CSCI 3155: Principles of Programming Languages

Exercise sheet #2

5th June 2007

You may hand in this worksheet together with up to two other students, or by yourself. If you hand the worksheet in collaboratively and decide that you would like to work together for the rest of the semester, please also pick a *group name* and enter it below.

Starting tomorrow, you *must* hand your worksheets in in a group of two or three.

Name #1: _____

Name #2: _____

Name #3: _____

Group Name: _____

Syntax

Exercise 1. (**Skill 2.1**) For the following language properties, determine whether they are part of the *syntax*, *static semantics* or *dynamic semantics*.

Property	**Syntax**	**Static Semantics**	**Dynamic Semantics**
Division by zero raises an exception			
Multiplication has a higher precedence than addition			
A block of statements is either a single statement or a sequence of statements enclosed by curly braces			
For any variable name, a class may declare at most one variable of that name			
The body of a `while` loop is executed until its precondition fails			

Exercise 2. Consider the following BNF grammar with start symbol $\langle S \rangle$:

$$\begin{array}{lcl} \langle S \rangle & \rightarrow & \langle A \rangle \langle S \rangle \langle B \rangle \mid \langle C \rangle \\ \langle A \rangle & \rightarrow & \mathbf{a} \\ \langle B \rangle & \rightarrow & \mathbf{b} \mid \varepsilon \\ \langle C \rangle & \rightarrow & \mathbf{c}\ \langle B \rangle \end{array}$$

(a) (**Skills 2.2, 2.4**) Out of "ε", "**acb**", "**aabb**", precisely one can be generated by the above grammar. Pick the right one and draw a parse tree for it. If there are multiple possible parse trees you may choose any one.

(b) (**Skills 2.2, 2.5**) Is the grammar ambiguous? If so, prove that it is.

(c) (**Skills 2.2, 2.4**) Which of the following can be generated by the above grammar:

(i) **acbb**
(ii) **aaaaacb**
(iii) **aaacc**
(iv) **aacbbbbb**

Exercise 3. Consider the grammar from Appendix A.

(a) (**Skills 2.2, 2.3**) Generate four distinct program fragments that can be generated from the nonterminal ⟨*Type*⟩.

(b) (**Skills 2.2, 2.6**) Consider the nonterminal ⟨*Expr*⟩. This nonterminal allows the construction of expressions involving the binary operators +, AND, and <. For each operator, determine the operator's associativity, and explain, using a parse tree for illustration.

Exercise 4. Assume a terminal "id", which describes tokens that have the form of BNF identifiers in pointy braces, such as "⟨*id*⟩" or "⟨*A*⟩" or "⟨*Start*⟩".

Further assume a terminal "token", which describes tokens that have the form of BNF tokens, such as "**while**", "**;**" or "[".

Using these two terminals, describe the grammar of BNF in BNF.

Exercise 5. Consider the following BNF grammar with start symbol $\langle S \rangle$:

$$
\begin{array}{lcl}
\langle S \rangle & \rightarrow & \langle L \rangle \mid \ \mathbf{!}\ \langle A \rangle\ \langle L \rangle\ \mathbf{!} \\
\langle L \rangle & \rightarrow & \langle A \rangle \mid \langle A \rangle\ \langle L \rangle \\
\langle A \rangle & \rightarrow & \langle A \rangle + \langle B \rangle \mid \langle B \rangle \\
\langle B \rangle & \rightarrow & \mathbf{a} \mid \mathbf{!}\ \langle S \rangle\ \mathbf{!} \mid \mathbf{b}
\end{array}
$$

(**Skills 2.2, 2.5**; tricky!) Is this grammar ambiguous? If so, prove that it is ambiguous. If not, explain why you believe that it is not, by analogy with the examples used in the book for discussing ambiguity.