

#3) Michael Hug

$\langle \text{assign} \rangle \rightarrow \langle \text{ID} \rangle = \langle \text{expr} \rangle$

$\langle \text{ID} \rangle \Rightarrow A | B | C$

$\langle \text{expr} \rangle \Rightarrow \langle \text{expr} \rangle * \langle \text{term} \rangle | \langle \text{term} \rangle$

$\langle \text{term} \rangle \Rightarrow \langle \text{factor} \rangle + \langle \text{term} \rangle | \langle \text{factor} \rangle$

$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) | \langle \text{ID} \rangle$

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

$\langle \text{assign} \rangle \rightarrow \langle \text{var} \rangle \langle \text{assign} \rangle | \langle \text{r} \rangle$

$\langle \text{var} \rangle \rightarrow !$

$\langle \text{relational} \rangle \rightarrow > = | < =$

$\langle \text{equality} \rangle \rightarrow = = | ! =$

$\langle \text{AND} \rangle \rightarrow \& \&$

these are ampersands

$\langle \text{OR} \rangle \rightarrow ||$

$\langle \text{operand} \rangle \rightarrow A | B | C | D \dots | \langle \text{assign} \rangle$

you can bang anything
or go crazy with recursion

$\langle \text{r} \rangle \rightarrow \langle \text{operand} \rangle \langle \text{relational} \rangle \langle \text{operand} \rangle | \langle \text{e} \rangle$

$\langle \text{e} \rangle \rightarrow \langle \text{operand} \rangle \langle \text{equality} \rangle \langle \text{operand} \rangle | \langle \text{A} \rangle$

$\langle \text{A} \rangle \rightarrow \langle \text{operand} \rangle \langle \text{AND} \rangle \langle \text{operand} \rangle | \langle \text{O} \rangle$

$\langle \text{O} \rangle \rightarrow \langle \text{operand} \rangle \langle \text{OR} \rangle \langle \text{operand} \rangle | \langle \text{operand} \rangle | \langle \text{assign} \rangle$

these $\langle \text{operand} \rangle$'s can be complete mathematical expressions

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

$\langle \text{Assign} \rangle \rightarrow \langle \text{ID} \rangle = \langle \text{expr} \rangle$

$\rightarrow A = \langle \text{expr} \rangle$

$\rightarrow A = \langle \text{term} \rangle$

$\rightarrow A = \langle \text{term} \rangle * \langle \text{factor} \rangle$

$\rightarrow A = \langle \text{factor} \rangle * \langle \text{factor} \rangle$

$\rightarrow A = \langle \text{ID} \rangle * \langle \text{factor} \rangle$

$\rightarrow A = B * \langle \text{factor} \rangle$

$\rightarrow A = B * (\langle \text{expr} \rangle)$

$\rightarrow A = B * (\langle \text{term} \rangle)$

$\rightarrow A = B * (\langle \text{term} \rangle * \langle \text{factor} \rangle)$

$\rightarrow A = B * (\langle \text{factor} \rangle * \langle \text{factor} \rangle)$

$\rightarrow A = B * (\langle \text{ID} \rangle * \langle \text{factor} \rangle)$

$\rightarrow A = B * (C * \langle \text{factor} \rangle)$

$\rightarrow A = B * (C * (\langle \text{expr} \rangle))$

$\rightarrow A = B * (C * (\langle \text{expr} \rangle + \langle \text{term} \rangle))$

$\rightarrow A = B * (C * (\langle \text{term} \rangle + \langle \text{term} \rangle))$

$\rightarrow A = B * (C * (\langle \text{factor} \rangle + \langle \text{term} \rangle))$

$\rightarrow A = B * (C * (\langle \text{ID} \rangle + \langle \text{term} \rangle))$

