#3) Michael Hug

<assign> -> <I D> = <expr>

<ID> => A | B | C

<expr> => <expr> * <term> | <term>

<term> -> <facter> + <term> | <factor>

<facter> -> (<expr>) | <I D>

#5) Used: docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html
reference

<assign> -> <urn><r> | <r>

<urn> -> !

<relational> -> >= | <=

<equality> -> == | !=

<AND> -> &&   these are ampersands

<OR> -> ||

<operand> -> A | B | C | D.... | <assign>   you can bang anything or go crazy with recursion

<r> -> <operand><relational><operand> | <e>

<e> -> <operad><equality><operand> | <A>

<A> -> <operand><AND><operand> | <o>

<o> -> <operand><OR><operand> | <operand> | <assign>

these <operand>'s can be complete mathematical expressions

Michael Hug

7) D) A = B*(C*(A+B))

<Assign> -> <ID> = <expr>

-> A = <expr>

-> A = <term>

-> A = <term> * <factor>

-> A = <factor> * <factor>

-> A = <ID> * <factor>

-> A = B * <factor>

-> A = B*(<expr>)

-> A = B*(<term>)

-> A = B*(<term> * <factor>)

-> A = B*(<factor> * <factor>)

-> A = B*(<ID> * <factor>)

-> A = B*(C * <factor>)

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

-> A = B*(C*(<expr> + <term>))

-> A = B*(C*(<term> + <term>))

-> A = B*(C*(<facter> + <term>))

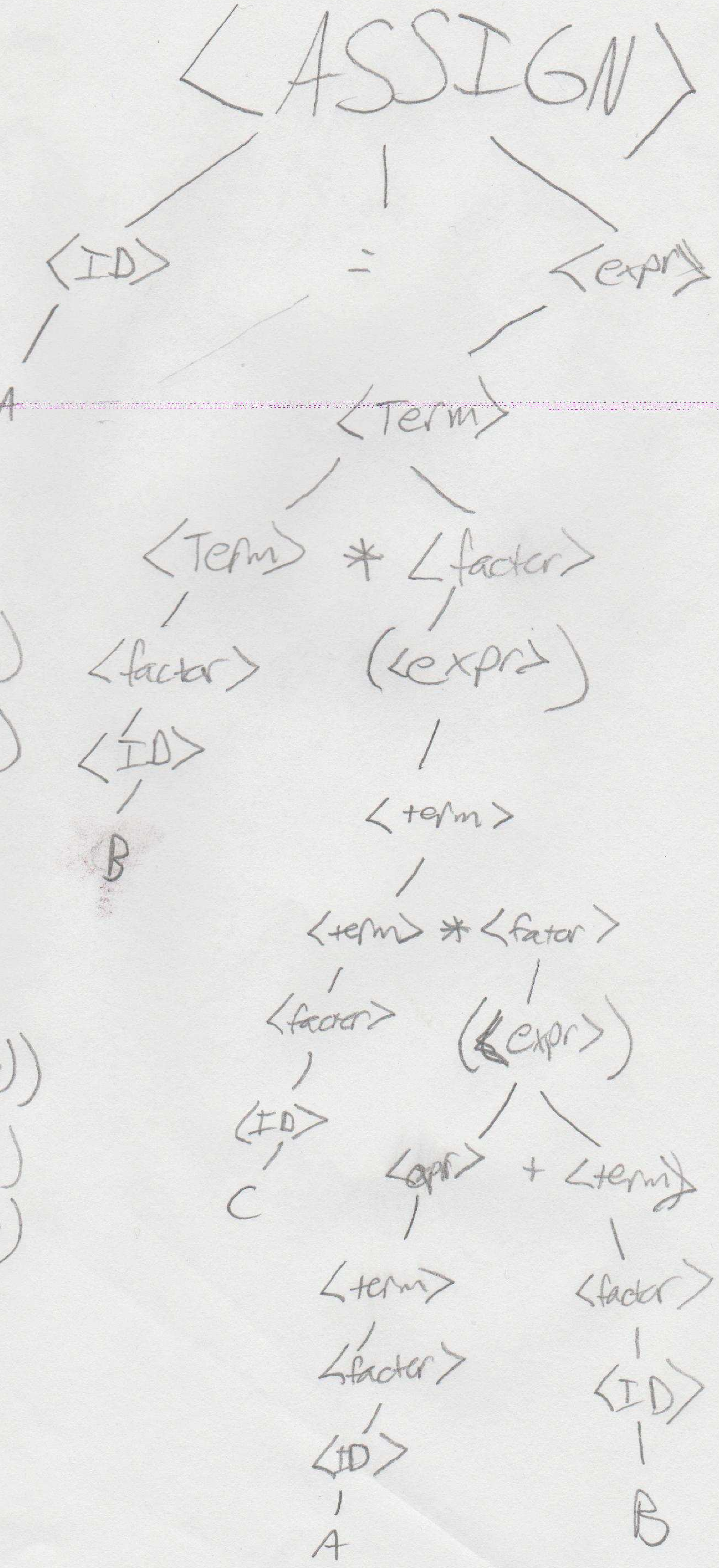
-> A = B*(C*(<ID> + <term>))

