Lab 9: Unification & Backtracking

Prolog first calls the functor murderer(X) which is unified with hair(X, brown). Next, Prolog calls hair(X, brown) which is unified with attire(X, pincenez). From there, Prolog goes to check the first attire functor which is attire(mr_woodley, pincenez), and it unifies with attire(sir_raymond, tattered_cuffs). Then, Prolog calls attire(X, tattered_cuffs) where X is sir_raymond, and Prolog checks if sir_raymond was in room 16. The result of this check comes out to be false since sir_raymond was in room 10. Then Prolog backtracks to attire(sir_raymond, tattered_cuffs) where it is now false, and it backtracks all the way to attire(X, pincenez) to redo the procedure. Now Prolog checks the second attire functor which is attire(sir_raymond, pincenez) where it unifies with attire(mr_woodley, tattered_cuffs). Prolog then checks if mr_woodley was in room 16 which comes out to be true which means that attire(mr_woodley, tattered_cuffs) is true and attire(sir_raymond, pincenez) is true. Prolog instantiates the variable X to sir_raymond, and it backtracks to the hair functor where X is now sir_raymond where it comes out to be true since the murderer had brown hair which now means that murderer(sir_raymond) is now true, thus; sir_raymond was the murderer.