Example for the proof

This is a small example to give an idea how to approach the proof part of assignment #1. How the FIRST and FOLLOW were obtained at this point is irrelevant. This is only to show you how to address the various types of rules. Again, single letters are terminals while lower case names represent non-terminal as you can see from the grammar.

block ::= B {statemt} E [D]
statemt ::= asignmt | ifstmt | while | inpout | block
asignmt ::= A ident ~ exprsn
ifstmt ::= I comprsn T block [L block]

block **FIRST(statemt) ∩ { E } = { A, I, W, R, O, B } ∩ { E }= Ø**
FOLLOW(block) ∩ { D } = { $, E, L, I, W } ∩ { D } = Ø

statemt **FIRST(assnmt), FIRST(ifstmt), FIRST(while), FIRST(read), FIRST(inpout), FIRST(block) all pairwise disjoint**

asignmt **no issue**

ifstmt **{ L } ∩ FOLLOW (ifstmt) = { L } ∩ { E, A, I, W, R, O, B }= Ø**

As you can see block has { } brackets, statemt has straightforward choices, asignmt has no issue as it does not contain any choice, and ifstmt has [ ] brackets.