2IS15 Generic Language Technology (2014-2015) Generation of parsers using JFlex and Beaver

Assignment 3 (deadline: October 6th 2014, 09:00h)

Introduction

The goal of the first set of assignments of the course Generic Language Technology is to get acquainted with the basic concepts of scanning and parsing. There are several scanner and parser generators, for instance LEX+YACC, that can be used to generate a scanner and parser. In order to get some idea of the underlying technologies, you have to construct parsers and corresponding scanners using the Java variant version of LEX+YACC: “JFlex” and “Beaver”.

The third exercise focuses on the generation of bottom-up parsers. “Beaver” can be downloaded via `http://beaver.sourceforge.net` (use version `beaver-0.9.11.zip`) and here you can find documentation on how to write a proper language definition. Furthermore, you can find a link to “JFlex” (`http://www.jflex.de/`, version `JFlex-1.4.3`), the scanner generator that has to be used. (it seems that development on “JFlex” has restarted as newer versions have been released in 2014; latest version is `JFlex-1.6.0`).

Transformation of the Pico grammar into bottom-up parser

1. Given a lexical and context-free grammar of a simple language, Pico, a parser has to be generated. For the scanning of the input tokens use the scanner generator (“JFLex”).

   Note that on the web page of Beaver in the command `java -jar beaver.jar [options] language.grammar` you should use `beaver-cc.jar` from the package `beaver-0.9.11` in order to be able to generate a parser. In the package `beaver-0.9.11` you will find an example directory.

```
start-symbol PROGRAM

context-free syntax
  PROGRAM ::= "begin" DECLS (STATEMENT ";")* "end"
  DECLS ::= "declare" (PICO-ID ",")*

context-free syntax
  STATEMENT ::= PICO-ID ":=" EXP

context-free syntax
```

```
  EXP ::= PICO-ID
  EXP ::= NatCon
  EXP ::= EXP "+" EXP  {left}
  EXP ::= EXP "-" EXP  {left}
  EXP ::= EXP "*" EXP  {left}
  EXP ::= "(" EXP ")"

lexical syntax
  PICO-ID ::= [a-z][a-z0-9]*

lexical syntax
  NatCon ::= [0-9]+

lexical syntax
  LAYOUT ::= [\ \t\n\r]*

context-free priorities
  EXP ::= EXP "*" EXP >
  EXP ::= EXP "-" EXP >
  EXP ::= EXP "+" EXP
```

2. Write a number of test programs, both correct and erroneous.

Generate a parser for Oberon-0

1. Given the grammar definition of the Oberon-0 language (see Chapter 6 and the discussion in Chapter 7 of `http://www.ethoberon.ethz.ch/WirthPubl/CBEAll.pdf`), develop “JFlex” and “Beaver” specifications for Oberon-0 language.

2. Use a number of example programs from this book to test the generated parser.

Submission

Submit via PEACH:

1. “JFlex” and “Beaver” specifications for the Pico language, the java-files generated from these specifications, and the example Pico programs.

2. “JFlex” and “Beaver” specifications for the Oberon-0 language, the java-files generated from these specifications, and the example Oberon-0 programs.