Universitetet i Oslo
Institutt for Informatikk
PMA
Martin Steffen

INF 5110: Compiler construction

Spring 2018

Handout 4

15. 1. 2018

Handout 4: Parsing

Issued: 15. 1. 2018

For reference, to follow the slides, the handout includes some grammars we repeteadly used for illustration. These are various versions of the context-free grammar for expressions. The first version is the "obvious" one. Also some definitions are added.

Some definitions

Definition 1 (First set) Given a grammar G and a non-terminal A. The *first-set* of A, written $First_G(A)$ is defined as

$$First_G(A) = \{a \mid A \Rightarrow_G^* a\alpha, \quad a \in \Sigma_T\} + \{\epsilon \mid A \Rightarrow_G^* \epsilon\} \ . \tag{1}$$

Definition 2 (Nullable) Given a grammar G. A non-terminal $A \in \Sigma_N$ is *nullable*, if $A \Rightarrow^* \epsilon$.

Definition 3 (First set of a symbol) Given a grammar G and grammar symbol X. The *first-set* of X, written $First(X)$, is defined as follows:

1. If $X \in \Sigma_T + \{\boldsymbol{\epsilon}\}$, then $First(X) = \{X\}$.
2. If $X \in \Sigma_N$: For each production
$$X \to X_1 X_2 \ldots X_n$$
   (a) $First(X)$ contains $First(X_1) \setminus \{\boldsymbol{\epsilon}\}$
   (b) If, for some $i < n$, *all* $First(X_1), \ldots, First(X_i)$ contain $\boldsymbol{\epsilon}$, then $First(X)$ contains $First(X_{i+1}) \setminus \{\boldsymbol{\epsilon}\}$.
   (c) If all $First(X_1), \ldots, First(X_n)$ contain $\boldsymbol{\epsilon}$, then $First(X)$ contains $\{\boldsymbol{\epsilon}\}$.

Definition 4 (First set of a word) Given a grammar G and word α. The *first-set* of

$$\alpha = X_1 \ldots X_n \ ,$$

written $First(\alpha)$ is defined inductively as follows:

1. $First(\alpha)$ contains $First(X_1) \setminus \{\boldsymbol{\epsilon}\}$
2. for each $i = 2, \ldots n$, if $First(X_k)$ contains $\boldsymbol{\epsilon}$ for *all* $k = 1, \ldots, i-1$, then $First(\alpha)$ contains $First(X_i) \setminus \{\boldsymbol{\epsilon}\}$
3. If all $First(X_1), \ldots, First(X_n)$ contain $\boldsymbol{\epsilon}$, then $First(X)$ contains $\{\boldsymbol{\epsilon}\}$.

Definition 5 (Follow set) Given a grammar G with start symbol S, and a non-terminal A. The *follow-set* of A, written $Follow_G(A)$, is

$$Follow_G(A) = \{a \mid S\,\$ \Rightarrow_G^* \alpha_1 A a \alpha_2, \quad a \in \Sigma_T + \{\,\$\,\}\} \ . \tag{2}$$

Definition 6 (Follow set of a non-terminal) Given a grammar G and nonterminal A. The *Follow-set* of A, written $Follow(A)$ is defined as follows:

1. If A is the start symbol, then $Follow(A)$ contains $\$$.
2. If there is a production $B \to \alpha A \beta$, then $Follow(A)$ contains $First(\beta) \setminus \{\epsilon\}$.
3. If there is a production $B \to \alpha A \beta$ such that $\epsilon \in First(\beta)$, then $Follow(A)$ contains $Follow(B)$.

Lemma 7 (LL(1) (without nullable symbols)) A reduced context-free grammar without nullable non-terminals is an LL(1)-grammar iff for all non-terminals A and for all pairs of productions $A \to \alpha_1$ and $A \to \alpha_2$ with $\alpha_1 \neq \alpha_2$:

$$First_1(\alpha_1) \cap First_1(\alpha_2) = \varnothing \ .$$

Definition 8 (Handle) Assume $S \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha \beta w$. A production $A \to \beta$ at position k following α is a *handle of* $\alpha\beta w$. We write $\langle A \to \beta, k \rangle$ for such a handle.