-> A = B*(C*(A + <term>))

-> A = B*(C*(A + <factor>))

-> A = B*(C*(A + <ID>))

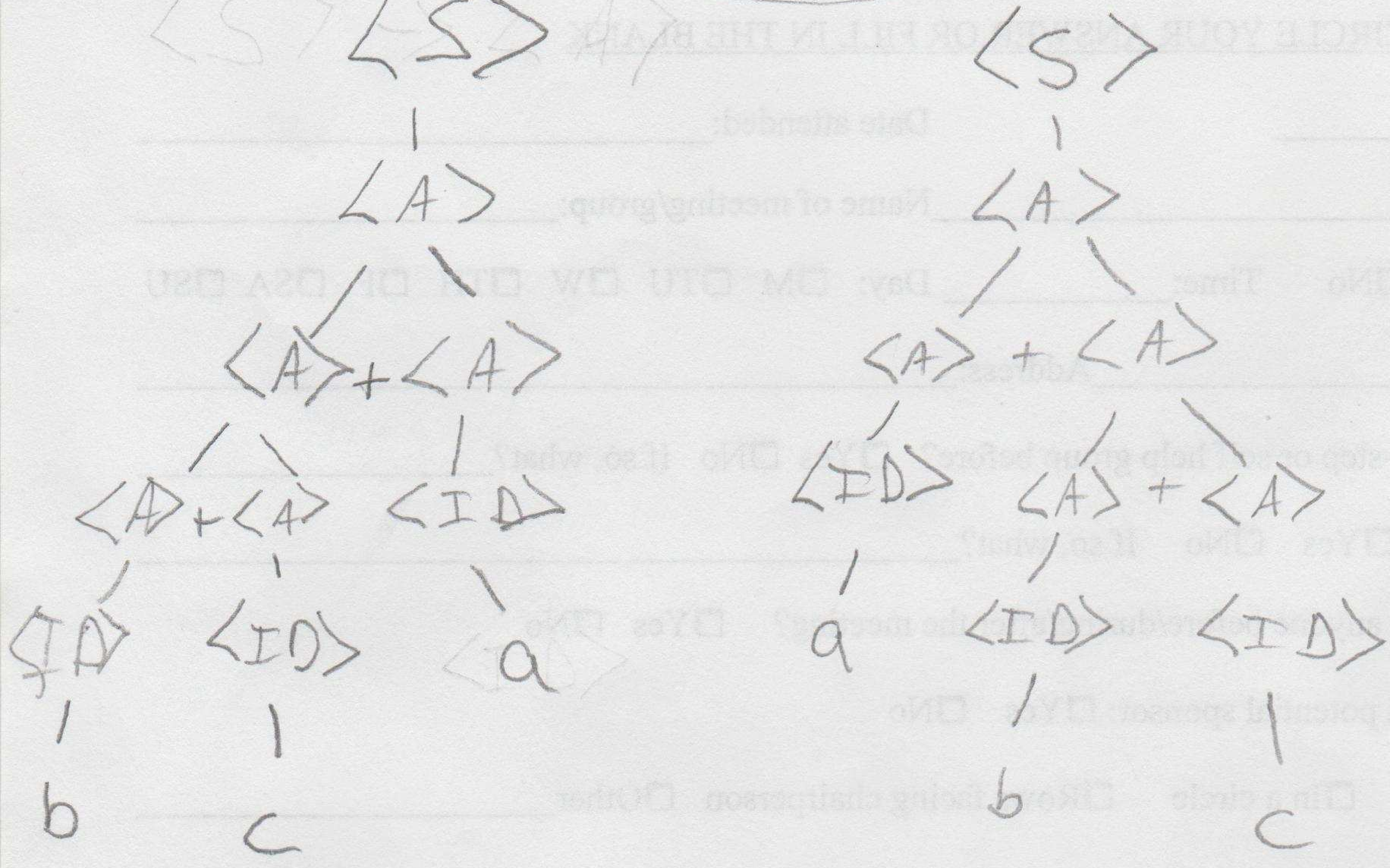
-> A = B*(C*(A + B))