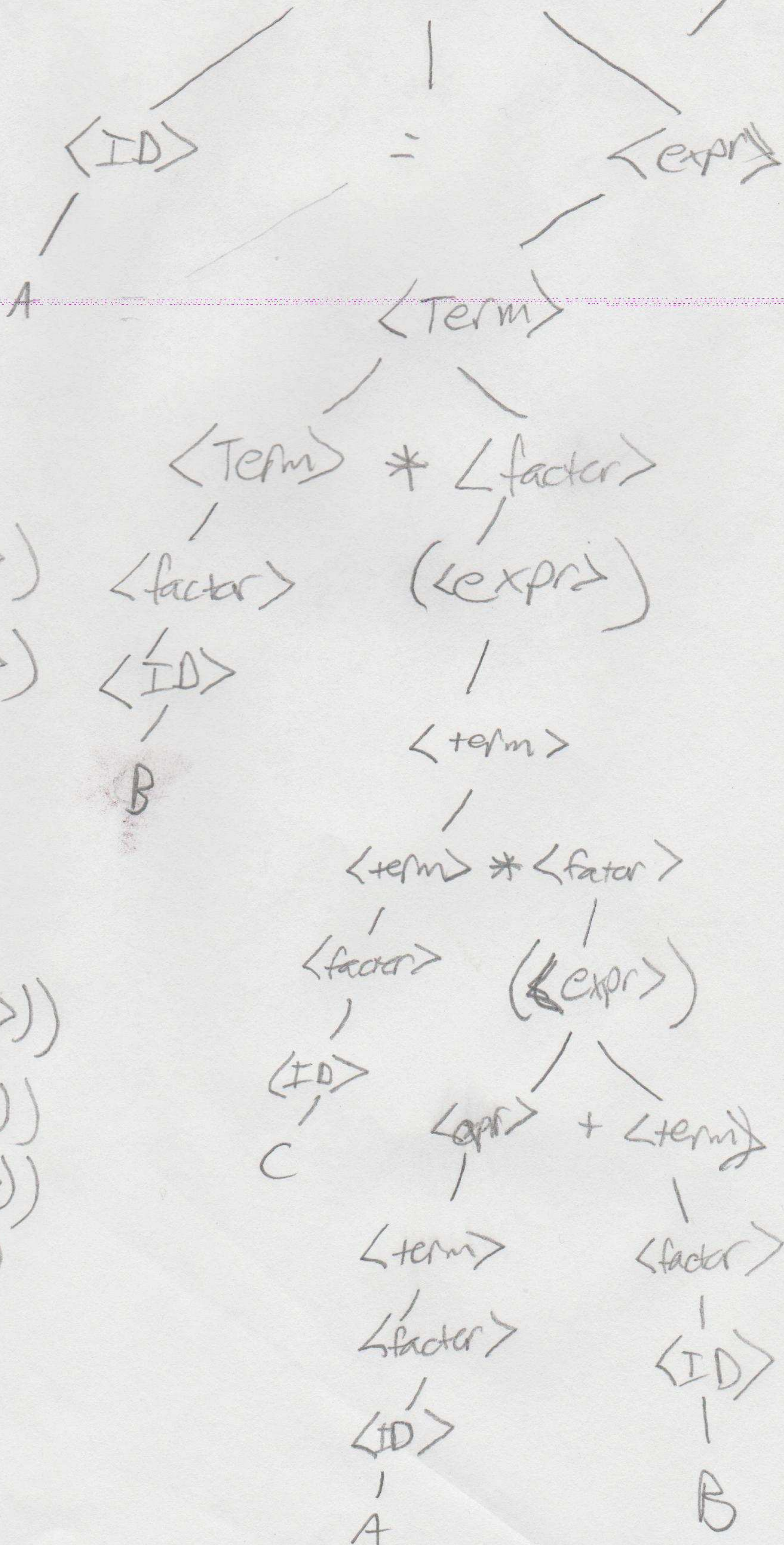
$\rightarrow A = B * (C * (A + \langle \text{term} \rangle))$

