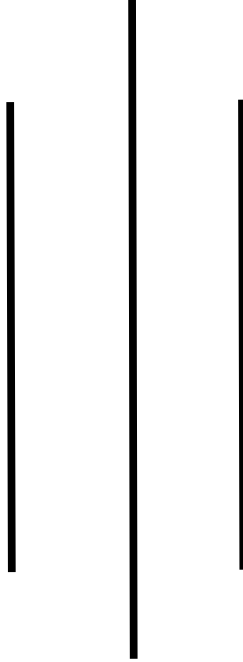


# Deerwalk Institute of Technology

Sifal, Kathmandu



## Artificial Intelligence Practical - 7

**Submitted By:**

Name: Sagar Giri

Roll No: 205

Section: A

**Submitted To:**

Sudan Prajapati

Date: 22 April 2015

# Tracing in Prolog

SWI Prolog allows us to trace the route that Prolog takes through a program when looking for a solution to a particular query. There are 2 ways to do this, the graphical and the non-graphical trace. Both yield up the same information but do so in different ways.

## Task 1:

```
adminworker(black).
```

```
adminworker(white).
```

```
officejunior(green).
```

```
manager(brown).
```

```
manager(grey).
```

```
supervise(X,Y):-adminworker(X),adminworker(Y).
```

```
supervise(X,Y):-adminworker(X),officejunior(Y).
```

```
supervise(X,Y):-manager(X),officejunior(Y).
```

## Tracing Output:

```
?- supervise(X,green).  
X = black ;  
X = white ;  
X = brown ;  
X = grey.
```

?- trace.

true.

[trace] ?- supervise(X,green).

Call: (6) supervise(\_G2507, green) ? creep

Call: (7) adminworker(\_G2507) ? creep

Exit: (7) adminworker(black) ? creep

Call: (7) adminworker(green) ? creep

Fail: (7) adminworker(green) ? creep

Redo: (7) adminworker(\_G2507) ? creep

Exit: (7) adminworker(white) ? creep

Call: (7) adminworker(green) ? creep

Fail: (7) adminworker(green) ? creep

Redo: (6) supervise(\_G2507, green) ? creep

Call: (7) adminworker(\_G2507) ? creep

Exit: (7) adminworker(black) ? creep

Call: (7) officejunior(green) ? creep

Exit: (7) officejunior(green) ? creep

Exit: (6) supervise(black, green) ? creep

X = black ;

Redo: (7) adminworker(\_G2507) ? creep

Exit: (7) adminworker(white) ? creep

Call: (7) officejunior(green) ? creep

Exit: (7) officejunior(green) ? creep

Exit: (6) supervise(white, green) ? creep

X = white ;

Redo: (6) supervise(\_G2507, green) ? creep

Call: (7) manager(\_G2507) ? creep

Exit: (7) manager(brown) ? creep

Call: (7) officejunior(green) ? creep

Exit: (7) officejunior(green) ? creep

Exit: (6) supervise(brown, green) ? creep

X = brown ;

Redo: (7) manager(\_G2507) ? creep

Exit: (7) manager(grey) ? creep

Call: (7) officejunior(green) ? creep

Exit: (7) officejunior(green) ? creep

Exit: (6) supervise(grey, green) ? creep

X = grey.

## Task 2:

kicking(malfoy,harry).

eating(dudley).

happy(aunt\_petunia):-happy(dudley).

happy(uncle\_vernon):-happy(dudley),unhappy(harry).

happy(dudley):-kicking(dudley,harry).

happy(dudley):-eating(dudley).

## Trace Output:

```
?- trace.  
true.  
  
[trace] ?- happy(aunt_petunia).  
  Call: (6) happy(aunt_petunia) ? creep  
  Call: (7) happy(dudley) ? creep  
  Call: (8) kicking(dudley, harry) ? creep  
  Fail: (8) kicking(dudley, harry) ? creep  
  Redo: (7) happy(dudley) ? creep  
  Call: (8) eating(dudley) ? creep  
  Exit: (8) eating(dudley) ? creep  
  Exit: (7) happy(dudley) ? creep  
  Exit: (6) happy(aunt_petunia) ? creep  
true.
```