# Report for ordinary compiler exam at DIKU - Taskset 4

Jonas Brunsgaard (msn368

January 18, 2013

## 1 Top-Down Parsing for if-then-else

In this assignment we will work our way forward to determine whether or not a grammar is LL1. This can be determined by deriving the Look-ahead sets for the grammar.

To derive the Look-ahead sets we need to eliminate any left recursion, left-factorize the grammar, determine nullabilility, first sets and follow sets.

The procedure used in this assignment are described in the book section 2.12, With the one exception that we use Look-ahead sets instead of a parse table to determine if the grammar is LL1. Look ahead sets are described in the slides from the parser lecture.

### 1.1 Left-factorization of the grammar

Below is the new grammar after left-factorization.

$$
\begin{array}{lcl}
S & \rightarrow & \text{if}\, BSS_* \\
S & \rightarrow & \text{return}\,\text{NUM;} \\
S_* & \rightarrow & \varepsilon \\
S_* & \rightarrow & \text{else}\, S \\
B & \rightarrow & (\,\text{NUM}\,)
\end{array}
$$

### 1.2 Nullablility and First sets

First we determine if terminals and nonterminals are *nullable*.

| Right-hand side | Init | First Iter | Sec Iter |
|---|---|---|---|
| if $BSS*$ | *false* | *false* | *false* |
| return NUM; | *false* | *false* | *false* |
| $\varepsilon$ | *false* | *true* | *true* |
| else $S$ | *false* | *false* | *false* |
| ( NUM ) | *false* | *false* | *false* |
| Nonterminal | | | |
| $S$ | *false* | *false* | *false* |
| $S_*$ | *false* | *true* | *true* |
| $B$ | *false* | *false* | *false* |

Then we use fixed point iteration to determine the $FIRST$-sets:

| Right-hand side | Init | First Iter | Sec Iter |
|---|---|---|---|
| if $BSS_*$ | $\emptyset$ | {if} | {if} |
| return NUM; | $\emptyset$ | {return} | {return} |
| $\varepsilon$ | $\emptyset$ | $\emptyset$ | $\emptyset$ |
| else $S$ | $\emptyset$ | {else} | {else} |
| ( NUM ) | $\emptyset$ | {(} | {(} |
| Nonterminal | | | |
| $S$ | $\emptyset$ | {if,return} | {if,return} |
| $S_*$ | $\emptyset$ | {else} | {else} |
| $B$ | $\emptyset$ | {(} | {(} |

## 1.3 Calculate Follow sets for all nonterminals

By following the procedure found on page 59 in the book, we derive the table below. To handle the end-of-string condition we add

$$S' \to S\$$$

to the production

| Production | Constraints |
|---|---|
| $S' \to S\$$ | $\{\$\} \subseteq FOLLOW(S)$ |
| $S \to$ if $BSS_*$ | $\{\text{return}, \text{if}\} \subseteq FOLLOW(B)$,<br>$FOLLOW(S) \subseteq FOLLOW(S_*)$,<br>$\{\text{else}\} \subseteq FOLLOW(S)$ |
| $S \to$ return NUM; | |
| $S_* \to \varepsilon$ | |
| $S_* \to$ else $S$ | $FOLLOW(S_*) \subseteq FOLLOW(S)$ |
| $B \to$ ( NUM ) | |

We first use the constraints $\{\$\} \subseteq FOLLOW(S)$ and constraints of the form $FIRST(\dots) \subseteq FOLLOW(\dots)$ to get the initial sets.

$$
\begin{aligned}
FOLLOW(S) &\subseteq \{\text{else}, \$\} \\
FOLLOW(S_*) &\subseteq \{\emptyset\} \\
FOLLOW(B) &\subseteq \{\text{if}, \text{return}\}
\end{aligned}
$$

and then use the constrains on the form $FOLLOW(\dots) \subseteq FOLLOW(\dots)$:

$$
\begin{aligned}
FOLLOW(S) &\subseteq \{\text{else}, \$\} \\
FOLLOW(S_*) &\subseteq \{\text{else}, \$\} \\
FOLLOW(B) &\subseteq \{\text{if}, \text{return}\}
\end{aligned}
$$

### 1.4 Look-aheads sets

From the lecture slides the look ahead set is defined as

$$
la(X \to \alpha) = \begin{cases} FIRST(\alpha) \cup FOLLOW(X) & \text{, if } NULLABLE(\alpha) \\ FIRST(\alpha) & \text{, otherwise} \end{cases}
$$

Below the lookahead sets for our productions are shown.

$$
\begin{aligned}
LA(S \to \text{if } BSS_*) &= \{\text{if}\} \\
LA(S \to \text{return NUM;}) &= \{\text{return}\} \\
LA(S_* \to \varepsilon) &= \{\text{else,\$}\} \\
LA(S_* \to \text{else } S) &= \{\text{else}\} \\
LA(B \to (\,\text{NUM}\,)) &= \{(\}
\end{aligned}
$$

No, the grammar is not LL1.