$\rightarrow A = B * (C * (A + \langle \text{factor} \rangle))$

$\rightarrow A = B * (C * (A + \langle \text{ID} \rangle))$

$\rightarrow A = B * (C * (A + B))$

$\langle \text{ASSIGN} \rangle$

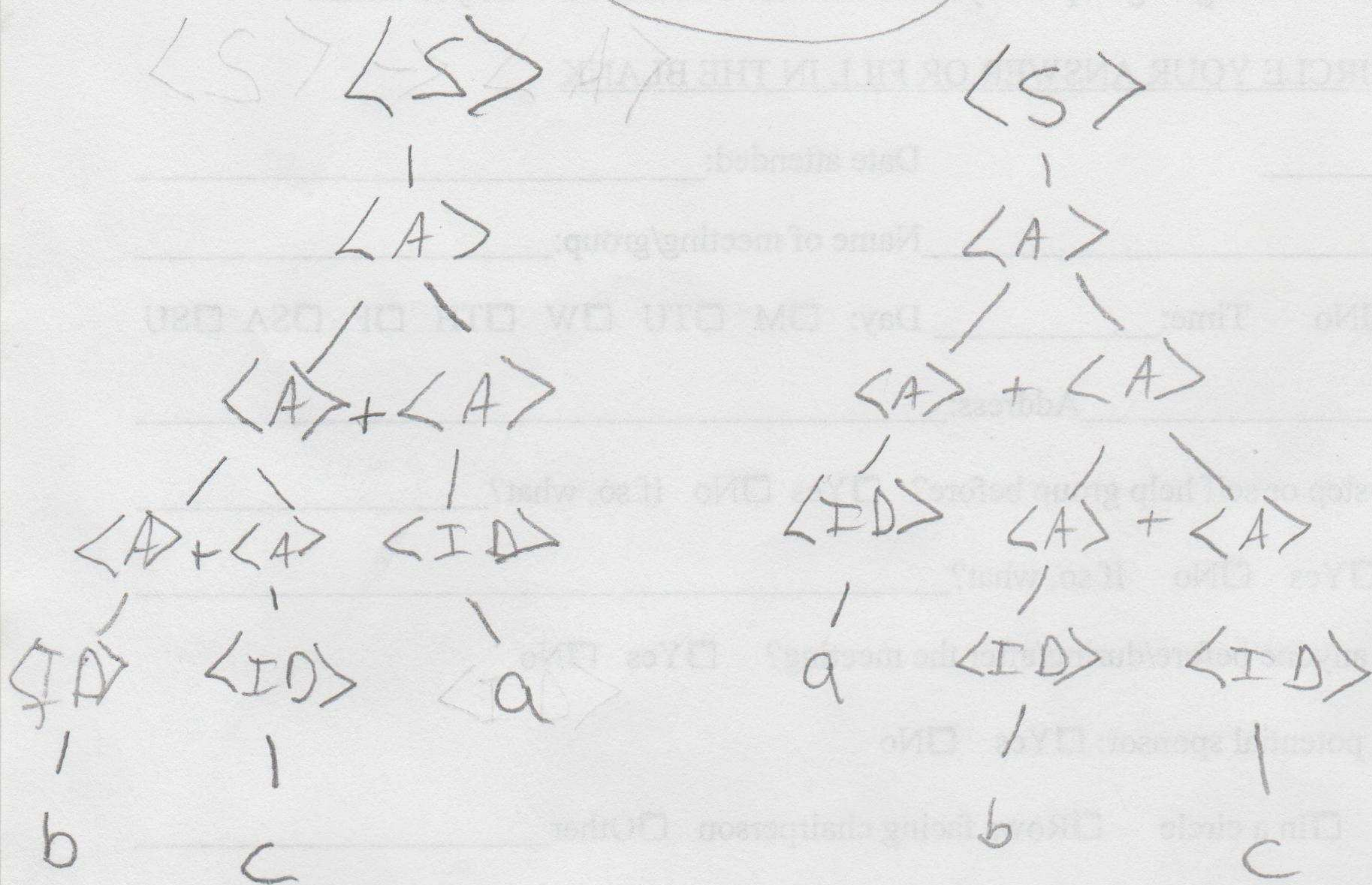


8)

Prove ambiguity:

