Assignment 2
Prolog

Himanshu Jindal
108208979

Question 1

The Monkey and Banana Problem

Answer

Source Code

Please find the source code with comments as **'Q1.pl'**

Trace

Below is the trace of the execution of the program. It is also present inside the folder Q1 as 'Trace_Q1.txt'.

```
| ?- reach(X).
   (0) Call: reach(_h162) ?
   (1) Call: traverse_#301(state(a,b,c,start),_h162) ?
   (2) Call: path(state(a,b,c,start),_h218,_h223) ?
   (2) Exit: path(state(a,b,c,start),walk(a,c),state(c,b,c,with_chair)) ?
   (3) Call: traverse(state(c,b,c,with_chair),_h219) ?
   (4) Call: path(state(c,b,c,with_chair),_h258,_h263) ?
   (4) Exit: path(state(c,b,c,with_chair),push_chair(c,b),state(b,b,b,with_chair_below_banana)) ?
   (5) Call: traverse(state(b,b,b,with_chair_below_banana),_h259) ?
   (6) Call: path(state(b,b,b,with_chair_below_banana),_h298,_h303) ?
   (6) Exit:
path(state(b,b,b,with_chair_below_banana),climb_chair(b),state(b,b,b,on_chair_below_banana))
?
   (7) Call: traverse(state(b,b,b,on_chair_below_banana),_h299) ?
   (8) Call: path(state(b,b,b,on_chair_below_banana),_h337,_h342) ?
   (8) Exit: path(state(b,b,b,on_chair_below_banana),grasp(b),state(b,b,b,on_chair_with_banana))
?
   (9) Call: traverse(state(b,b,b,on_chair_with_banana),_h338) ?
   (9) Exit: traverse(state(b,b,b,on_chair_with_banana),[]) ?
   (7) Exit: traverse(state(b,b,b,on_chair_below_banana),[grasp(b)]) ?
   (5) Exit: traverse(state(b,b,b,with_chair_below_banana),[climb_chair(b),grasp(b)]) ?
   (3) Exit: traverse(state(c,b,c,with_chair),[push_chair(c,b),climb_chair(b),grasp(b)]) ?
   (1) Exit: traverse_#301(state(a,b,c,start),[walk(a,c),push_chair(c,b),climb_chair(b),grasp(b)]) ?
   (0) Exit: reach([walk(a,c),push_chair(c,b),climb_chair(b),grasp(b)]) ?

X = [walk(a,c),push_chair(c,b),climb_chair(b),grasp(b)]

yes
[trace]
```

Question 2

The analogy problem

Answer

Documentation

I have defined the facts for the figures separately in 2 files. So, I have 2 files with similar relate function and predicates, but different facts for the 2 figures

The first set of figure has different shaped objects, but orientation is same, that is one object inside another object

The second set of figures has the same objects in one figure, but orientation is different, with each object in a different part inside the figure.

Source Code

Please find the source code with comments as **'Q2_1.pl' and 'Q2_2.pl'**

Trace

Below is the trace of the execution of the program of figure 1. It is also present inside the folder Q2 as 'Trace_Q2_1.txt'.

```
| ?- analogy(1,5,3,X).
   (0) Call: analogy(1,5,3,_h186) ?
   (1) Call: figure(1,_h237) ?
   (1) Exit: figure(1,middle(triangle,square)) ?
   (2) Call: figure(5,_h258) ?
   (2) Exit: figure(5,middle(square,triangle)) ?
   (3) Call: relate(middle(triangle,square),middle(square,triangle),_h280) ?
   (3) Exit: relate(middle(triangle,square),middle(square,triangle),invert) ?
   (4) Call: figure(3,_h298) ?
   (4) Exit: figure(3,middle(square,circle)) ?
   (5) Call: relate(middle(square,circle),_h319,invert) ?
   (5) Exit: relate(middle(square,circle),middle(circle,square),invert) ?
   (6) Call: figure(_h186,middle(circle,square)) ?
   (6) Exit: figure(7,middle(circle,square)) ?
   (7) Call: write(invert) ?
invert   (7) Exit: write(invert) ?
   (0) Exit: analogy(1,5,3,7) ?

X = 7

yes
[trace]
```

Below is the trace of the execution of the program of figure 2. It is also present inside the folder Q2 as ‘Trace_Q2_2.txt’.

```
| ?- analogy(1,4,2,X).
   (0) Call: analogy(1,4,2,_h186) ?
   (1) Call: figure(1,_h237) ?
   (1) Exit: figure(1,middle(circle,center)) ?
   (2) Call: figure(4,_h258) ?
   (2) Exit: figure(4,middle(square,center)) ?
   (3) Call: relate(middle(circle,center),middle(square,center),_h280) ?
   (3) Exit: relate(middle(circle,center),middle(square,center),same_orientatio
) ?
   (4) Call: figure(2,_h298) ?
   (4) Exit: figure(2,middle(circle,top_left)) ?
   (5) Call: relate(middle(circle,top_left),_h319,same_orientation) ?
   (5) Exit: relate(middle(circle,top_left),middle(_h330,top_left),same_orienta
ion) ?
   (6) Call: figure(_h186,middle(_h330,top_left)) ?
   (6) Exit: figure(2,middle(circle,top_left)) ?
   (7) Call: write(same_orientation) ?
same_orientation   (7) Exit: write(same_orientation) ?
   (0) Exit: analogy(1,4,2,2) ?

X = 2,
   (0) Redo: analogy(1,4,2,2) ?
   (7) Redo: write(same_orientation) ?
   (7) Fail: write(same_orientation) ?
   (6) Redo: figure(2,middle(circle,top_left)) ?
   (6) Exit: figure(5,middle(square,top_left)) ?
   (8) Call: write(same_orientation) ?
same_orientation   (8) Exit: write(same_orientation) ?
   (0) Exit: analogy(1,4,2,5) ?

X = 5,
   (0) Redo: analogy(1,4,2,5) ?
   (8) Redo: write(same_orientation) ?
   (8) Fail: write(same_orientation) ?
   (6) Redo: figure(5,middle(square,top_left)) ?
   (6) Fail: figure(_h186,middle(_h330,top_left)) ?
   (5) Redo: relate(middle(circle,top_left),middle(_h330,top_left),same_orienta
ion) ?
   (5) Fail: relate(middle(circle,top_left),_h319,same_orientation) ?
   (4) Redo: figure(2,middle(circle,top_left)) ?
   (4) Fail: figure(2,_h298) ?
   (3) Redo: relate(middle(circle,center),middle(square,center),same_orientatio
) ?
   (3) Fail: relate(middle(circle,center),middle(square,center),_h280) ?
```

(2) Redo: figure(4,middle(square,center)) ?
(2) Fail: figure(4,_h258) ?
(1) Redo: figure(1,middle(circle,center)) ?
(1) Fail: figure(1,_h237) ?
(0) Fail: analogy(1,4,2,_h186) ?

no
[trace]