Some grammars

$$\begin{array}{rcl} exp & \to & exp\ \ op\ \ exp \mid \mathbf{(}\ exp\ \mathbf{)} \mid \mathbf{number} \\ op & \to & \mathbf{+} \mid \mathbf{-} \mid \mathbf{*} \end{array} \tag{3}$$

The second version is the slightly less obvious one, used to take care of precedences (like multiplication over addition). The fact that in this grammar we don't just stipulate "multiplication binds stronger than addition and substraction" on top of the obvious grammar rules, but encode that in the productions without resorting to addition conditions on top of the grammar, makes the grammar slightly less readable.

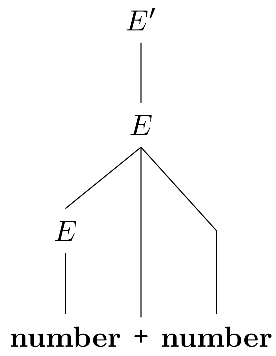
$$\begin{array}{rcl} exp & \to & exp\ \ addop\ \ term \mid term \\ addop & \to & \mathbf{+} \mid \mathbf{-} \\ term & \to & term\ \ mulop\ \ factor \mid factor \\ mulop & \to & \mathbf{*} \\ factor & \to & \mathbf{(}\ exp\ \mathbf{)} \mid \mathbf{number} \end{array} \tag{4}$$

Grammars to illustrate bottom-up

The following 2 (artificiald) grammar (and the parse-tree) is used to illustrate the bottom-up parsing process.

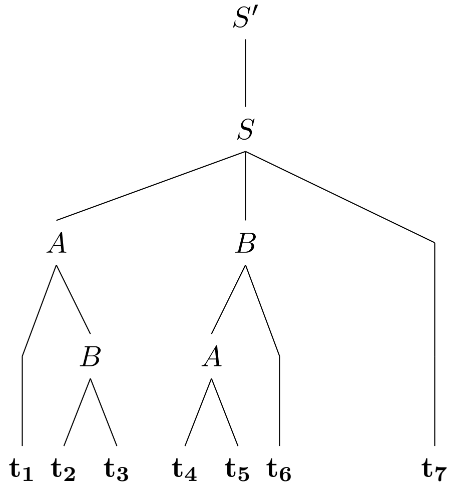
Simplistic addition expressions

$$
\begin{array}{rcl}
E' & \rightarrow & E \\
E & \rightarrow & E\,\mathbf{+}\,\mathbf{number} \;\mid\; \mathbf{number}
\end{array}
$$



Artificial grammar

$$
\begin{array}{rcl}
S' & \rightarrow & S \\
S & \rightarrow & AB\mathbf{t_7} \;\mid\; \ldots \\
A & \rightarrow & \mathbf{t_4 t_5} \;\mid\; \mathbf{t_1} B \;\mid\; \ldots \\
B & \rightarrow & \mathbf{t_2 t_3} \;\mid\; A\mathbf{t_6} \;\mid\; \ldots
\end{array}
$$



Grammars to illustrate LR(0) construction

Another example used in the lecture is the "simplistic additions" (see before).

Parentheses

$$
\begin{array}{rcl}
S' & \rightarrow & S \\
S & \rightarrow & \mathbf{(}\,S\,\mathbf{)}\,S \;\mid\; \boldsymbol{\epsilon}
\end{array}
$$

$$\begin{array}{rcl}
S' & \to & \mathbf{.}S \\
S' & \to & S\mathbf{.} \\
S & \to & \mathbf{.}\,\mathbf{(}\,S\,\mathbf{)}\,S \\
S & \to & \mathbf{(}\,\mathbf{.}S\,\mathbf{)}\,S \\
S & \to & \mathbf{(}\,S\mathbf{.}\,\mathbf{)}\,S \\
S & \to & \mathbf{(}\,S\,\mathbf{)}\,\mathbf{.}S \\
S & \to & \mathbf{(}\,S\,\mathbf{)}\,S\mathbf{.} \\
S & \to & \mathbf{.}
\end{array}$$

Simplistic addition

$$\begin{array}{rcl}
E' & \to & \mathbf{.}E \\
E' & \to & E\mathbf{.} \\
E & \to & \mathbf{.}E\,\mathbf{+}\,\mathbf{number} \\
E & \to & E\mathbf{.}\,\mathbf{+}\,\mathbf{number} \\
E & \to & E\,\mathbf{+}\,\mathbf{.number} \\
E & \to & E\,\mathbf{+}\,\mathbf{number.} \\
E & \to & \mathbf{.number} \\
E & \to & \mathbf{number.}
\end{array}$$