Free University of Bozen - Bolzano

Formal Languages and Compiler

Compiler Project

HSC

Author:

Giorgio Musciagna   gmusciagna@unibz.it
Ravinder Singh   rsingh@unibz.it

December 30, 2017

1 Introduction

The aim of the project is the realization of a compiler for a particular language by means of lex and yacc. A compiler is a program which converts a program written in a language, usually denoted as source into an equivalent program written in another language, usually denoted as target language. The process of compilation comprises two parts: analysis and syntesis.

Analysis: the source program is read and its structure is analysed.
Syntesis: an intermediate code is generated from the intermediate representation of the language returned at the end of the analysis part. An optimization process will occur and the target program will be eventually generated.

The project is focused on the first part of the compilation only i.e. the analysis part which consists of two subtasks:

1. Split the source program and categorize each sequence of characters into tokens by reading it left-to-right. Regular expression are used in order to tokenize the source program. (Lex)

2. Find the hierarchical structure of the source program expressed by recursive rules reflecting a context-free grammar. Moreover, additional information to give meaning to the program are computed involving adding information to the *symbol table* and type checking. (Yacc)

2 Symbol table

Structure of the symbol table.

Name	Type	Value	Scope
price	real	140.05	1
done	boolean	1	0
price	real	9.2	0
rate	real	12.57	0

2.1 Sample Listing

Sample listing

1. First

2. Second

or bullet points . . .

- First

- Second