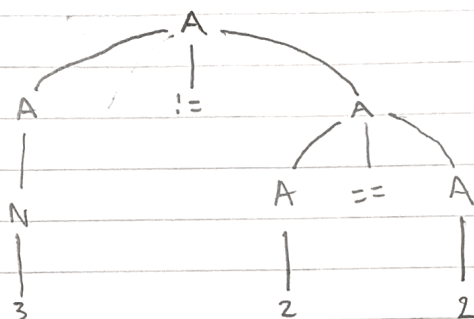


$$3 \neq 2 == 2$$

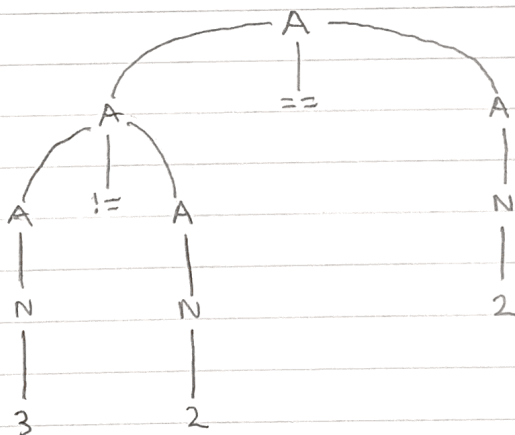
$$A \rightarrow A \neq A \rightarrow N \neq A \rightarrow 3 \neq A \rightarrow 3 \neq A == A \rightarrow 3 \neq N == A \rightarrow 3 \neq 2 == A$$

$$\hookrightarrow 3 \neq 2 == N \rightarrow 3 \neq 2 == 2$$



$$A \rightarrow A == A \rightarrow A \neq A == A \rightarrow N \neq A == A \rightarrow 3 \neq A == A$$

$$\hookrightarrow 3 \neq N == A \rightarrow 3 \neq 2 == A \rightarrow 3 \neq 2 == N \rightarrow 3 \neq 2 == 2$$



The problem with a grammar being ambiguous if we're trying to use it to represent a programming language is that it creates two or more different parse trees and when the compiler compiles the program, the compiler has a hard time figuring out which one of the parse trees is correct and makes the context work as intended. The different parse trees means different meanings thus different executable programs.