

Homework 4

Amy Murphy

31 Oct 2002

You will not hand in the answers to this homework assignment. Instead, at the beginning of class on **7 November 2002** you will be asked to solve one of the problems as a closed notes, closed book quiz. The question to be solved will be randomly selected.

1. Take the source code for Project 4 and compile it as is. Feed it a Java program as input. You will notice in the output that the bodies of methods are indented one level (4 spaces). A little inspection of the code will reveal that this is because the global variable `indent_level` is incremented and decremented at the beginning and end of function `parse_compound_stmt`. Why doesn't that function automatically cause the bodies of `while` loops, `if` statements, etc. to be indented, too?
2. In Project 4, in function `parse_field_tail` in `parser.c` there is a comment noting a problem for your formatting program: in a field of a class, if the first token after any modifiers is an identifier, that identifier may name a type or it may be the name of a constructor. You need to print it in red in the latter case, but not the former. Unfortunately, the code doesn't know which case applies at the time of the call to `parse_qualified_name`, so when that routine matches the identifier it won't know what to do. Discuss how you might solve this problem. (This answer should be written in prose. Don't send code.)
3. Identify the spot in the provided code of Project 4 (it's a single line) where you will need to make changes to control the spacing between tokens within a line of Java source.
4. Prove that the following CFG is ambiguous:

$$string \rightarrow string\ string \mid 0 \mid 1$$

5. Write a CFG for strings containing 0 and 1 such that all strings generated by your grammar contain the same number of 0's and 1's. Show parse trees for the strings 0001101 and 0110.
6. Informally $FIRST(A)$ is defined to be the set of tokens that can begin some string of tokens derived from A. Compute $FIRST(stmt)$ for the following grammar fragment:

$$\begin{aligned} stmt &\rightarrow labeledOpt\ loop \\ stmt &\rightarrow IDENT := expr \\ labeledOpt &\rightarrow NUMBER : \\ labeledOpt &\rightarrow \epsilon \\ loop &\rightarrow WHILE\ condition\ \{stmtList\} \\ loop &\rightarrow DO\ \{stmtList\}\ UNTIL\ condition \end{aligned}$$