Name: Tyson-Tien Nguyen Student ID: 921031514

ECS 140A: Summer 2025
Homework Assignment 1

Due Date: No later than Thursday, July 3, 10:00pm PDT
This assignment asks you to prepare written answers to questions on BNF and EBNF. Make sure your written work is very readable. We will not grade what we cannot read. Make sure you follow the policy regarding the use of outside sources such as ChatGPT. **Make sure your answers are contained in the answer box corresponding to each question. Do not submit any other pages or parts of your answers outside the answer boxes. (And please forgive the odd placement of some boxes...Microsoft Word is not my friend.) You must only submit these assignment sheets. After you are done, scan them into a PDF file and turn it in via Gradescope by the due date above.**

1. (15 pts) Consider the following BNF grammar:

```
<S> ::= a <S> c <B> | <A> | b
<A> ::= c <A> | c
<B> ::= d | <A>
```

For each of the strings below, indicate whether or not the string can be derived from the grammar ("yes" or "no"). If the string can be derived from the grammar, provide a left-most derivation that shows all derivation steps. Otherwise, simply indicate “no.”

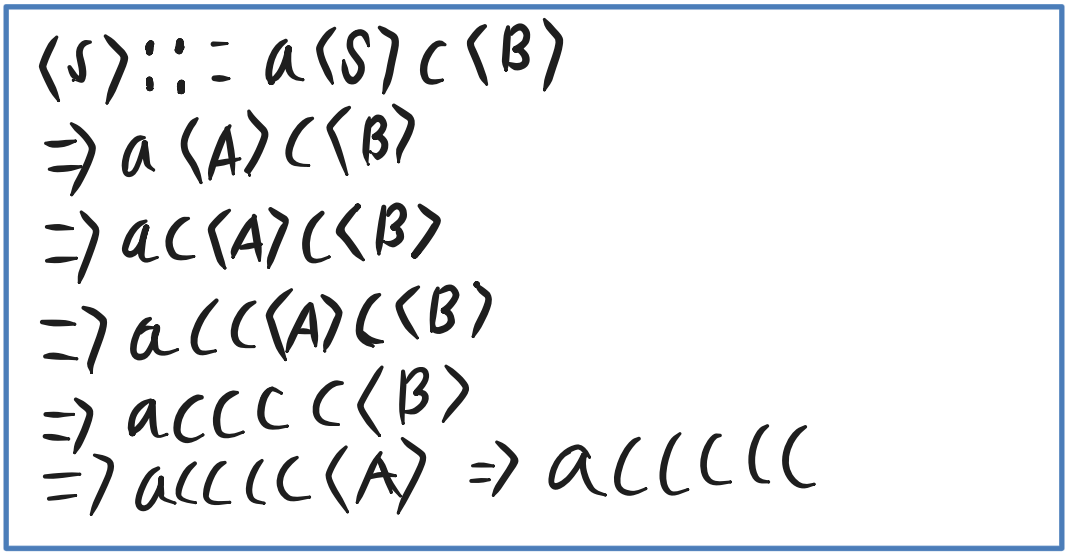
(a) (5 pts) `aabccd`

NO.

