Project 2: Type checking and Intermediate Code Generation

Due Mar 6, 2014

In this project, you are required to apply type checking to the C code parsed by your project1 and to generate an intermediate representation of the type-checked code. Your IR should be an AST (abstract syntax tree) representation of the input program, where each expression in the AST is tagged with its type information. After parsing an input code, e.g., the following sample input,

```
float a[100][100], b[100][100], c[100][100];
int i, j, k;
i = 0;
while (i < 100) {
  j = 0;
  while (j < 100) {
    if (!(c[i][j] == 0.0))
       c[i][j] = 0.0;
    k = 0;
    while (k < 100) {
       c[i][j] = c[i][j] + a[i][k] * b[k][j];
       k = k + 1;
    }
    j = j + 1;
  }
  i = i + 1;
}
```

your project needs to print out the AST to standard output as result, with a type assigned to each expression contained in the input code. An example output could be like the following.

```
AST:
  TypeDecl("int", (Var("i"), Var("j":int), (Var("k":int)))
  Assign(Var("i"), Plus(Var("i"), intval(1)): int): int
```

Implicit type conversion is NOT supposed to be supported. Specifically, if the input program is adding an integer with a float type, an error should be reported.

You are expected to build your project2 on top of your working directory for project1. However, if your project1 does not work, or if you prefer to switch the language used for project implementation, you can invoke "init-proj2 <type>", where <type> is poet, yacc, or other, to obtain a reference implementation of project1 as a basis for your project2 implementation. The reference proj1 implementation will be made available after project1 has been graded.

The given correct input test file, e.g., right1.inp, can be used to similarly test your project2 implementation. For these files, your projec2 need to print out an internal AST representation of the input code. Additionally, please create one correct input test file, right2,inp, and one additional wrong input test file, wrong2.inp, which are syntactically correct but contain some type errors. Your project2 should be able to report the type errors when parsing the wrong input files. The project will be graded based on both the quality of your project implementation and the quality of your test files.

To submit your project when ready, simply go to the parent of your project directory on the designated server and run "submit-proj2 <type> <dir>", where <type> is poet, yacc, or other, and <dir> is the name of the directory that contains your project code.