Bradley Trowbridge

Programming language

Ch 3 Assignment

pg 157

#2 a, b only

Write EBNF descriptions for the following:

1. A Java class definition header statement

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

b. A Java method call statement

<object\_method\_call> -> <object\_id>.<method\_id>([<inputs>]);

<inputs> -> <input> | <input> , <inputs>

<input> - > <id> | <literal>

c. A C switch statement

<switch\_stmt> ->switch(<expr>) case <literal> : <stmt\_list>

{case <literal> : <stmt\_list>} [default: <stmt\_list>]

#6 all

6. Using the grammar in Example 3.2, show a parse tree and a leftmost derivation for each of the following statements:

a. A = A \* (B + (C \* A))

<assign> -> <id> = <expr>

A = <expr>

A = <id> \* <expr>

A = A \* <expr>

A = A \* (<expr>)

A = A \* (<id> + <expr>)

A = A \* (B + <expr>)

A = A \* (B + (<expr>))

A = A \* (B + (<id> + <id>))

A = A \* (B + (C \* <id>))

A = A \* (B + (C \* A))

1. B = C \* (A \* C + B)

<assign> -> <id> = <expr>

B = <expr>

B = <id> \* <expr>

B = C \* <expr>

B = C \* (<expr>)

B = C \* (<id> \* <expr>)

B = C \* (A \* <expr>)

B = C \* (A \* <id > + <id>)

B = C \* (A \* C + B)

1. A = A \* (B + (C))

<assign> -> <id> = <expr>

A = <expr>

A = <id> \* <expr>

A = A \*<expr>

A = A \* (<expr>)

A = A\*(<id> + <expr>)

A = A \* (B + (<expr>))

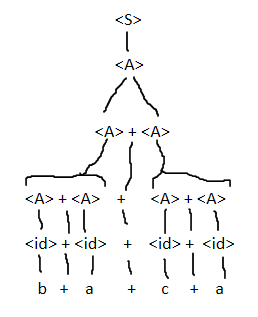
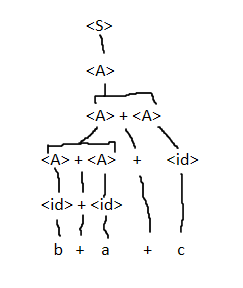
A = A \* (B + (<id>))

A = A \* (B + (C))

#8

8. Prove that the following grammar is ambiguous:





#11 all

<S> → <A> a <B> b

<A> → <A> b | b

<B> → a <B> | a

Which of the following sentences are in the language generated by this

grammar?

a. baab

b. bbbab

c. bbaaaaa

d. bbaab

a&d

#12

Consider the following grammar:

<S> → a <S> c <B> | <A> | b

<A> → c <A> | c

<B> → d | <A>

Which of the following sentences are in the language generated by this

grammar?

a. abcd Yes

b. acccbd No

c. acccbcc No

d. acd No

e. accc Yes

a&e

#23

Compute the weakest precondition for each of the following assignment

statements and postconditions:

1. a = 2 \* (b - 1) - 1 {a > 0}

0 < 2 \* (b – 1) – 1

½ + 1 < b

{b > 3/2}

1. b = (c + 10) / 3 {b > 6}

6 < (c + 10) / 3

18 < c + 10

{c > 8}

1. a = a + 2 \* b - 1 {a > 1}

1 < a + 2 \* b – 1

2 < a + 2 \* b

{b > (2-a)/2}

1. x = 2 \* y + x - 1 {x > 11}

2 \* y + x – 1 > 11

2 \* y + x > 12

{y > (12 – x ) / 2}

(#28 is for graduate students, but I will give 10 points extra credit)

28. Prove the following program is correct:

{n > 0}

count = n;

sum = 0;

while count <> 0 do

sum = sum + count;

count = count - 1;

end

{sum = 1 + 2 + . . . + n}

Proof:

Given: sum = sum + count

And: count = count – 1

Iteratively we evaluate:

sum = count + (count – 1) + (count – 2) … + 1 +(break condition count == 0)

Since count = n :

sum = n + (n-1) + (n-2) … + 3 +2 + 1

With per condition n > 0 and rule of addition

sum = 1 + 2 + 3 +…(n-2) + (n-1) + n

Therefore this program meets its post condition {sum = 1 + 2 + . . . + n}