(b) (5 pts) `accbcc`

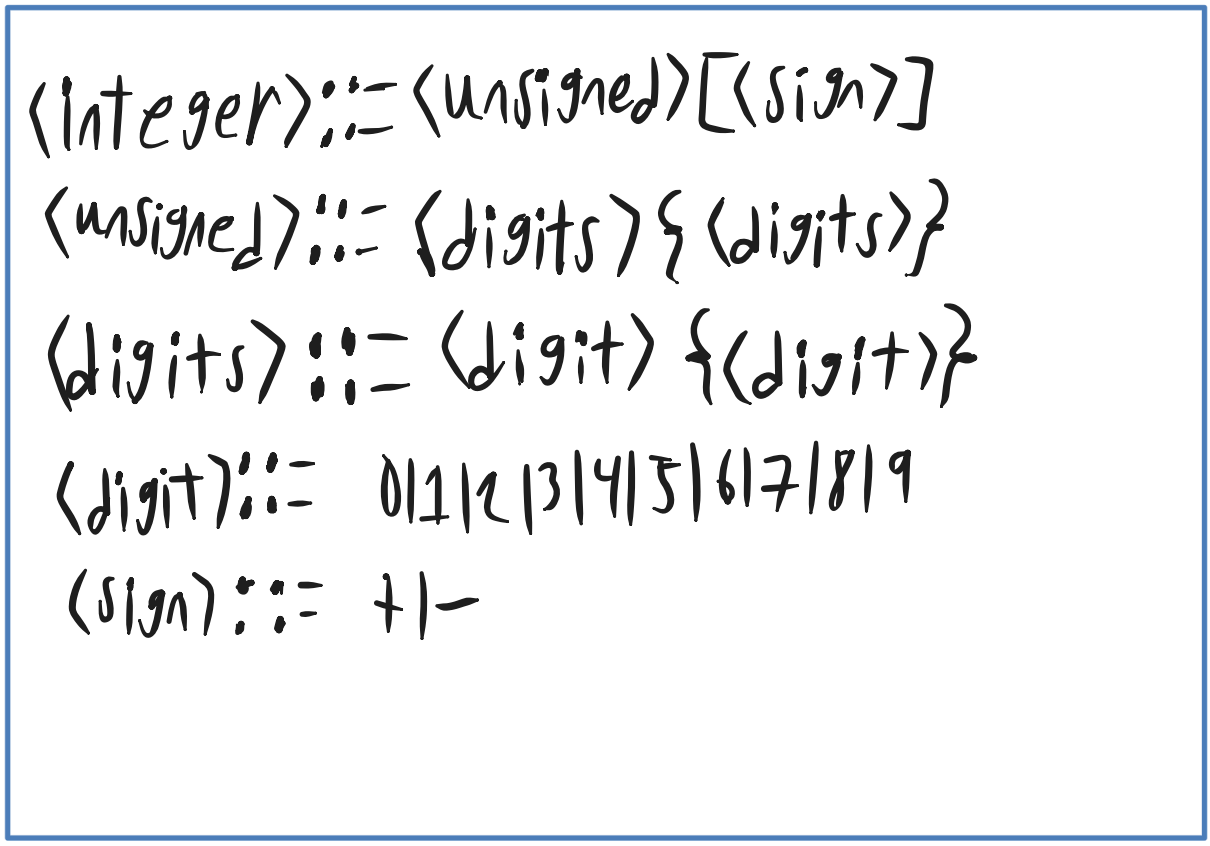
NO.

(c) (5 pts) `accccc`



2. (20 pts) Convert the following BNF grammar into EBNF.

```
<integer> ::= <unsigned> | <sign> <unsigned>
<unsigned>::= <digits> | <unsigned><digits>
<digits>  ::= <digits><digit> | <digit>
<digit>   ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<sign>    ::= + | -
```



3. (25 pts) What language is generated by each of the BNF grammars below (assuming that <S> is the start symbol)? You may describe the language in plain English, math notations, or a combination of both.

(a) (5 pts)

```
<S> ::= ab | a <S> b
```

$a^n b^n$, $n >= 1$

(b) (10 pts)

```
<S> ::= <A> <B> <C>
<A> ::= a <A> | a
<B> ::= b <B> | b
<C> ::= c <C> | c
```

$a^n b^m c^p$, $n >= 1$, $m >= 1$, and $p >= 1$

(c) (10 pts)

```
<S>  ::= <x> | <y>
<x>  ::= 0 <x> 1 | <x1>
<x1> ::= 0 <x1> | 0
<y>  ::= 0 <y> 1 1 | <y1>
<y1> ::= <y1> 1 | 1
```

