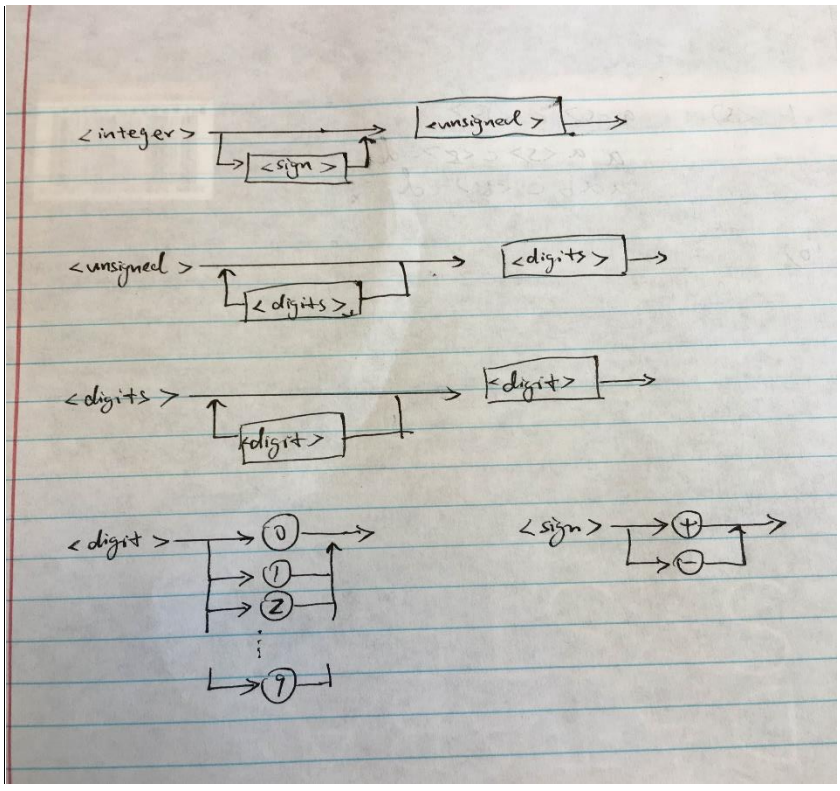
ECS140A
Assignment1
Zuolin Li
ID:998829351

1. (a) We want to generate aabccd. Since the second letter is a and the third is a b, so we have:
   <S> => a<S>c<B> => aa<S>c<B>c<B> => aabc<B>c<B> => aabc<A>cd => aabcccd. which is not aabccd, so we can't generate the string from the grammar..

   (b) Since the second letter is c, so <S> => a<A>c<B> =>acc<B>. However there is no way to generate a b through the grammar. So we can't derive the string from the given grammar.

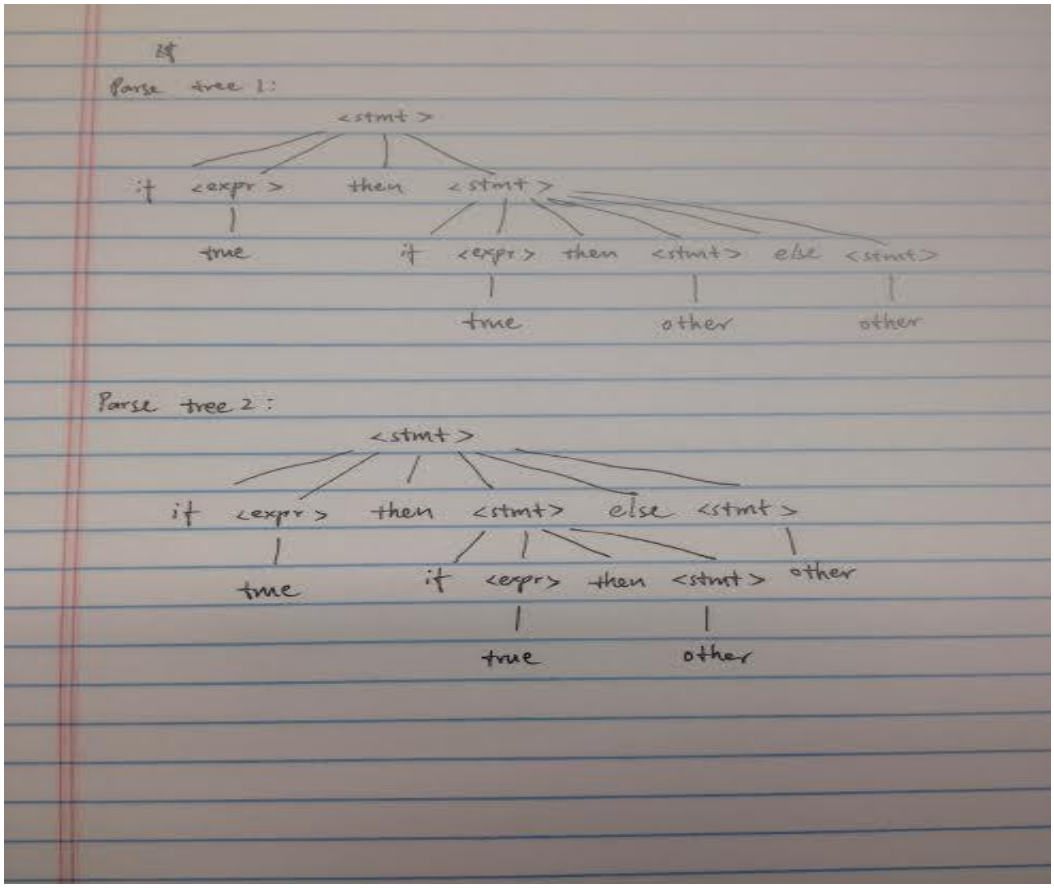
   (c) <S> => a<S>c<B> => a<A>c<A> => ac<A>cc<A> => accccc
   Yes, we can generate the form from the grammar.

2. (a)
   <integer> ::=[<sign>]<unsigned>
   <unsigned> ::={<digits>}<digits>
   <digits> ::={<digit>}<digit>
   <digit> ::= 0 | 1 | … | 9
   <sign> ::= + | -
   (b)



3. (a) Language={$a^n b^n$ | n>=1}
   (b) Language={$a^n b^m c^k$ | n>=1,m>=1,k>=1}
   (c) Language={$0^n 1^m$ | n>=m+1 or $0^n 1^m$ | m>=2n+1}

4.(a)
String: if true then if true then other else other.



(b)
<stmt>::=<matchingStmt> | <unmStmt>
<matchingStmt>::=if <expr> then <matchingStmt>else<matchingStmt> | other
<unmStmt>::=if<expr> then <stmt> | if<expr>then<matchingStmt> else <unmStmt>
<expr>::=true | false

5. (a) BNF:
   <exp>::=<A><exp> | <A>
   <A>::= a | ε
   EBNF:
   <exp>::={<A>}
   <A>::= a | ε

(b) BNF:

<exp>::=<A><exp> | a

<A>::= a, |ε

EBNF:

<exp>::={<A>}a

<A>::= a, |ε