both legal

Michael Hug



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

Consider the case $a = -1$; $b = 1$; $c = \text{max int}$

we basically lose associativity

#10) Michael Hug

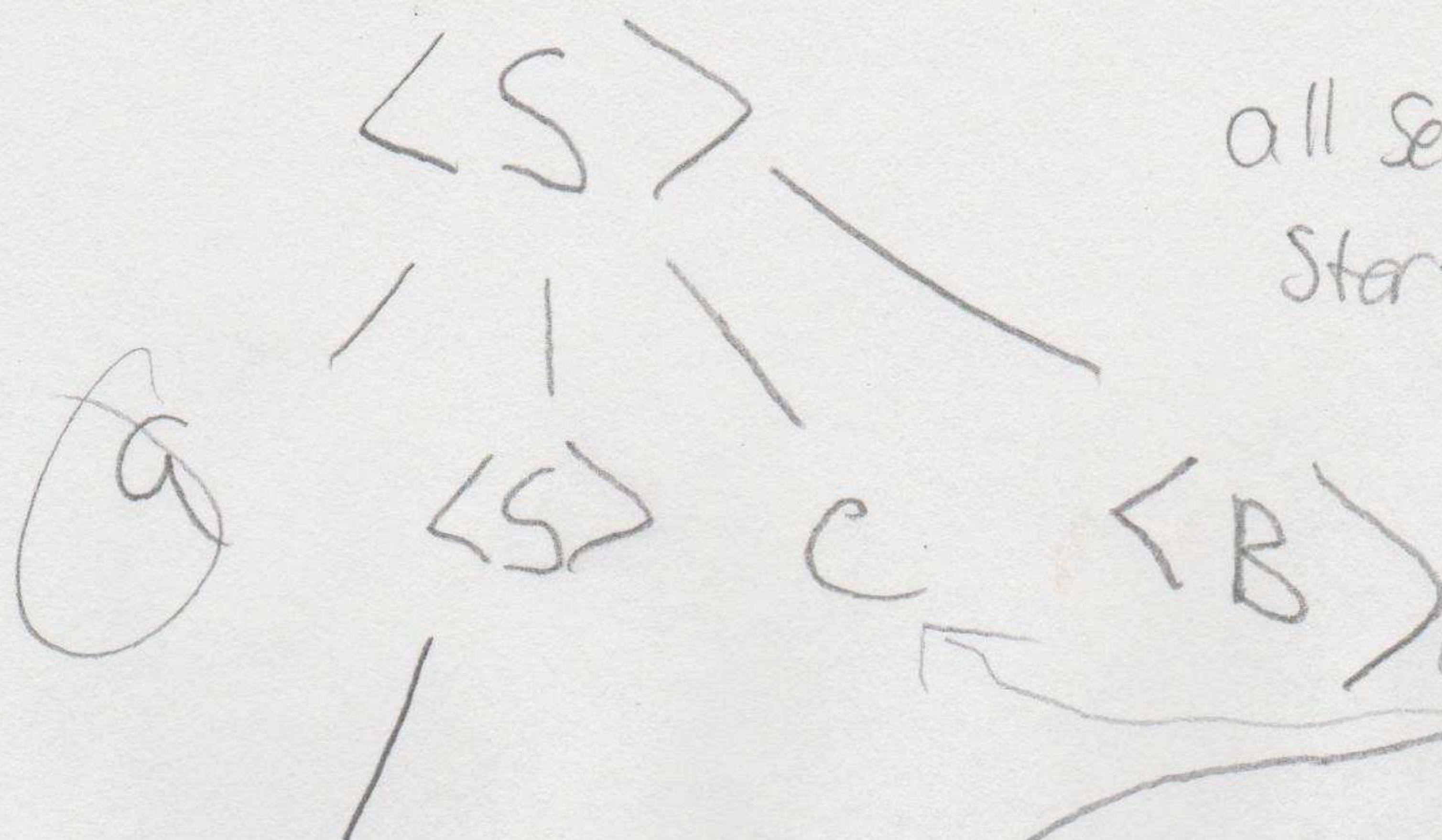
One or more a

followed by one or more b

followed by one or more c

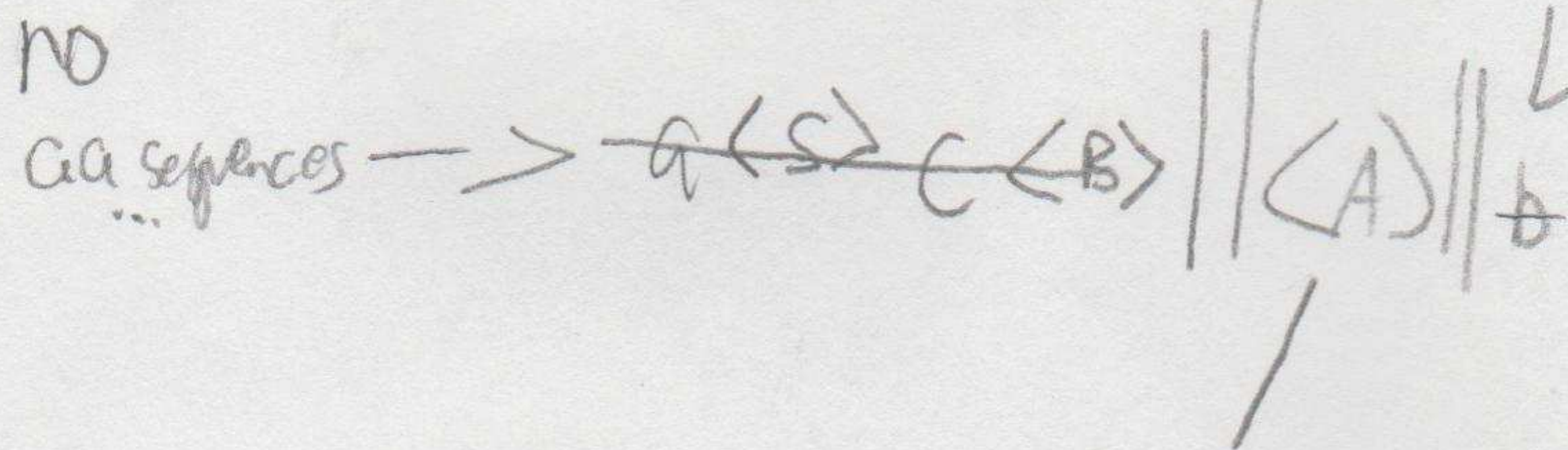
#12)
Michael
Hug

all sequences given as options
Start with a...

