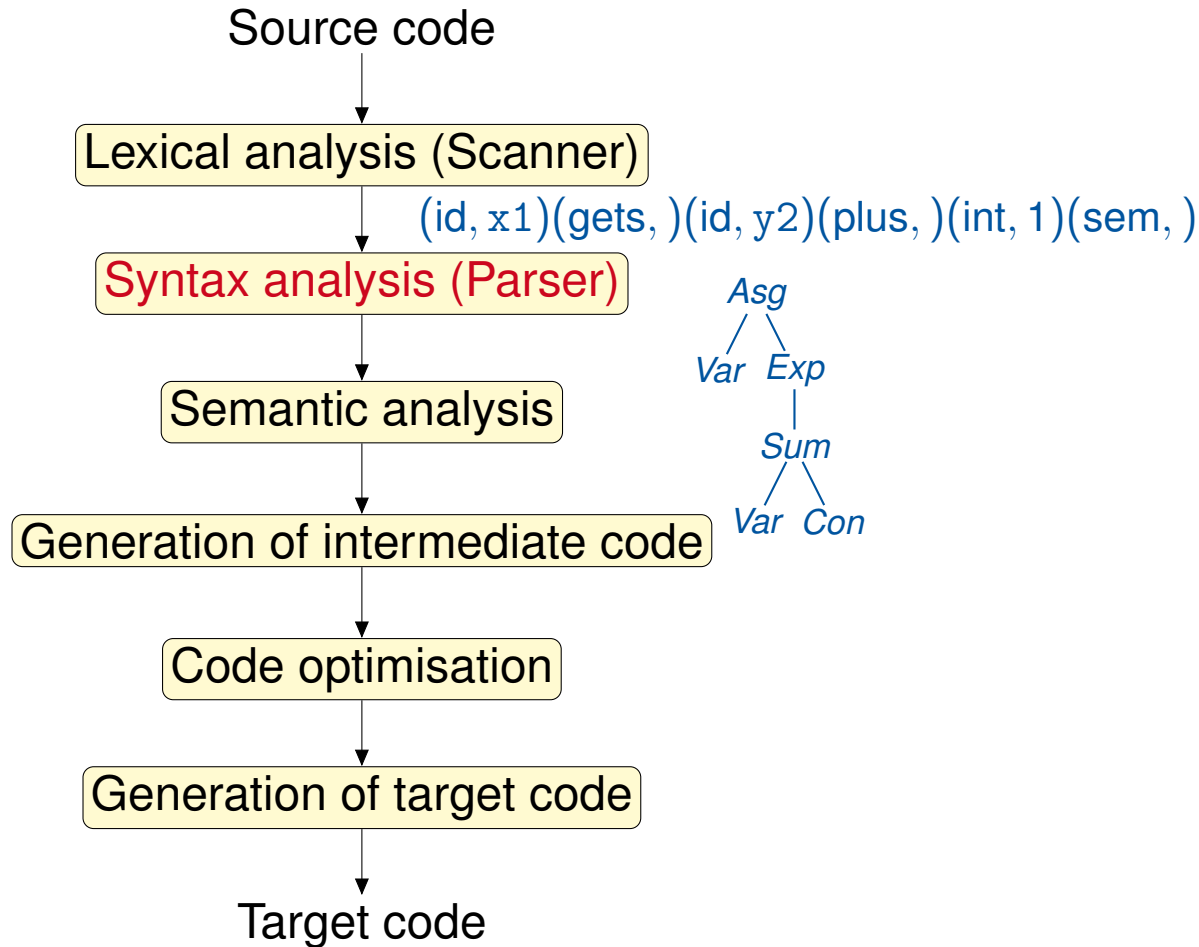
Thomas Noll

Software Modeling and Verification Group

RWTH Aachen University

<https://moves.rwth-aachen.de/teaching/ss-17/cc/>

Conceptual Structure of a Compiler



context-free grammars/
pushdown automata

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
 - 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 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 $\pi_j = A \rightarrow \beta$, 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 $A \rightarrow \beta \mid \gamma$

Correctness of $NTA(G)$

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 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



Adding Lookahead

Adding Lookahead

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

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

Definition 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 , $\text{first}_k(\alpha) \subseteq \Sigma^*$, is given by

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

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

Example 6.3 (first_k set)

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

$$\begin{aligned} \text{first}_1(ab) &= \{a\} = \text{first}_2(a) \\ \text{first}_3(S) &= \{\varepsilon, ab, aab, aaa\} \\ \text{first}_3(Sa) &= \{a, aba, aab, aaa\} \end{aligned}$$

LL(k) Grammars

LL(k) Grammars I

LL(k): reading of input from **L**eft to right with **k**-lookahead, computing a **L**eftmost 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

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

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).

LL(k) Grammars II

Remarks:

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

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

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

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

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

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 $\beta\alpha \Rightarrow_i^* x$ and $\gamma\alpha \Rightarrow_i^* y$)?

LL(k) Grammars

LL(k) Grammars III

Lemma 6.5 (Characterisation of LL(k))

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

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

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

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

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: $fi := first_1$, $fo := follow_1$, $\Sigma_\epsilon := \Sigma \cup \{\epsilon\}$

Corollary 6.7

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

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

2. For every $A \in N$,

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

LL(1) Grammars

Lookahead Sets

Definition 6.8 (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 6.9

1. For all $a \in \Sigma$,

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

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

LL(1) Grammars

Characterisation of LL(1)

Theorem 6.10 (Characterisation 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)