dOvs - Parsing

Group 9
Miran Hasanagić - 20084902
Jakob Graugaard Laursen - 20093220
Steven Astrup Sørensen - 201206081

September 21, 2015

1 Introduction

This report describes our approach to develop the parser, which is the first component in the compilation process. This parser is implemented using the ML-Yacc, as it is suggested by the book and the project description.

The first three section describe different parts of the parser development. First, important parts of the tiger grammar are introduced with respect to the ML-Yacc. Secondly, it is described how the semantics actions are used to construct the AST for the program. Then it is shown how conflicts where detected, and what was done in order to resolve them.

Following these sections an overview of the problems encountered and experience gain during the work process is provided. Finally, five tiger programs are provided as test cases for different aspects of the developed parser.

2 The tiger grammar

The tiger grammar was implemented in the file `tiger.grm`. The parser development was divided into two separate parts; first the tiger grammars was implemented, and next the semantics action where added. These two parts reflect the two *program* section for the chapters three and four, respectively. Using this approach it was possible to divide the problem in two parts. First it could be ensured that we did not have any conflicts in the grammars, and afterwards the semantics actions could be implemented in order to create the Abstract Syntax Tree (AST).

In order to specify the grammar and rules Appendix A was used. However, using this grammar directly provided some conflicts and problem, which is discussed below.

3 The abstract syntax

The abstract syntax was provided with the handout file `absyn.sml`. This file provided the types of the different nodes that can be used when creating an AST from a tiger program.

```
Listing test
```

4 Conflict management

4.1 lvalue

4.2 Mutual recursie function declarations

4.3 If-Else conflict

5 Problems encountered & experience gained

5.1 Solving conflicts

5.2 Creating the abstract syntax

6 5 tiger programs

These five tiger programs presented here are created simple in order to focus on different aspects of the implemented parser. These tests contain different syntax errors as required by the project description.

6.1 Test1

6.2 Test1

6.3 Test1

6.4 Test1

6.5 Test1

7 Conlusion