Will Christie
Ryan O'Connell

Lab 2 Write Up

1. Grammars: Synthetic Examples:
    a. Judgments:

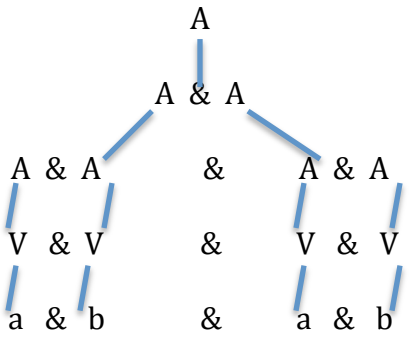
$$\frac{}{a \in VObjects} \qquad \frac{}{b \in VObjects}$$

$$\frac{s1 \in AObjects \quad s2 \in AObjects}{s1 \;\&\; s2 \in AObjects}$$

$$\frac{S \in V\ Objects}{S \in AObjects}$$

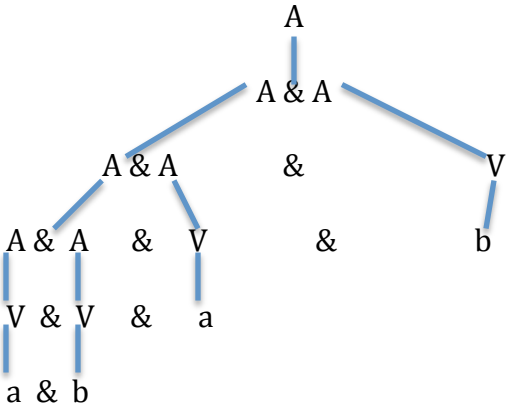
    b. The grammar from the previous portion is ambiguous because we are able to obtain the same result through multiple parse trees.
    Desired sentence: **a & b & a & b**

1st parse tree:



Output: **a & b & a & b**

2nd parse tree:



Output: a & b & a & b

c. L(S) = { a^n, b^n, c^n; n >=0 }
   L(A) = { a^n; n >=1 }
   L(B) = { b^m; m>=0 }
   L(C) = { c^n; n >= 1 }
d. Derivations: S ::= AaBb A::=Ab|b B::= aB|a
   i. baab

      S ::= AaBb => baBb => **baab**

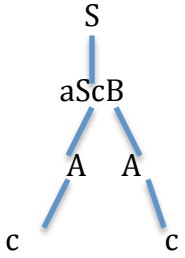
   ii. bbbab--"bbbab" is not in the language generated by the above grammar.
   iii. bbaaaaa—"bbaaaaa" is not in the language generated by the above grammar.
   iv. bbaab

      S ::= AaBb => AbaBb => bbaBb => **bbaab**

e. Parse Trees: S ::= aScB|A|b A::=cA | c B::= d | A
   i. abcd

Output: abcd

   ii. acccbd --- "acccbd" is not in the language.
   iii. acccbcc---"acccbcc" is not in the language.
   iv. acd---"acd" is not in the language.
   v. accc



Output: accc

2. Grammars: Understanding a language.
   a. Considering the following two grammars:
      i. The expressions generated by the two grammars can be the same but can be generated differently.

   L(e) = { (operand)^n (operand operand)^m; n = 1 m >=0}

      ii. The grammars can generate the same expression, with the bottom grammar using left associativity and the top grammar

using right associativity. Because of this, the grammars do not generate the same expression.

b. Using the fact that: 5 – 3 << 3 === 16

```
val a = 5 - 2 << 3
val minus = (5 - 3) << 3
val shift = 5- (1 << 3)
if (a == minus)
        println("- has higher precedence.")
if (a == shift)
        println("<<" has higher precedence.")
```

By inputting this expression into the scala interpreter, it was determined that "-" has higher precedence than "<<" because the minus expression was found to be in agreement with our original test expression that 5 – 3 << 3 is in fact 16.

c. BNF Grammar:

<FP> ::= <Op><Term>.<N><Exp>
<Term> ::= <Nzint><N>
<Exp> ::= E <Op> <Nzint> <N> | ε
<N> ::= <Nzint> <N> | <Z><N> | ε
<Op> ::= - | ε
<Nzint> ::= 1|2|3|4|5|6|7|8|9
<Z> ::= 0