

# **Compilers: P-language**

*Sam Mylle, Federico Quin*

*University of Antwerp*

May 28, 2017

## 1 Result

We have allowed all mandatory C constructions in our grammar and we are able to create the AST and the symbol table. We have also implemented some extras, e.g. continue and break statement and for loops.

There is error checking. The symbol table has been adjusted to provide the required functionality for the translation unit. The translation unit is also fully operational.

## 2 Grammar

Currently, the grammar we wrote supports all mandatory C constructions, and of course also the optional constructions we mentioned above. The grammar can be found in the ‘res’ directory. We have provided some comments and we tried to make the names of nonterminals/terminals as clear as possible.

## 3 Sources

Our compiler currently consists of 4 big parts: the processing and creation of the Abstract Syntax Tree (`ASTCreator.py`, `ASTBuilder.py`, ...), the construction of the Symbol Table (`SymbolTable.py`, `SymbolTableBuilder.py`), Syntax Analysis (`TypeChecking.py`, `ExistenceChecker.py`, ...) and the actual translation to P code (`PTranslation.py`).

The code for the different big parts (except for p translation) is located in different subdirectories.

- AST - all files related to the Abstract Syntax Tree structure, creation, etc.
- ST - all files related to the Symbol Table.
- SA - all files related to Syntax Analysis.
- UTIL - general files such as the TypeDeductor which is used in different parts of the compiler.

Firstly, the AST related classes make the decorated AST.

Secondly, the symbol table related classes make the symbol table based on this AST. The symbol table is a tree like structure. It provides a separate level for the global scope, and different sublevels for all the local scopes.

Thirdly, syntax analysis does its part. This involves type checking of variables (for example assignments, return types, ...) and existence checking (checking if a variable/function is referenced before assignment). If anything fails, it will yield an error with an appropriate error message and on which line plus column

number the error is located.

Compared to the first evaluation, we now added an extra part: the translation unit. Using the resulting AST and the symbol table, the C program is translated into a P program.

## 4 Testing

For testing our compiler, we chose for pytest. We have 43 test programs, where each program tests one encapsulated part of functionality. These tests also include several death tests.