LW4

DA53: Compilers and Language Theory

stephane.galland@utbm.fr

Intermediate Code Generation

00 | Goal

The goal of this on-computer tutorial session is to write a small compiler that is able to compile a Tiny Basic program and generate three-address code program.

The development must be done in Java.

You must reuse your works from the previous on-computer tutorial sessions.

01 | Working Steps

01.1 Three-Address Code

- Select the three-address instructions to support and put them in the ThreeAddressInstruction enumeration.

- Write the class ThreeAddressRecord. It is a record that contains the three-address instruction, the parameters (String), and the result (String). A record may also be associated to a label, and optionally to a comment. **Note that during this tutorial, you should not compute the addresses of the instructions and of the variables (this computation will be done during the tutorial TU5).**

- Write the class ThreeAddressCode. It is a list of three-address records in one hand. In the other hand, it should permit to :

    1. create temporary variables ;

    2. maintain a mapping between a TinyBasic line and a three-address code line ;

    3. access to the symbol table ;

    4. create labels.

01.2 Three-Address Code Generation

- Update classes of the syntax tree with the generate() function. This function put in the given ThreeAddressCode all the three-address statements that may corresponds to the syntax tree node.

- Note that the gosub and goto statements take a TinyBasic line number. You must, dynamically, retreive the three-address code line that is corresponding to a given TinyBasic line. This mapping is a part of the generated code by the ThreeAddressCode class.

- Note that the symbol table may contains string literals (constants). You may define the three-address variables that are corresponding to these constants prior to any execution of the program.