 Questions 1: <http://www.cs.unb.ca/~wdu/cs4613/a3ans.htm>

1. (15%) Write EBNF descriptions for the following

a)      A Java class definition header statement

The following is an example class header statement:

public class A extends B implements C, D

where “public” is a modifier and “A” ,”B”, “C”, and “D” are identifiers. Assume non-terminal <id> is given.

<method\_head> -> [public] [(abstract | final)] class <id> [extends <id>] [implements <id> {, <id>}]

b)      A C/C++/Java switch statement

The following is an example switch statement:

switch (a+b)

{

case 1 : x = 7; break;

case 2 :  x = 8; break;

default : x = 9;

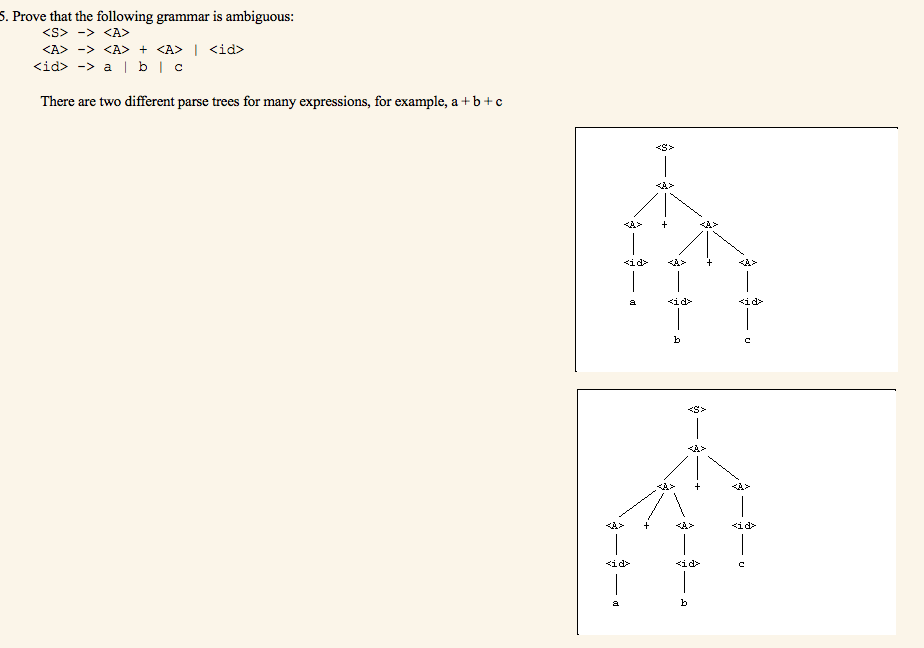
            }

            where “a+b” is an expression, “1” and “2” are literals, and “x=7;break;”, “x=8;break;” and “x=9;” are statement lists. Assume non-terminals <expr>, <literal>, and <stmt\_list> are given.

      <switch> -> switch ‘(‘ <expr> ‘)’ ‘{‘ {case <literal> : <stmt\_list>} [default : <stmt\_list>] ‘}’

Question 2, 3, 5, 6: ASK MATT FOR BOOK EXAMPLE

Question #4:



Question #7:

Compute the weakest precondition for the assignment statement a := 2\*(b – 1) –1 given the postcondition {a > 0).

            Substitute the right hand side of the assignment statement for a in the postcondition

            2\*(b-1) > 0

       2b–2  > 0

          2b > 2

            b > 1

**Compute the weakest precondition for each of the following assignment statements and postconditions:**  
      (a) a = 2 \* (b – 1) – 1 {a > 0}  
           2 \* (b – 1) – 1 > 0  
           2 \* b – 2 – 1 > 0  
           2 \* b > 3  
           b > 3 / 2  
      (b) b = (c + 10) / 3 {b > 6}  
           (c + 10) / 3 > 6  
            c + 10 > 18  
            c > 8  
      (c) a = a + 2 \* b – 1 {a > 1}  
           a + 2 \* b – 1 > 1  
           2 \* b > 2 – a  
           b > 1 – a / 2  
      (d) x = 2 \* y + x – 1 {x > 11}  
           2 \* y + x – 1 > 11  
           2 \* y + x > 12

Question #8:

**Compute the weakest precondition for each of the following sequences of assignment statements and their postconditions:**  
      (a) a = 2 \* b + 1  
           b = a – 3  
           {b < 0}  
           a – 3 < 0 -> a < 3  
           2 \* b + 1 < 3 -> 2 \* b < 2 -> b < 1

          The weakest precondition is {b < 1}

Questions 7 and 8 source: <https://forpakarana.wordpress.com/2013/03/26/>

Question #6:

