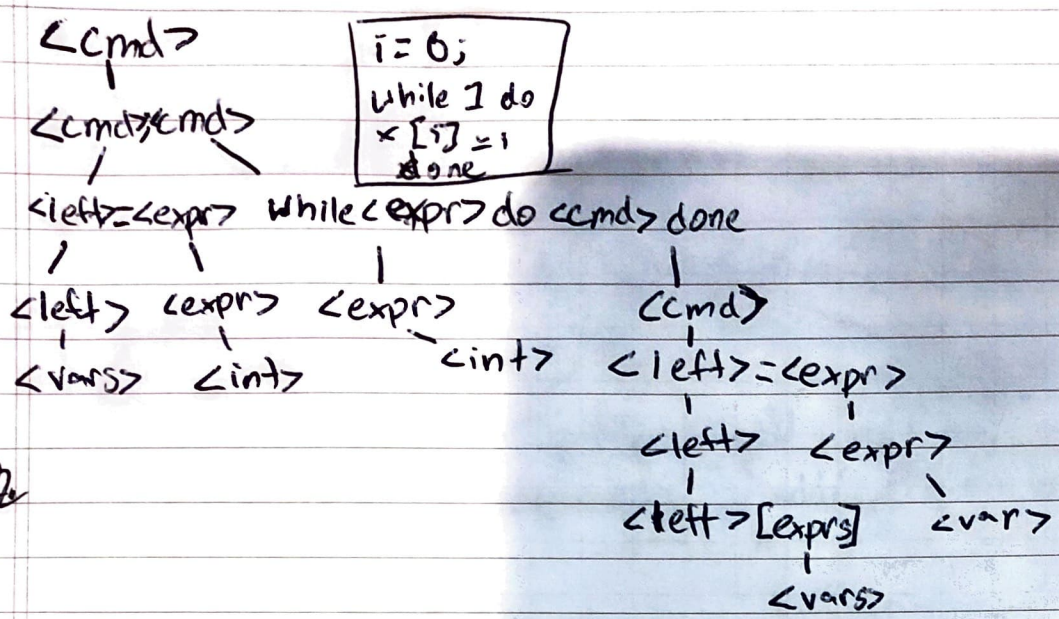
1. This grammar is unambiguous because you can't make more than one parsetree from one expression. No



2. $ab^*c^*\varepsilon$
3. $a^*bb^*c\varepsilon$
4. <A> :: a<B>
   | c<C>
   | ε
   <B> :: b<A>
   | b<C>
   <C> :: c<C>
   | ε

5, add(1,5) → 6
add(1, add(2,3)) → add(1,5)
add(1, add(2,3)) → 2 6

[illegible]

•
eq(3,3) → true
eq(3, add(2,1)) → eq(3,3)
eq(add(1,2)), add(2,1)) → eq(3, add(2,1))
eq(add(1,2)), add(2,1)) → true
eq(add(1,2)), 3) → (eq(3,3))
eq(add(1,2), add(2,1)) → eq(add(1,2), 3)

•
add(3, false) → error
add(add(1,2), false) → add(3, false)
add(add(1,2), eq(1,2)) → add(add(add(1,2), false))
add(3, eq(1,2)) → add(3, false)
add(add(1,2), eq(1,2)) → add(3, eq(1,2))
add(add(1,2), eq(1,2)) → error

6a) (add; quit/2::1::[]) → (quit/3;;[])
(push 2; add; quit/1::[]) → (add; quit/2::1::[])
(push 1; push 2, add; quit/[]) → (push 2, add; quit/1::[])
(push 1; push 2; add; quit/[])

b) (sub; quit/2::1::[]) → (quit/-1;:[])
(push 2; sub; quit/1::[]) → (sub; quit/2::1::[])
(push 1; push 2; sub, quit/[]) → (push 2; sub; quit/1::[])
(push 1; push 2; sub; quit/[])

c) (pop; quit/[]) → error
(if push 1 else pop and; quit/-1::[]) → (pop; quit/[])
(sub; if push 1 else pop end; quit/2::1::[]) → (if push 1 else pop end; quit/-1::[])
(push 2; sub; if push 1 else pop end; quit/1::[]) → (sub; if push 1 else pop end; quit/2::1::[])
(push 1; push 2; sub; if push 1 else pop end; quit/[]) → (push 2; sub; if push 1 else pop end; quit/1::[])
(push 1; push 2; sub; if push 1 else pop end; quit/[])