UCT 2005 CSC305 Compilers

Practical Assignment 1: Abstract Syntax Trees

Create an Abstract Syntax Tree from the concrete syntax information parsed by the MiniJava parser. Instead of hand-coding your nodes and data structures, use the production rule transformations and “abstract syntax tree” section features of SableCC.

Your main program must output the AST for a file that is read from standard input. Your output must make obvious what the structure and contents are, as this will form the primary basis for determining correctness. For simplicity, you may draw your trees in ASCII, similar to the example below:

```
Root
----|FirstChild
----|SecondChild
--------|GrandChild1
--------|GrandChild2
----|ThirdChild
```

Test your code with and provide output for the 2 Minijava files attached. In each case, use output redirection to capture the output and save it to appropriately-named files.

You may use any code available on the textbook’s website, but are not required to do so. Use the included grammar file as a starting point.

> Assume that all the test/sample programs will be error-free – you may do type-checking but it is not required.

Your assignment will be marked according to the following criteria:
Correctness (40%) (test programs generate correct output at each layer)
Documentation (15%) (comments within hand-written source and short report/README, 3 pages max)
Efficiency (15%)
Stress (20%) (hidden test programs generate correct output at each layer)
Creativity (10%) (code optimisations, etc.)