CSC488
ASSIGNMENT 3
TESTING

DANIEL BLOEMENDAL

Contents

1. Overview

Our approach to testing was to stress each semantic action. To clarify this, it should be noted that some semantic actions perform internal book keeping rather than a check. So we decided to write one or more tests for every semantic action that involve a semantic check of some kind. Please refer to the tests/README.tests for a listing of all tests.

2. Passing tests

The passing tests are relatively straight forward. We tried to include a variety of relatively complicated code stressing many of the features of the language. Some individual tests for possible corner cases were included as well. Those tests are `pass/S10.488`, `pass/S50.488`, `pass/S51.488` and `pass/S52.488`. Please see the comments within the tests for more information on each test.

3. Failing tests

As mentioned in the overview, our approach for failing tests was to include one or more case for each semantic action involving checks. Each failing tests includes a comment describing the point of failure and some meta data `@line=<line>` or `@line=[<line_1>, <line_2>, ..., <line_n>]`. This is important to note because the test runner `TESTS.py` that I developed uses this meta data to verify that the test does indeed fail on the line(s) specified. The test cases are named according to the semantic actions they fail on. The test runner also uses the filenames to ensure that the failures do indeed correspond to the correct semantic action.

4. Test runner

The test runner `TESTS.py` is a Python program which tests each passing and failing test case, and uses the meta data in the failing test cases to verify that the failures are the expected failures.