1. Given the following grammar, argue either that both the WithElse non-terminals in the WithElse production (highlighted in **bold**) are required to guarantee non-ambiguity or argue that one or both can be replaced with Stmt

Yes, I believe that both WithElse pieces are needed to guarantee non-ambiguity. I think this because otherwise the else statements couldn’t be guaranteed to be matched with the correct if statement. Using this grammar, you are required that if an if statement has an else attached to it, all nested if statements also have an else statement tied to them. I believe this eliminates any possible chances for ambiguity to occur.

1. Given the grammar below, argue either that you can or cannot generate all valid if statements, where “valid if statements” means all if statements you could generate in C without the use if curly braces.

Assuming that we aren’t talking about the fact that the only thing that can be done in these if statements are assignments, instead of other types of statements such as function calls or loops, then yes, I believe that one could generate all valid if statements. I think this because of the nested if statements all require an else tied to them as long as the parent if has an else.

Stmt -> if (Expr) then Stmt

-> if (Expr) then WithElse else Stmt

-> Assignment

WithElse -> if (Expr) then **WithElse** else **WithElse**

-> Assignment