8)

Michael Hug

Prove ambiguity: both legal



$b + c + a \neq a + b + c$

Consider the case $a = -1$; $b = 1$, $c =$ max int

we basically lose asscioativity

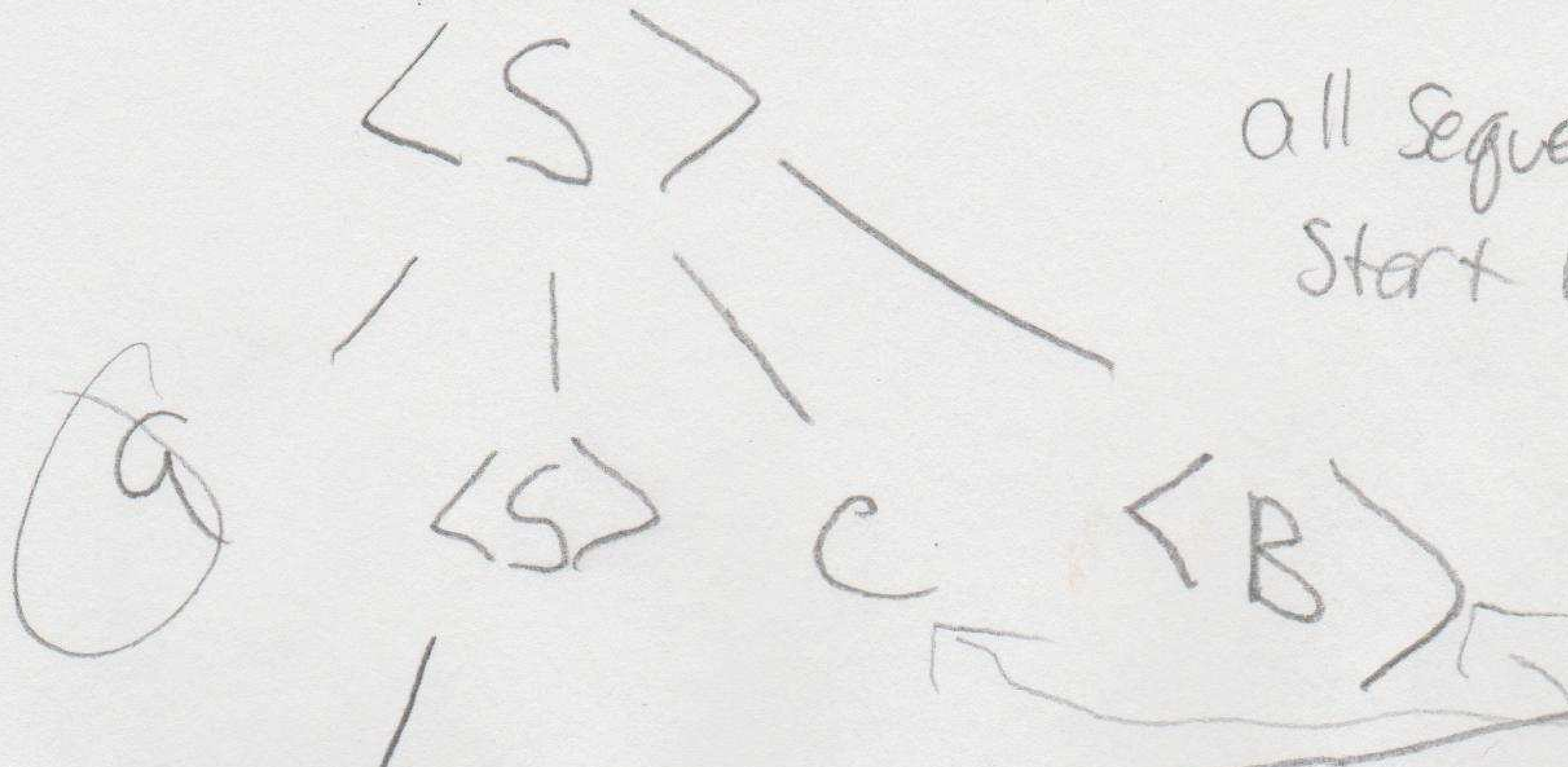