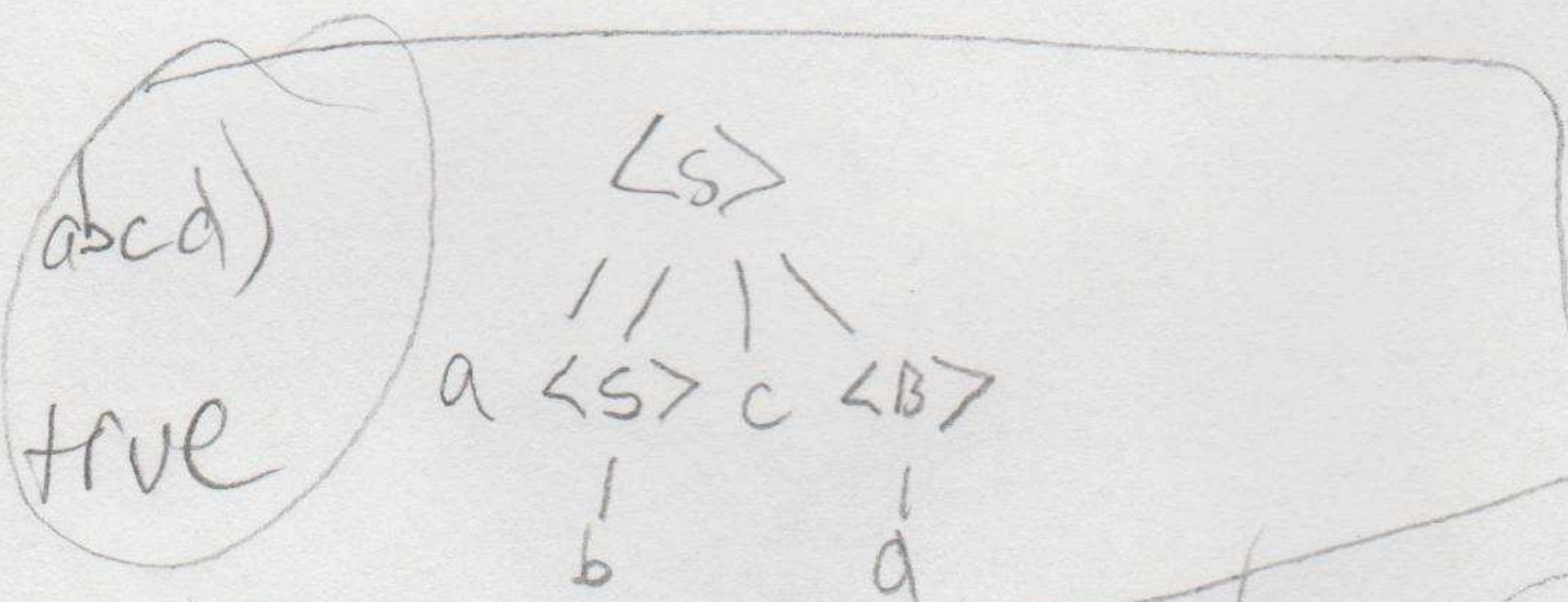


only 1 ab sequence

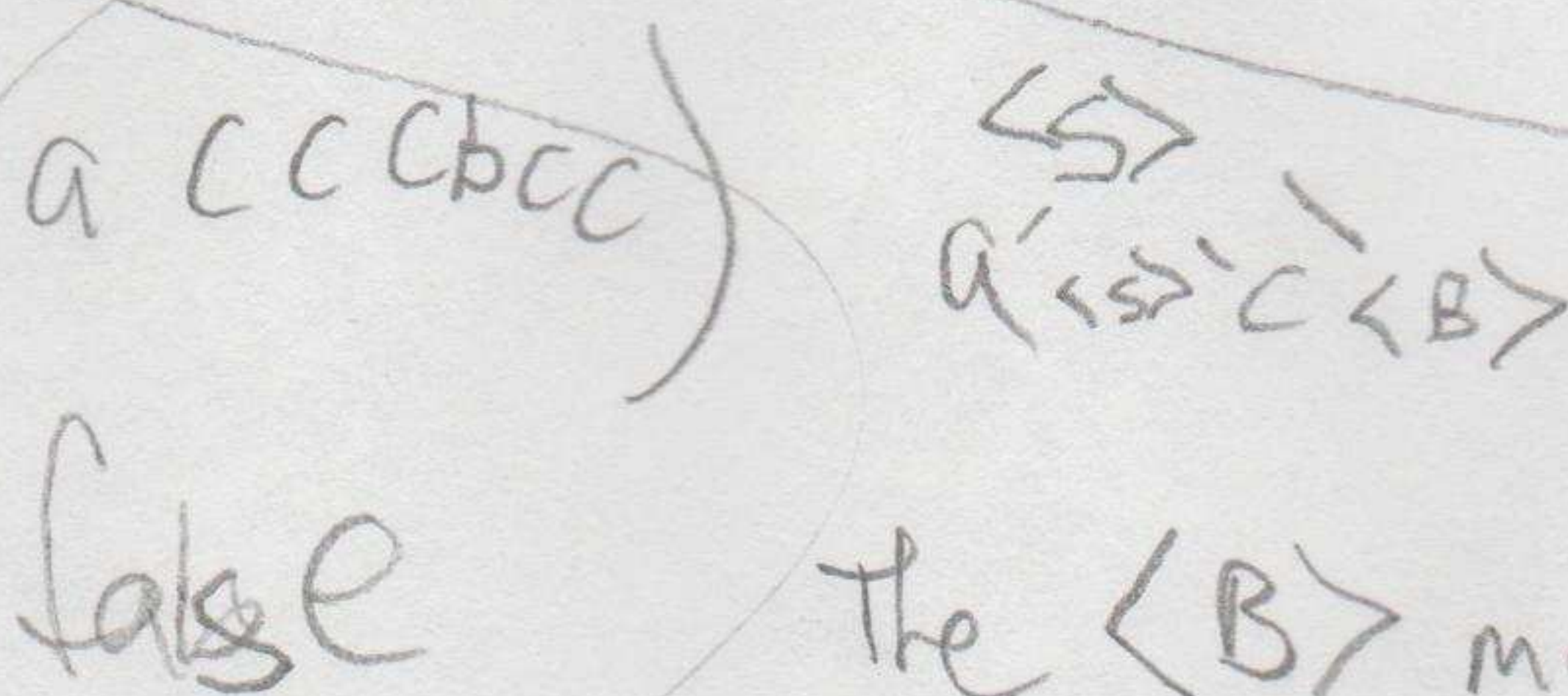
