HOW DO I KNOW MY ROBOT IS SAFE?

Louise Dennis, University of Liverpool

Dejanira Araiza-Illan (Bristol Robotics Lab), Christian Blum (Humbolt-Universitat zu Berlin), Kirstin Eder (Bristol Robotics Lab), Michael Fisher, Maryam Kamali (Liverpool), Nick Lincoln (Southampton), Wenguo Liu (UWE), Owen McAree, Sandor Veres (Sheffield), Matt Webster (Liverpool), Alan Winfield (UWE)

YOU CAN'T

AUTONOMOUS SYSTEMS

WE ALREADY HAVE ROBOTS

- Work to pre-determined sequences of exact instructions.

- In environments where the unexpected is unlikely.

- Are kept away from people.

- And/or are controlled by a human.

WE WANT ROBOTS…

- In unpredictable environments (search and rescue, urban streets, the home)
- To work with people,
- Without constant supervision

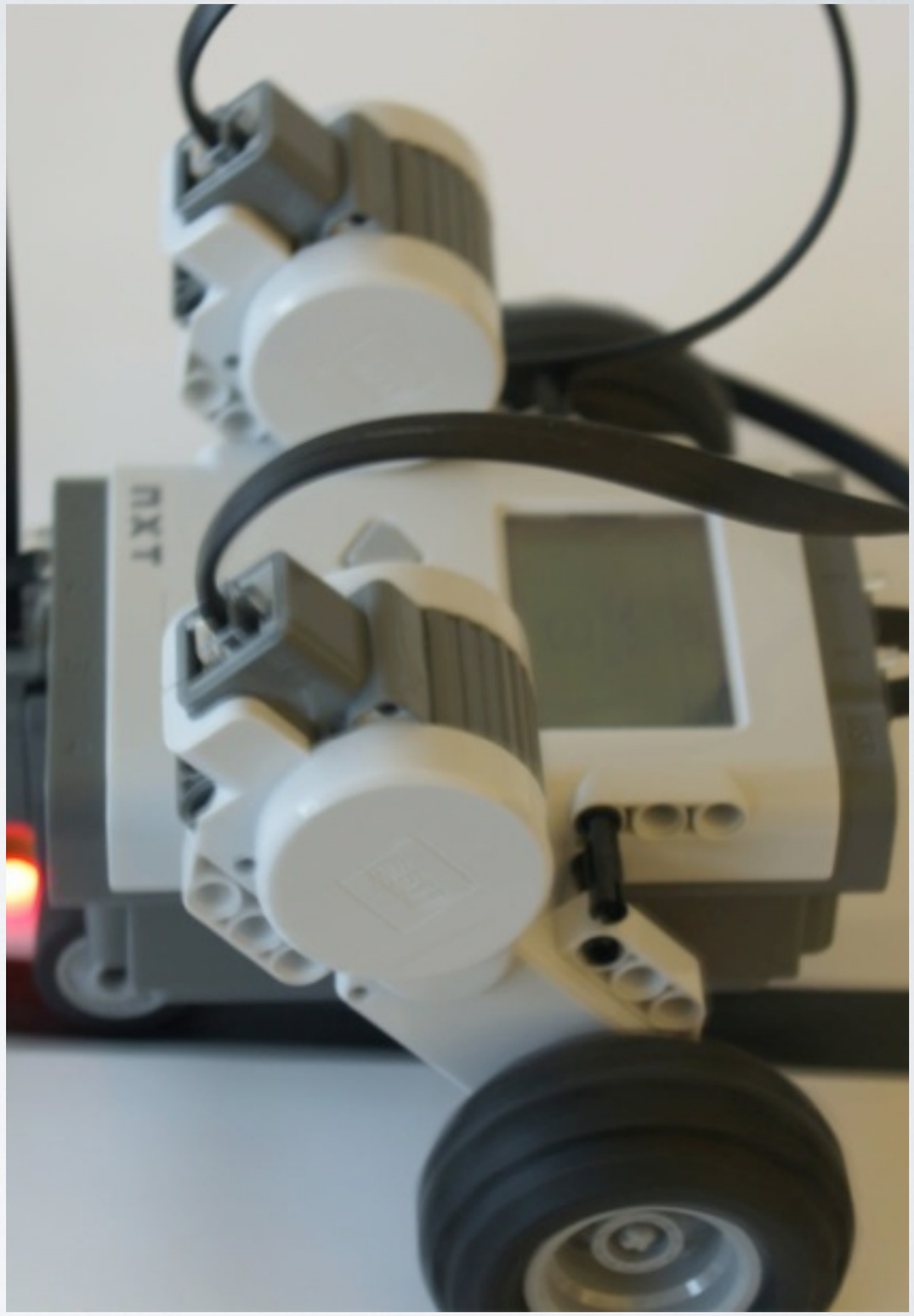
WHAT MAKES A ROBOT?