#10) Michael Hug

One or more a

followed by one or more b

followed by one or more c

#12)
Michael
Hug



all Sequences given as options Start with a...,

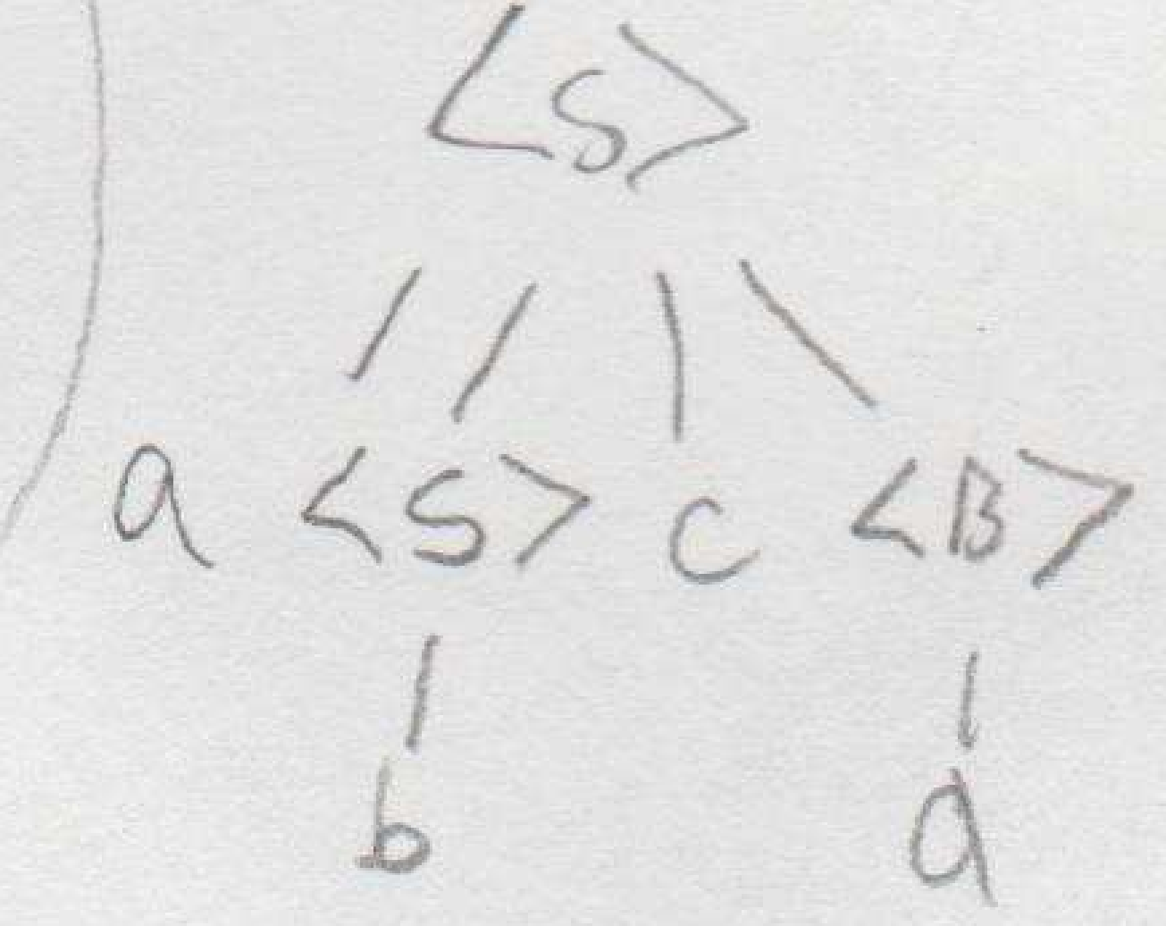