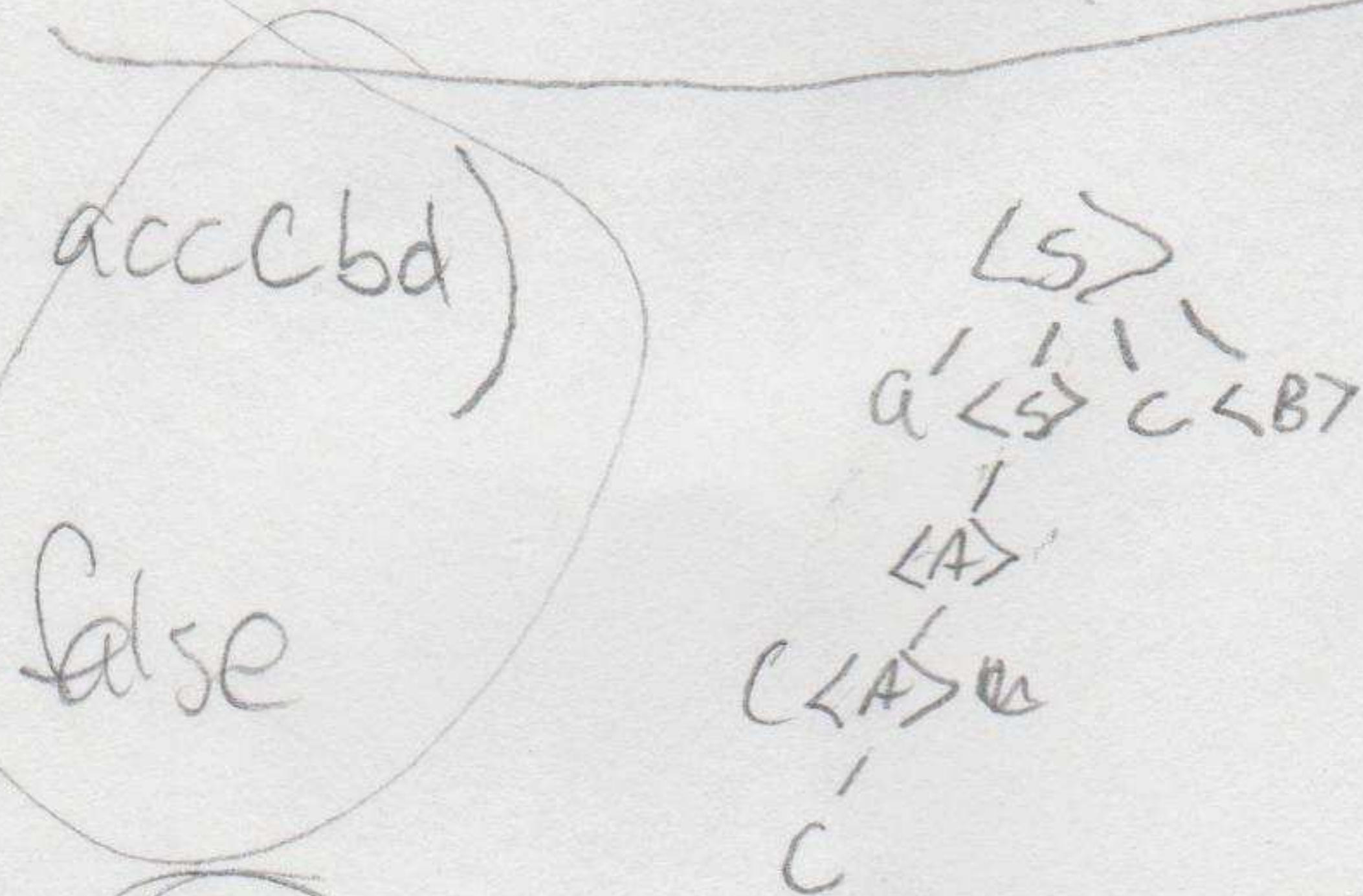
and it goes abcd so that is legal