$\langle S \rangle ::= 0^k 1^m \mid 0^a 1^c$

$k > m >= 0$

$k = m + n$

$n >= 1$

$c = 2a + b$

$b >= 1$

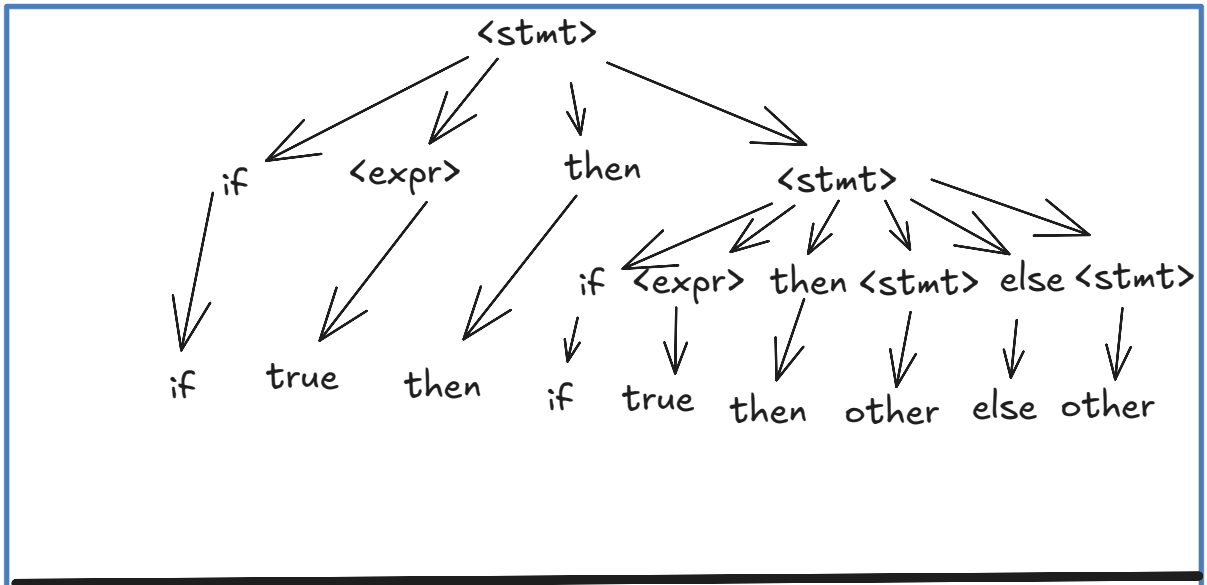
$a >= 0$

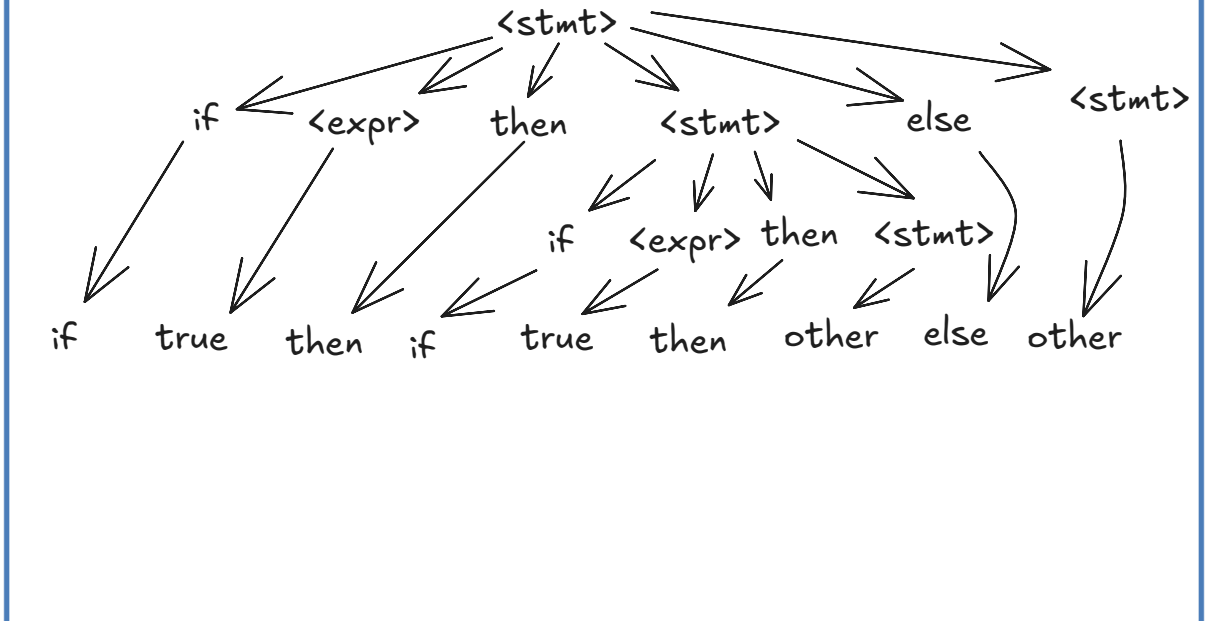
4. (20 pts) Given the following grammar:

```
<stmt> ::= if <expr> then <stmt>
         | if <expr> then <stmt> else <stmt>
         | other
<expr> ::= true | false
```

where `other` is a terminal that stands for any other kind of statement.

(a) (10 pts) This grammar is ambiguous. Give a string having two different parse trees and draw the parse trees.





(b) (10 pts) If we adopt the disambiguating rule (used in most languages) "match each `else` with the closest previous unmatched `then`," write an equivalent, un-ambiguous grammar.

```
<stmt> ::= <enclosed_stmt> | <open_stmt>

<enclosed_stmt> ::= other | if <expr> then <enclosed_stmt>
else <enclosed_stmt>

<open_stmt> ::= if <expr> then <enclosed_stmt>

<expr> ::= true | false
```

5. (20 pts) Give a BNF and an EBNF for each of the languages below.

(a) (10 pts) The set of all strings consisting of zero or more **a**'s.

```
<string> ::= <string> a | empty - BNF

<string> ::= {a} - EBNF
```

(b) (10 pts) The set of all strings consisting of one or more **a**'s, where there is a comma in between each **a** and the following **a**. Note that there is no comma before the first **a** or after the last **a**.

<string> ::= a, <string> | a - BNF

<string> ::= a {,a} - EBNF