Artificial Intelligence Lab

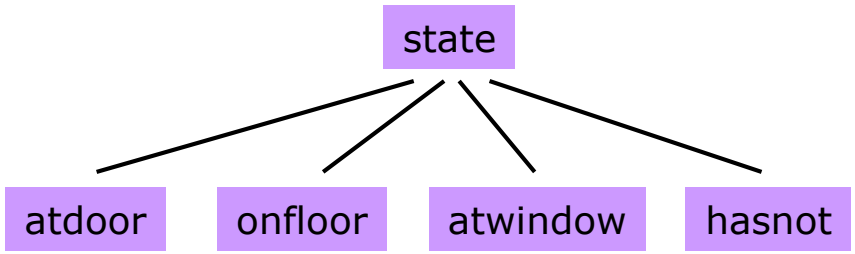
Experiment 3
Monkey and Banana

Monkey and Banana

- **Problem:**
  - There is a monkey at the door into a room.
  - In the middle of the room a banana is hanging from the ceiling.
  - The monkey is hungry and wants to get the banana, but he cannot reach high enough from the floor.
  - At the window of the room there is a box the monkey may use.
  - The monkey can perform the following actions: walk on the floor, climb the box, push the box around and grasp the banana if standing on the box directly under the banana.
  - Can the monkey get the banana?

Monkey and Banana

- The representation of the problem:
  - The initial state:
    (1) Monkey is at door.
    (2) Monkey is on floor.
    (3) Box is at window.
    (4) Monkey does not have banana.
    state(atdoor, onfloor, atwindow, hasnot)



  - The goal of the game:
    state(_, _, _, has)

Monkey and banana

- Four types of moves:
  (1) grasp banana,
  (2) climb box,
  (3) push box,
  (4) walk around.
- A three-place relation:

  move(State1, Move, State2)

‘grasp’:

move(state(middle, onbox, middle, hasnot),
grasp,
state(middle, onbox, middle, has)).

Monkey and banana

'walk':

```
move( state( P1, onfloor, B, H),
          walk( P1, P2),
          state( P2, onfloor, B, H)).
```

'push':

```
move( state( P1, onfloor, P1, H),
          push( P1, P2),
          state( P2, onfloor, P2, H)).
```

'climb':

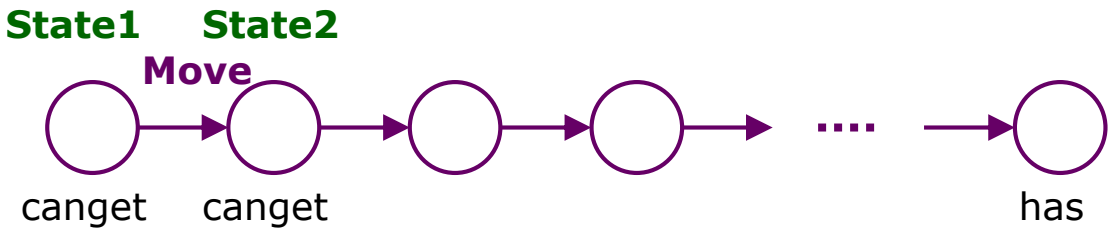
```
move( state( P, onfloor, P, H),
         climb,
         state( P, onbox, P, H)).
```

Monkey and Banana

- Question: Can the monkey in some initial state **State** get the banana?

  canget(State)

```
canget( state( _, _, _, has)).
canget(State1) :-
        move( State1, Move, State2),
        canget( State2).
```



Monkey and Banana

A program for the monkey and banana problem.

```
move( state( middle, onbox, middle, hasnot),
        grasp,
        state( middle, onbox, middle, has) ).

move( state( P, onfloor, P, H),
        climb,
        state( P, onbox, P, H) ).

move( state( P1, onfloor, P1, H),
        push( P1, P2),
        state( P2, onfloor, P2, H) ).

move( state( P1, onfloor, B, H),
        walk( P1, P2),
        state( P2, onfloor, B, H) ).

canget( state( _, _, _, has) ).

canget( State1)  :- move( State1, Move, State2), canget( State2).
```

Monkey and Banana

```
| ?- canget( state( atdoor, onfloor, atwindow, hasnot)).
true ?
Yes (The monkey can grasp the banana from this state.)

{trace}
| ?- canget( state( atdoor, onfloor, atwindow, hasnot)).
    1    1  Call: canget(state(atdoor,onfloor,atwindow,hasnot)) ?
    2    2  Call: move(state(atdoor,onfloor,atwindow,hasnot),_45,_85) ?
    2    2  Exit: move(state(atdoor,onfloor,atwindow,hasnot),
                      walk(atdoor,_73),state(_73,onfloor,atwindow,hasnot)) ?
    3    2  Call: canget(state(_73,onfloor,atwindow,hasnot)) ?
    4    3  Call: move(state(_73,onfloor,atwindow,hasnot),_103,_143) ?
    4    3  Exit: move(state(atwindow,onfloor,atwindow,hasnot),
                      climb,state(atwindow,onbox,atwindow,hasnot)) ?
    5    3  Call: canget(state(atwindow,onbox,atwindow,hasnot)) ?
    6    4  Call: move(state(atwindow,onbox,atwindow,hasnot),_158,_198) ?
    6    4  Fail: move(state(atwindow,onbox,atwindow,hasnot),_158,_186) ?
    5    3  Fail: canget(state(atwindow,onbox,atwindow,hasnot)) ?
    4    3  Redo: move(state(atwindow,onfloor,atwindow,hasnot),
                      climb,state(atwindow,onbox,atwindow,hasnot)) ?
```

Monkey and Banana

```
    4    3  Exit: move(state(atwindow,onfloor,atwindow,hasnot),
                    push(atwindow,_131),state(_131,onfloor,_131,hasnot)) ?
    5    3  Call: canget(state(_131,onfloor,_131,hasnot)) ?
    6    4  Call: move(state(_131,onfloor,_131,hasnot),_161,_201) ?
    6    4  Exit: move(state(_131,onfloor,_131,hasnot),
                    climb,state(_131,onbox,_131,hasnot)) ?
    7    4  Call: canget(state(_131,onbox,_131,hasnot)) ?
    8    5  Call: move(state(_131,onbox,_131,hasnot),_216,_256) ?
    8    5  Exit: move(state(middle,onbox,middle,hasnot),
                    grasp,state(middle,onbox,middle,has)) ?
    9    5  Call: canget(state(middle,onbox,middle,has)) ?
    9    5  Exit: canget(state(middle,onbox,middle,has)) ?
    7    4  Exit: canget(state(middle,onbox,middle,hasnot)) ?
    5    3  Exit: canget(state(middle,onfloor,middle,hasnot)) ?
    3    2  Exit: canget(state(atwindow,onfloor,atwindow,hasnot)) ?
    1    1  Exit: canget(state(atdoor,onfloor,atwindow,hasnot)) ?
true ?
(62 ms) yes
{trace}
```