University of Toronto
CSC324 – Principles of Programming Languages, Fall 2004

Assignment # 1

Due Date

This assignment is due on **Friday October 1, 2004 at 5 p.m.**

Silent Policy

A silent policy will take effect 24 hours before this assignment is due. This means that no question about this assignment will be answered, whether it is asked on the newsgroup, by email, or in person.

Handing in this Assignment

You should hand in this assignment on paper in the CSC324 drop box in the Bahen Computer Lab BA2220 or electronically. See http://www.cdf.toronto.edu/workathome/index.php3?id=42 for electronic submission instructions. If you submit your assignment both electronically and on paper, only the electronically submitted version will be marked. If you submit your assignment electronically, it is your responsibility to make sure it can be printed properly on CDF. Unreadable portions of an assignment will be given 0 marks. Marks also will be deducted for incorrect electronic submission.

We will accept your electronic submission in plain ASCII (in which case your submission file must be called a1.txt), in PostScript (a1.ps), or in PDF (a1.pdf). We recommend ASCII – the electronic submission option is meant to be convenient and easy, not to create additional work for you, so don't waste your time on a fancy presentation. You may draw your parse trees using any reasonable representation, as long as it is clear.

Please include this cover sheet

Last Name:_____ **First Name:**_____

Student #:_____ **CDF Login:**_____

Email:_____ **Date & Time:**_____

Grace Days used to date:_____

Grace days used for this assignment: _____

I understand that collaboration is not allowed. All answers are my own, written in isolation, without help from others. This submission is in accordance with the University of Toronto Code of Behavior on Academic Matters (http://www.artsandscience.utoronto.ca/ofr/calendar/rules.htm#behaviour)

Signature: _____

University of Toronto
CSC324 – Principles of Programming Languages, Fall 2004

Assignment # 1

[1 mark]
Fill out the cover sheet correctly.

[28 marks, 4 marks each]

1. For each of the following languages, (1) provide a context-free grammar in BNF that generates all strings in the language and no other strings or say it cannot be done, and (2) provide a regular expression that accepts all strings in the language or say it cannot be done. If you claim that a context-free grammar / regular expression cannot be provided, you do not have to explain why.

   a) All binary strings with no adjacent zeros.

   b) All binary strings with an equal number of zeros and ones.

   c) All strings over {1,2,3,4,5}, such that the sum of the number that corresponds to a string and the number that corresponds to the reverse of that string is made up completely of 6's.

      Example: 12345 + 54321 = 66666, therefore 12345 is in the language.

   d) All non-empty strings over {1,2,3,4,5}, such that the sum of the number that corresponds to a string and the number that corresponds to the reverse of that string is made up completely of 6's.

   e) All strings over {a,b,c} of the form $a^k\ b^m\ c^n$, where n, m, $k \geq 0$.

   f) All strings over {a,b,c} of the form $a^n\ b^m\ c^n$, where n, $m \geq 1$.

   g) All strings over {a,b,c} of the form $a^n\ b^{n+m}\ c^m$, where n, $m \geq 1$.

[9 marks]

2. Consider the following grammar:

<S> ::= (<L>) | a
<L> ::= <L> , <S> | <S>

[1 mark]

a) What are the terminals in this grammar?

[1 mark]

b) What are the non-terminals in this grammar?

[1 mark]

c) What is the start symbol in this grammar?

[1 mark]

d) Give three strings generated by this grammar.

[1 mark]

e) Give three strings **not** generated by this grammar.

[2 marks]

f) Show a right-most derivation for (a , ((a , a) , (a , a)))

[2 marks]

g) Show a parse tree for (a , ((a , a) , (a , a)))

[6 marks]

3. Consider the following regular expression:

a(a*bb*a)*a*bb*|b(a|b)*

Specify a regular grammar that generates all the strings accepted by this regular expression and no other strings. Include all components of the grammar. The production rules should be specified in BNF notation. Simpler grammars will be given more marks.

[6 marks]

4. Consider the following grammar:

terminals: **{a,b,c,+,-,*}**
non-terminals: **{ <expr>, <term>, <factor>, <const> }**
start symbol: **<expr>**
productions:

<expr> ::= <expr> + <term> | <expr> - <term> | <term>
<term> ::= <term> * <factor> | <factor>
<factor>::= <expr> | <const>
<const>::= a | b | c

Is the grammar ambiguous? If you think the grammar is ambiguous, prove it. If you think it is not, explain why.

[20 marks]

5. The set of First-Order Logic Formulas can be defined inductively as follows:

i. An atomic formula is a formula
ii. If **A** and **B** are formulas, then **not A** , **A and B** , **A or B** , and **A implies B** are formulas
iii. If **A** is a formula and **x** is an identifier, then **exists x.A** and **forall x.A** are formulas

Assume that there are production rules for generating atomic formulas and identifiers starting from non-terminals **<atomic>** and **<identifier>** , respectively.

For example, suppose that **<identifier>** generates an identifier **y** and **<atomic>** generates atomic formulas **P**, **Q**, and **R**. Then

(not exists y. P) and (forall y. not Q) implies R

is a First-Order Logic Formula.

Your task is to give an unambiguous context-free grammar that generates the language of First-Order Logic Formulas.

In addition, the following properties should hold:

a) The precedence order is as follows (from highest to lowest priority):
   1. not
   2. and
   3. or
   4. implies
   5. exists, forall

   and parentheses override the precedence order.

b) **and** and **or** are left-associative, **implies** is right-associative.