<https://ejopapasaurus.wordpress.com/2013/03/13/chapter-3-describing-syntax-and-semantics/>

21.  Using the virtual machine instructions given in section 3.5.1.1, give an operational semantic definition of the following

a. java

loop: (do body)  
if<relational\_expression> goto out  
goto loop  
out: …

b. Ada (for)  
for I in first..last loop  
I=first  
loop: if I<last go out  
…  
I=I+1  
goto loop  
out: …

c. C++ if-then-else

if(condition){  
statement1;  
statement2;  
}  
else{  
statement3;  
statement4;  
}

d. C for  
for(expr1;expr2;expr3)  
evaluate (expr1);  
loop=control= evaluate(expr2)  
if control == 0 goto out  
evaluate (expr3)  
goto loop  
out: …

Question 5:

<https://wildanarifrahmanmauwa.wordpress.com/2014/10/15/chapter-3-r-sebesta/>

**19. Write an attribute grammar whose BNF basis is that of Example 3.6 in Section 3.4.5 but whose language rules are as follows: Data types cannot be mixed in expressions, but assignment statements need not have the same types on both sides of the assignment operator.**

1) Syntax rule: <assign> → <var> = <expr>

2) Syntax rule: <expr> → <var>[2] + <var>[3]

Predicate: <var>[2].actual\_type == <var>[3].actual\_type

3) Syntax rule: <expr> → <var>

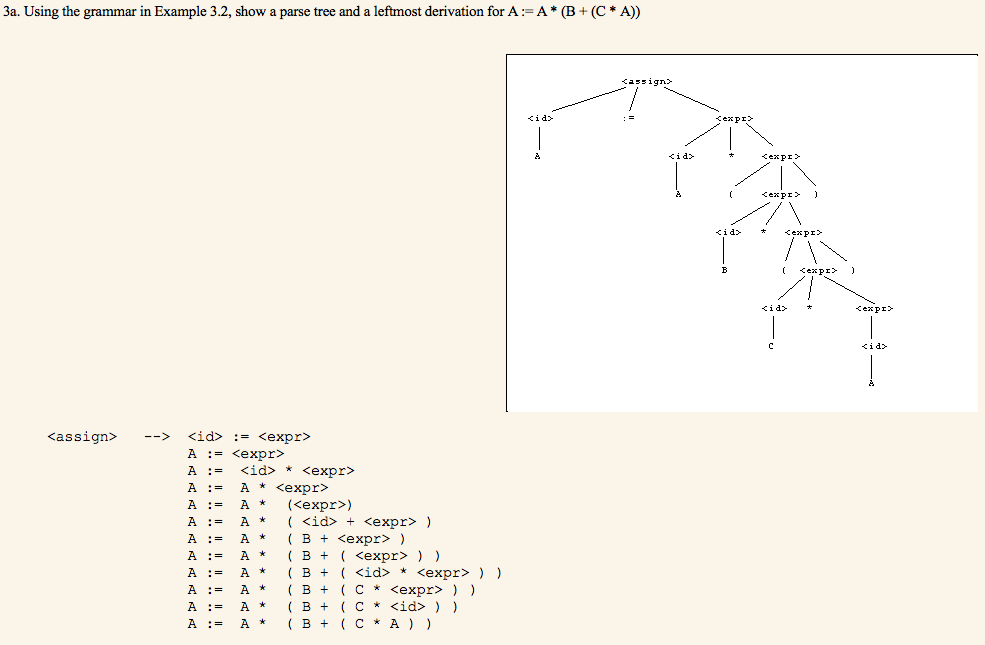
4) Syntax rule: <var> → A | B | C

Semantic rule: <var>.actual\_type ← lookup(<var>.string)

**OR**

Replace the second semantic rule with:  
<var>[2].env ← <expr>.env  
<var>[3].env ← <expr>.env  
<expr>.actual\_type ← <var>[2].actual\_type  
predicate: <var>[2].actual\_type = <var>[3].actual\_type

Question #3:



<http://www.cse.scu.edu/~rdaniels/html/courses/Coen171/HW1Soln.htm>

Question #2:

<https://soccerdics.wordpress.com/2013/03/26/chapter-3-assignment-from-mr-tri-djoko-wahjono-ir-m-sc/>

*Rewrite the BNF of Example 3.4 to give + precedence over \* and force + to be right associative.*  
Answer:

<assign>→<id>=<expr>  
<id>→A|B|C  
<expr>→< term >**\***< expr >| <term>  
<term>→<term>**+**<factor>| <factor>  
<factor>→(<expr>) | <id>