

```
% INST0072 example answer to exercise 6-8
```

```
/* Monkey and Banana program (adapted from Bratko)
```

```
THE 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 wants to get the banana, but he cannot stretch 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 (if it is already at the box) and grab the banana if standing on the box directly under the banana. What sequence of actions does the monkey have to perform in order to get the banana?

The arguments of "state" are as follows:

```
1st argument: horizontal position of monkey
                  (possible values "atDoor", "atWindow" or "middle")
2nd argument: vertical position of monkey
                  (possible values "onFloor" or "onBox")
3rd argument: position of box
                  (possible values "atDoor", "atWindow" or "middle")
4th argument: whether the monkey has got the banana or not
                  (possible values "has" or "hasNot")
```

```
EXAMPLE QUERY:
```

```
?- is_plan(state(atDoor, onFloor, atWindow, hasNot), state(atDoor, onBox, atWindow, has), Plan)
```

```
*/
```

```
place(atWindow).
place(middle).
place(atDoor).
```

```
can_do(state(middle, onBox, middle, hasNot),
        grab,
        state(middle, onBox, middle, has)).
```

```
can_do(state(Position, onFloor, Position, X),
        climbOn,
        state(Position, onBox, Position, X)) :-
    place(Position).
```

```
can_do(state(Position_before, onFloor, Position_before, X),
        push(Position_before, Position_after),
        state(Position_after, onFloor, Position_after, X)) :-
    place(Position_before),
    place(Position_after),
    Position_before \== Position_after.
```

```
can_do(state(Position_before, onFloor, X, Y),
        walk(Position_before, Position_after),
        state(Position_after, onFloor, X, Y)) :-
    place(Position_before),
    place(Position_after),
    Position_before \== Position_after.
```

```
is_plan(Goal_state, Goal_state, []).
```

```
is_plan(State, Goal_state, [Action|Plan_for_state_after]) :-
    can_do(State, Action, State_after),
    is_plan(State_after, Goal_state, Plan_for_state_after).
```