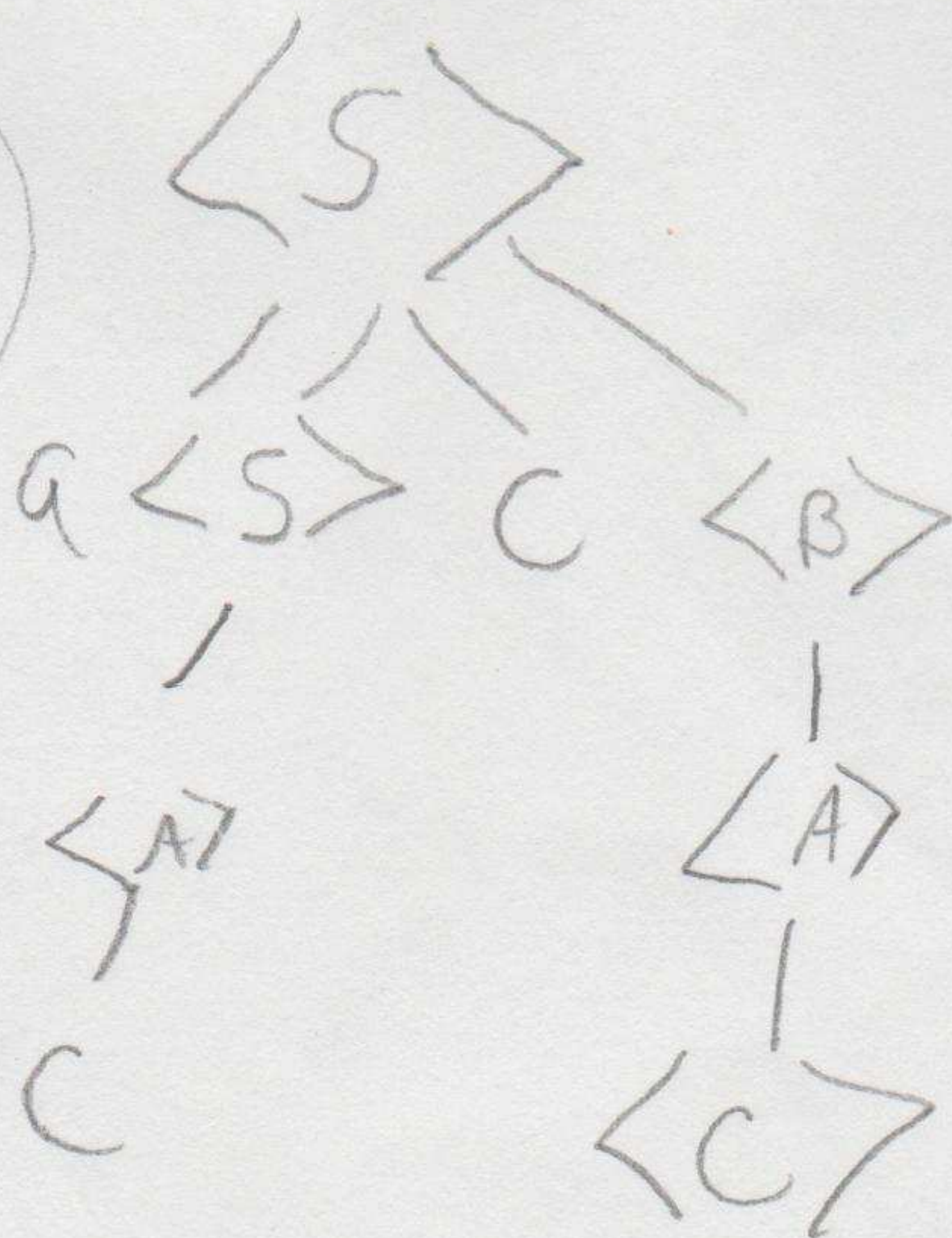
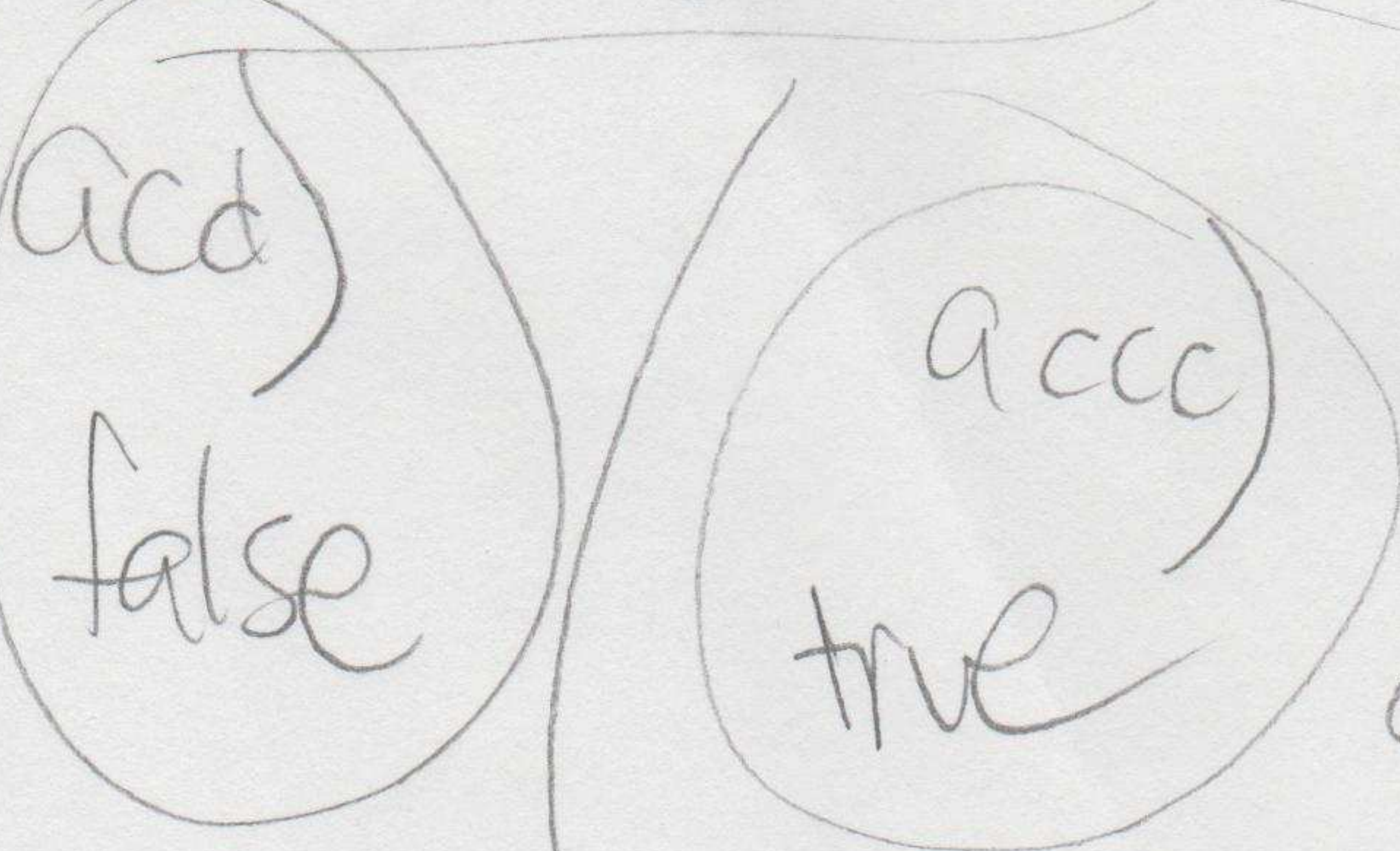
no sequences {a b}