Only 1 ab... sqence

and it goes abcd so that is legal

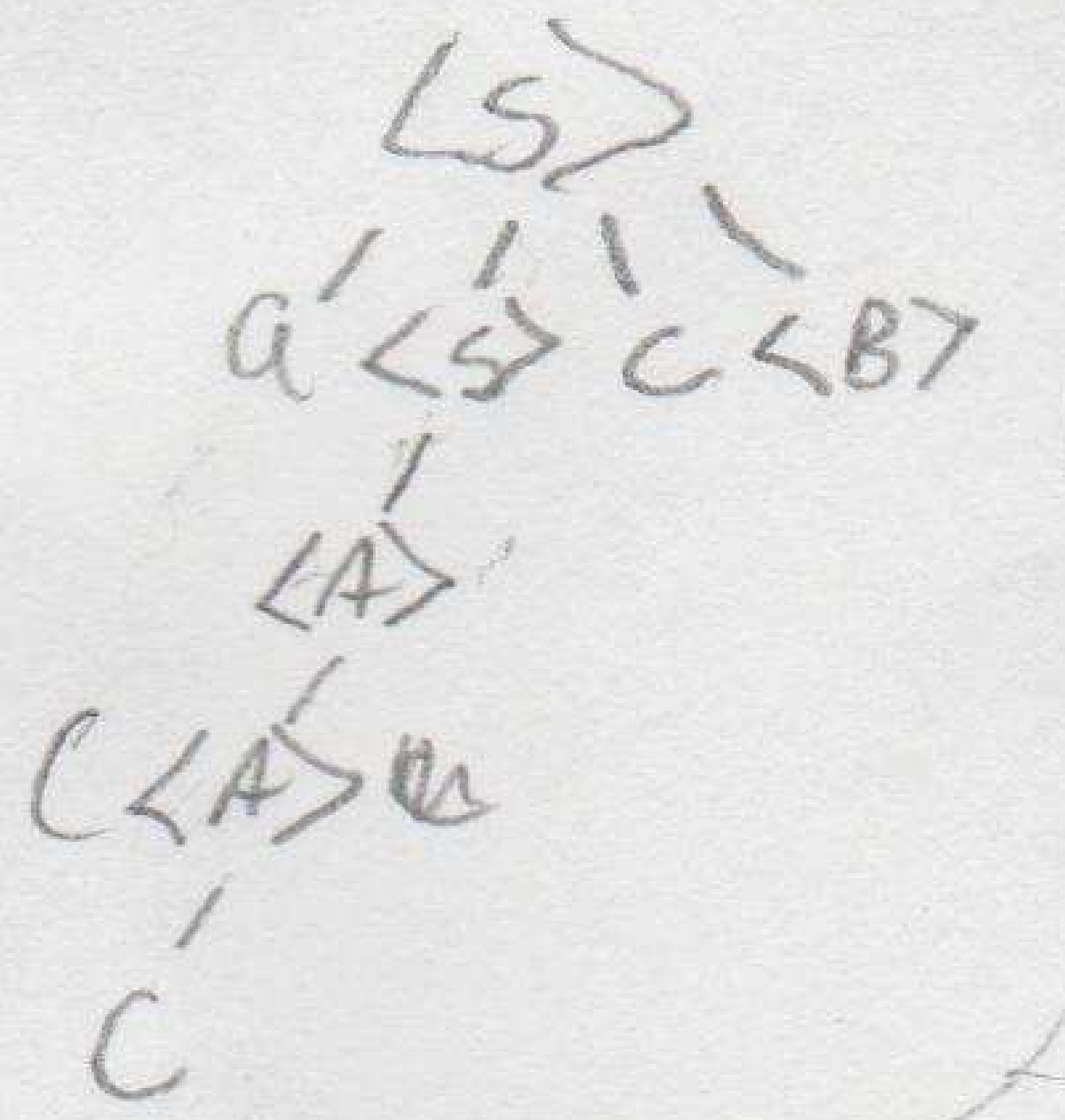
no aa sequences... —> ~~a <S>~~ c <B> || <A> || ~~b~~ <— no Seqences {a b}

