Adnar Lozano
CSE 320
Lab #9
11/21/17

Lab #9

The murderer is Sir Raymond.
The program begins with the statement that the murderer had brown hair.

```
murderer(X):-hair(X, brown).
```

The program is going to look at who had a pincenez for attire and unify the result to the person with brown hair and then to the murderer. Basically a unification and backtracking of true statements
So in order to solve this, the program needs to check who had brown hair based on some logic statements.
If we look at this line:

```
hair(X, brown) :- attire(X, pincenez).
```

we know that a person has brown hair if they were wearing the pincenez. So the person with brown hair is associated with the pincenez. The program then looks at who had the pincenez, but there is no explicit expression of who was wearing the pincenez. Instead, the program is given two choices: if Sir Raymond is wearing the tattered cuffs, then Mr Woodley is wearing the pincenez, and vice versa. These lines state the above.

```
% (1st choice) If Sir Raymond has tattered cuffs, then Mr. Woodley has pincenez
attire(mr_woodley, pincenez) :-
    attire(sir_raymond, tattered_cuffs).
% (2nd choice) vice versa
attire(sir_raymond, pincenez) :-
    attire(mr_woodley, tattered_cuffs).
```

The program then chooses the 1st option to see if it's true. So the program checks if Sir Raymond had tattered cuffs and Mr Woodley had pincenez. If we look at this line:

```
attire(X, tattered_cuffs) :- room(X, 16).
```

We can see that whoever was in room 16 was wearing tattered cuffs, so the program checks if Sir Raymond was in room 16. This call returns false (or fails) because it knows Mr Woodley was in room 16, this line states the above.

```
room(mr_woodley, 16).
```

So the program goes back to the previous state where it chose the 1st option and replaces it with the 2nd option to see if this time it's indeed true (Redo). The program then checks if Sir Raymond had pincenez, which is the same as Mr Woodley having the tattered cuffs. Again, to see who has tattered cuffs, the program looks at who was in room 16. This time, the parameter for who was in room 16 is now Mr Woodley. The program returns true (or exits) because there is an expression stating that Mr Woodley was in room 16. At this point, the program backtrack, unifying all the true statements. This means that if Mr Woodley was in room 16, then he was wearing the tattered cuffs, which is the same as Sir Raymond having the pincenez. If Sir Raymond was wearing the pincenez, then he is the one with brown hair. If he is the one with brown hair, then he is the murderer.

The following is a systematic of how the program flows

Murderer ---> Brown Hair ---> Pincenez ?
    If   Mr Woodley ---> Pincenez ---> Sir Raymond ---> Tattered Cuffs && Room 16 ---> FALSE
    else   Sir Raymond ---> Pincenez ---> Mr Woodley ---> Tattered Cuffs && Room 16 ---> TRUE

Murderer ---> Brown Hair ---> Pincenez ---> Sir Raymond   (Unification)
Sir Raymond ---> Pincenez ---> Brown Hair ---> Murderer   (Backtracking)

Below is a step by step explanation of how Prolog figured it out with the trace, murderer(X).

```
?- trace, murderer(X).                                  % program starts
   Call: (9) murderer(_3702) ? creep                    % calls the murderer expression
```

```
  Call: (10) hair(_3702, brown) ? creep                       % calls the see who had brown hair
  Call: (11) attire(_3702, pincenez) ? creep                  % calls to see who was wearing the pincenez
  Call: (12) attire(sir_raymond, tattered_cuffs) ? creep      % chooses 1st option (sir raymond has tattered cuffs)
  Call: (13) room(sir_raymond, 16) ? creep                    % checks if sir raymond was in room 16
  Fail: (13) room(sir_raymond, 16) ? creep                    % returns false for sir raymond being in room 16
  Fail: (12) attire(sir_raymond, tattered_cuffs) ? creep      % returns false for sir raymond wearing tattered cuffs
  Redo: (11) attire(_3702, pincenez) ? creep                  % calls to see who was wearing the pincenez. Again.
  Call: (12) attire(mr_woodley, tattered_cuffs) ? creep       % chooses 2nd option (mr woodley has tattered cuffs)
  Call: (13) room(mr_woodley, 16) ? creep                     % checks if mr woodley was in room 16
  Exit: (13) room(mr_woodley, 16) ? creep                     % returns true for mr woodley being in room 16
  Exit: (12) attire(mr_woodley, tattered_cuffs) ? creep       % backtracks for mr woodley wearing tattered cuffs
  Exit: (11) attire(sir_raymond, pincenez) ? creep            % backtracks for 2nd option that raymond had pincenez
  Exit: (10) hair(sir_raymond, brown) ? creep                 % backtracks for sir raymond having brown hair
  Exit: (9) murderer(sir_raymond) ? creep                     % backtracks for sir raymond being the murderer
X = sir_raymond .                                             % sets sir raymond as the murderer
```

The program then outputs 'The game is afoot....' and returns true because it was able to solve it.