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



The must go to either
{d} or {c...} n number of c's



#13) Michael Hug

$$\langle S \rangle \Rightarrow ab \mid a \langle S \rangle b$$

Only give the options to ^{drive} generate ab or generate
on ab to the center of ~~array~~ sequence

#3) Michael Hug

$\langle \text{assign} \rangle \rightarrow \langle \text{ID} \rangle = \langle \text{expr} \rangle$

$\langle \text{ID} \rangle \Rightarrow A | B | C$

$\langle \text{expr} \rangle \Rightarrow \langle \text{expr} \rangle * \langle \text{term} \rangle | \langle \text{term} \rangle$

$\langle \text{term} \rangle \Rightarrow \langle \text{factor} \rangle + \langle \text{term} \rangle | \langle \text{factor} \rangle$

$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) | \langle \text{ID} \rangle$

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

$\langle \text{assign} \rangle \rightarrow \langle \text{var} \rangle \langle \text{assign} \rangle | \langle \text{r} \rangle$

$\langle \text{var} \rangle \rightarrow !$

$\langle \text{relational} \rangle \rightarrow > = | < =$

$\langle \text{equality} \rangle \rightarrow = = | ! =$

$\langle \text{AND} \rangle \rightarrow \& \&$

these are ampersands

$\langle \text{OR} \rangle \rightarrow ||$

$\langle \text{operand} \rangle \rightarrow A | B | C | D \dots | \langle \text{assign} \rangle$

you can bang anything
or go crazy with recursion

$\langle \text{r} \rangle \rightarrow \langle \text{operand} \rangle \langle \text{relational} \rangle \langle \text{operand} \rangle | \langle \text{e} \rangle$

$\langle \text{e} \rangle \rightarrow \langle \text{operand} \rangle \langle \text{equality} \rangle \langle \text{operand} \rangle | \langle \text{A} \rangle$

$\langle \text{A} \rangle \rightarrow \langle \text{operand} \rangle \langle \text{AND} \rangle \langle \text{operand} \rangle | \langle \text{O} \rangle$

$\langle \text{O} \rangle \rightarrow \langle \text{operand} \rangle \langle \text{OR} \rangle \langle \text{operand} \rangle | \langle \text{operand} \rangle | \langle \text{assign} \rangle$

these $\langle \text{operand} \rangle$'s can be complete mathematical expressions