Ciências ULisboa | Informática

Compiler Techniques 2024/25
AGUDA M1: Tests
Deliver by April 8, 2025

For milestone 1 (M1) we shall prepare tests for the AGUDA programming language.

Test proposals are issued via a pull request at `https://git.alunos.di.fc.ul.pt/tcomp000/aguda-testing`.

There are two folders,:

- `test/valid` for *valid* tests, that is, programs that should compile;
- `test/invalid` for *invalid* tests, including source code that does not represent a program, either because of a lexing, parsing or validation problem.

Each program (valid or invalid) must be included in a distinct folder. Valid test folders should contain two files. For a program `p`, include a `p.agu` (the source code) and a `p.expect` (a txt file with the expected output of program `p`). Folders for invalid tests contain one `.agu` file only.

Naming your folders and tests:

- Folders start with your student number. For example `65432_undeclared_variable` or `65432_matrix_multiplication`
- No special rules for naming your tests; descriptive names encouraged. For example `matrix_multiplication.agu` and `matrix_multiplication.expect`

The first line in each `.agu` file should identify the author. For example

```
-- Author: 65432, Ada Lovelace
```

You are to prepare:

- 6 invalid tests: one with a lexical problem, two with syntactic problems, three with semantic problems (undeclared variables, type checking problems)
- 10 valid tests: Try exercising different features of the language, corner cases, features that may be eassily misunderstood by programmers. One of you test cases must contain at least 20 lines of code. Use comments at will.

Example to be included in the `test/valid/tcomp000_collatz` folder.
File `collatz.agu`:

```
-- Author: tcomp000, Vasco T. Vasconcelos

-- See https://en.wikipedia.org/wiki/Collatz_conjecture
   for the Collatz Conjecture

let f (n) : Int -> Int =
  if n % 2 == 0 then n / 2 else 3 * n + 1

let conjecture (n) : Int -> Unit =
  if f (n) != 0 then conjecture (n - 1)

let _ : Unit =
  conjecture(43786) ;
  print("Terminates on input 43786!")
```

File `collatz.expect`:

```
Terminates on input 43786!
```