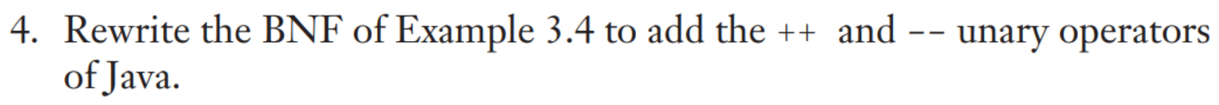
Jeffrey Lansford

9/17/2020

Chapter 3

4.



4. Rewrite the BNF of Example 3.4 to add the ++ and -- unary operators of Java.

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

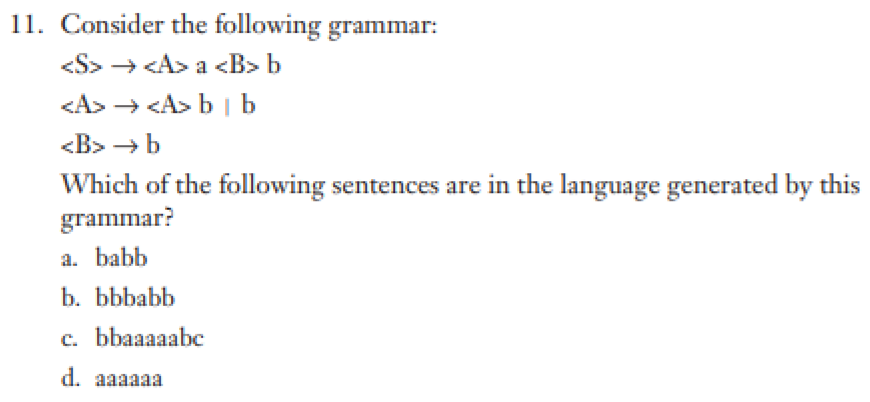
<id> -> A | B | C

<expr> -> <expr> + <term> | <term>

<term> -> <term> * <factor> | <factor>

<factor> -> ( <expr> ) | <id> | <id>++ | <id>--

11.



11. Consider the following grammar:

<S> → <A> a <B> b

<A> → <A> b | b

<B> → b

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

a. babb

b. bbbabb

c. bbaaaaabc

d. aaaaaa

a & b

<S> -> <A>a<B>b

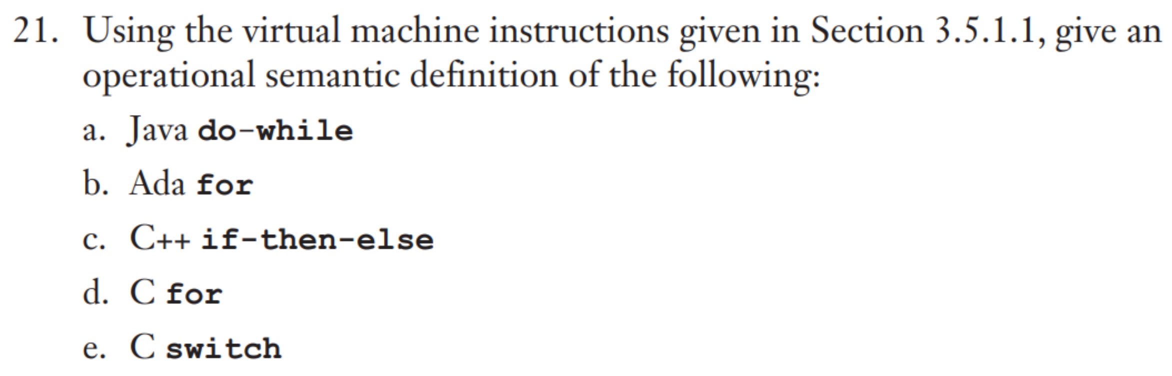
➔ babb

<S> -> <A>a<B>b

➔ <A>babb

➔ <A>bbabb
➔ bbbabb

21.



21. Using the virtual machine instructions given in Section 3.5.1.1, give an operational semantic definition of the following:
    a. Java **do-while**
    b. Ada **for**
    c. C++ **if-then-else**
    d. C **for**
    e. C **switch**

Java Do-while

do { statements} while (expr);

Operational Semantic:

L1:

…

Statements

…

If ( expr == True ) goto L1

…

Ada for loop

For var in A .. B loop

    Statements

out loop

Operational Semantic:

var = A

L2:

```
If (var <= B) goto L1
goto out
L1:
…
Statements
…
Var = var +1
Goto L2
out:
…
```

C++ if then else

```
If ( expr) {
        Statements
}
Else {
        Statements
}
```

Operational Semantic:

```
If (expr == True) goto L1
Goto L2

L1:
…
Statements
…
Goto out
L2:
```

```
…
Statements
…
out:
…
```

C For

```
For (expr1;expr2;expr3){
        statements
}
```

Operational Semantic:

```
expr1
L2:
If (expr2 == True) goto L1
goto out
L1:
…
statements
…
expr3
goto L2
out:
…
```

C switch

```
Switch (A) {
        case c1 : S1
        case  c2: S2
```

```
    …
    Case cn: SN
    <default>
}
```

Operational Semantic:

```
If (A == c1) goto C1
If (A == c2) goto C2
…
If (A == cn) goto CN
goto out

C1: S1
C2: S2
…
CN: SN
out:
…
```