<A> leads us into a repeating c situation that will terminate after n c's

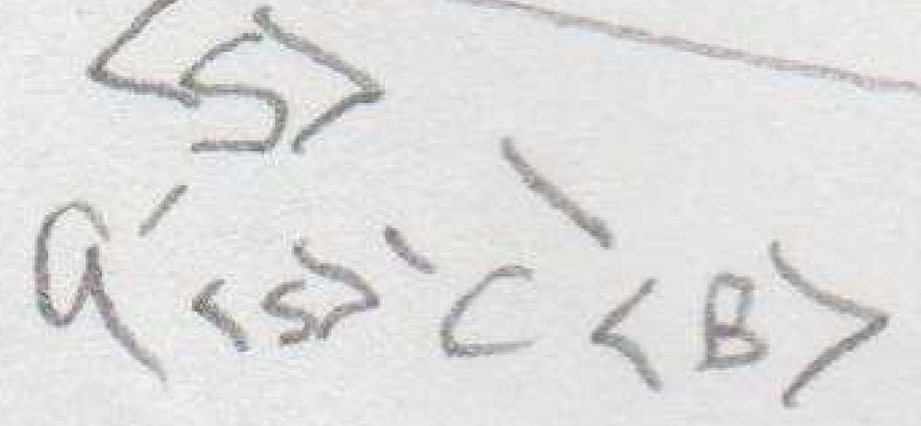
abcd) true



acccbd) false



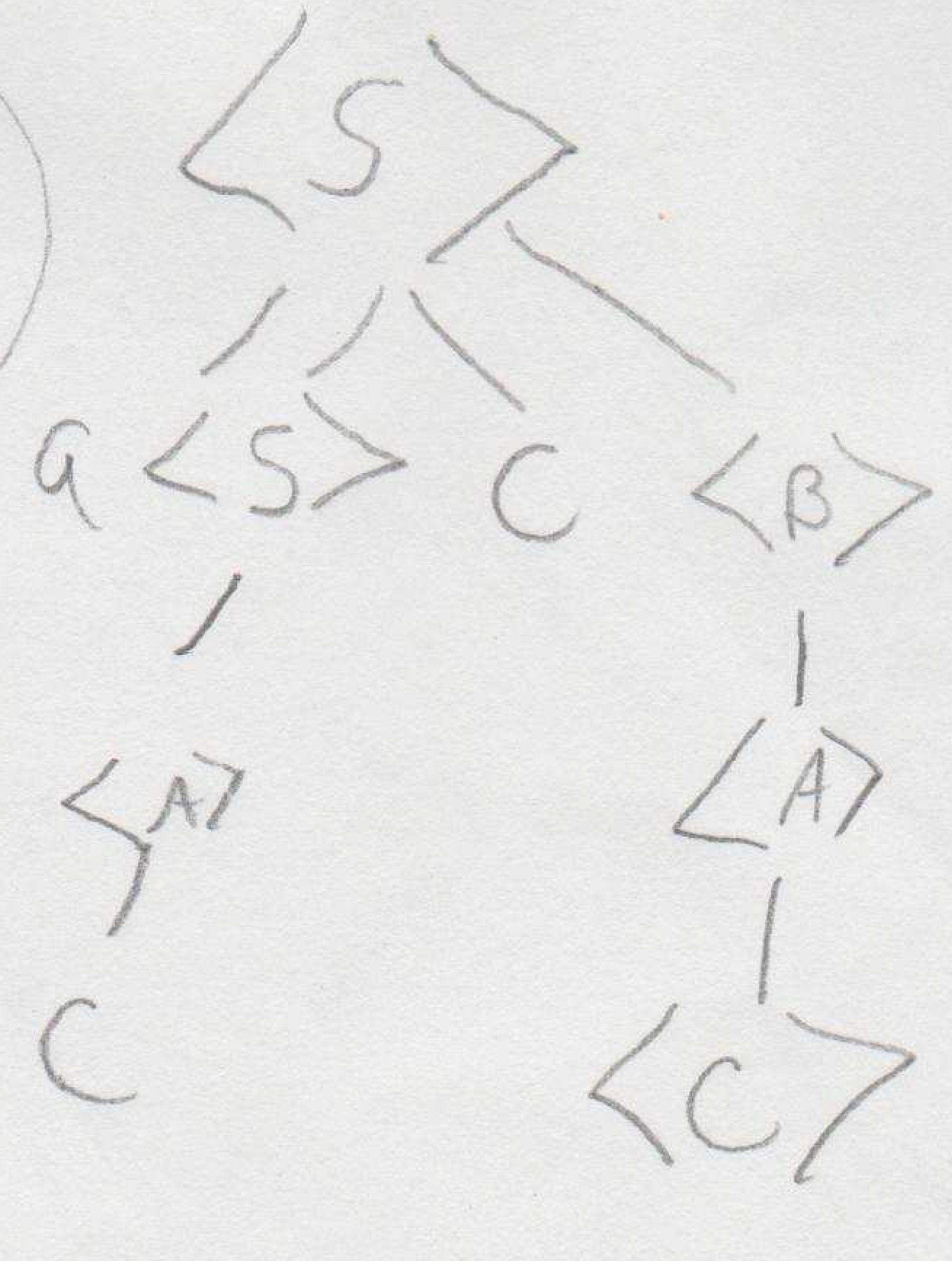
a cccbcc false



The <B> must go to either {d} or $\{\underbrace{c\ldots}_{n \text{ number of } c\text{'s}}\}$

acd) false

accc) true



#13) Micheal Hug

$$\langle s \rangle \rightarrow ab \mid a\langle s \rangle b$$

Only give the options to generate ab or generate an ab to the center of ~~array~~ sequence

#3) Michael Hug

```
<assign> -> <I D> = <expr>

<ID> => A | B | C
<expr> => <expr> * <term> | <term>
<term> -> <facter> + <term> | <factor>
<facter> -> ( <expr> ) | <I D>
```

#5) Used: docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html
reference

```
<assign> -> <urn> <r> | <r>
<urn> -> !
<relational> -> >= | <=
<equality> -> == | !=
<AND> -> &&                                   these are ampersands
<OR> -> ||
                                              you can bang anything
<operand> -> A | B | C | D . . . | <assign>   or go crazy with recursion
<r> -> <operand><relational><operand> | <e>
<e> -> <operand><equality><operand> | <A>
<A> -> <operand><AND><operand> | <o>
<o> -> <operand><OR><operand> | <operand> | <assign>
```

these <operand>'s can be complete mathematical expressions