The trace for me starts with murderer(\_5382) which I assume the \_5382 means that the variable for murderer is undeclared. It then calls hair(\_5382, brown) because it is the if of the murderer function. Hair(x, brown) has an if statement which is attire(x, pincenez) which calls attire(sir\_raymond, tattered\_cuffs) because it is the if of attire(mr\_woodley, pincenez). attire(sir\_raymond, tattered\_cuffs) then calls room(X, 16) because if something has tattered\_cuffs they stayed in room 16, which we find out sir\_raymond didn’t stay in room 16, but in room 10. Because it failed it backtracks back to attire(\_5382, pincenez) but this time calls attire(mr\_woodley, tattered\_cuffs) and then room(mr\_woodley, 16) which it fines that is true because mr\_woodley stayed in room 16. The program then backtracks to attire(sir\_raymond, pincenez) which leads to hair(sir\_raymond, brown) and then all the back to the start murderer(\_5382) which now \_5382 is set to sir\_raymond because he has brown hair and pincenez which all of these the murderer has. To me prolog checks all ifs down a path before checking the function that if comes from to make sure it checks all possibilities and if the possibilities fail in that path it backtracks and goes down a different path.