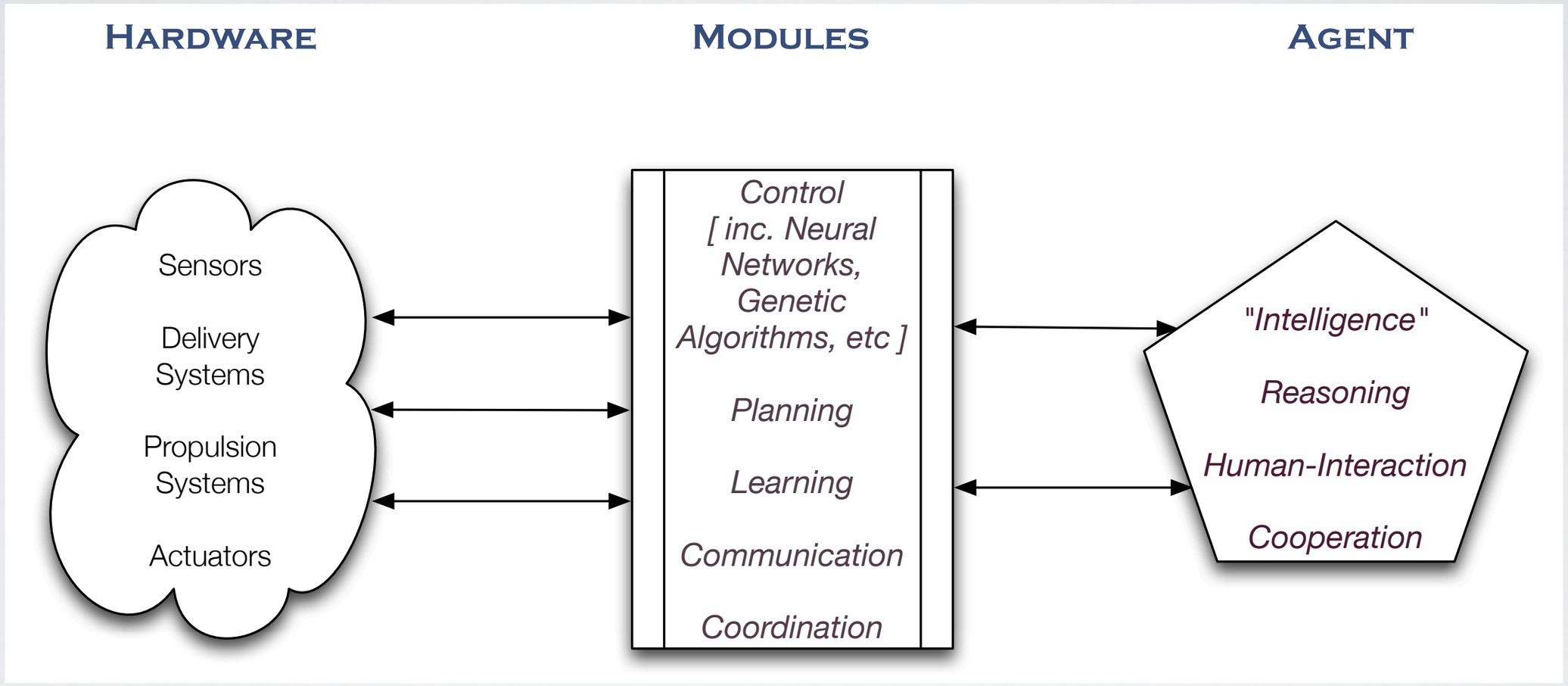
Sensors, Actuators, Control

CONTROL THEORY

Following a Line
No decisions required

DEMO




Hardware
Modules
Agent
Sensors
Delivery Systems
Propulsion Systems
Actuators
Control [inc. Neural Networks, Genetic Algorithms, etc]
Planning
Learning
Communication
Coordination
"Intelligence"
Reasoning
Human-Interaction
Cooperation

```
+obstacle : {G keep_moving [perform], ~B shutting_down} <-
    print("Believe there is an obstacle"),
    print("Turning right"),
    perf(right);
-obstacle : {G keep_moving [perform], ~B shutting_down} <-
print("Believe there is no obstacle"),
print("Going forward"),
perf(forward);
```

```
+click : {~G keep_moving [perform], ~B shutting_down} <-
      +! keep_moving [perform];
-click: {True} <-
       +waiting_for_second_click;
+click : {B waiting_for_second_click, ~B shutting_down} <-
       +shutting_down,
       perf(stop),
       shutdown;

+! keep_moving [perform] : {~B obstacle} <-
   perf(forward),
   *waiting_for_second_click,
   *click,
   print("Goal performed");
+! keep_moving [perform] : {B obstacle} <-
   perf(right),
   *waiting_for_second_click,
   *click,
   print("Goal performed");

+obstacle : {G keep_moving [perform], ~B shutting_down} <-
    print("Believe there is an obstacle"),
    print("Turning right"),
    perf(right);
-obstacle : {G keep_moving [perform], ~B shutting_down} <-
    print("Believe there is no obstacle"),
    print("Going forward"),
    perf(forward);
```

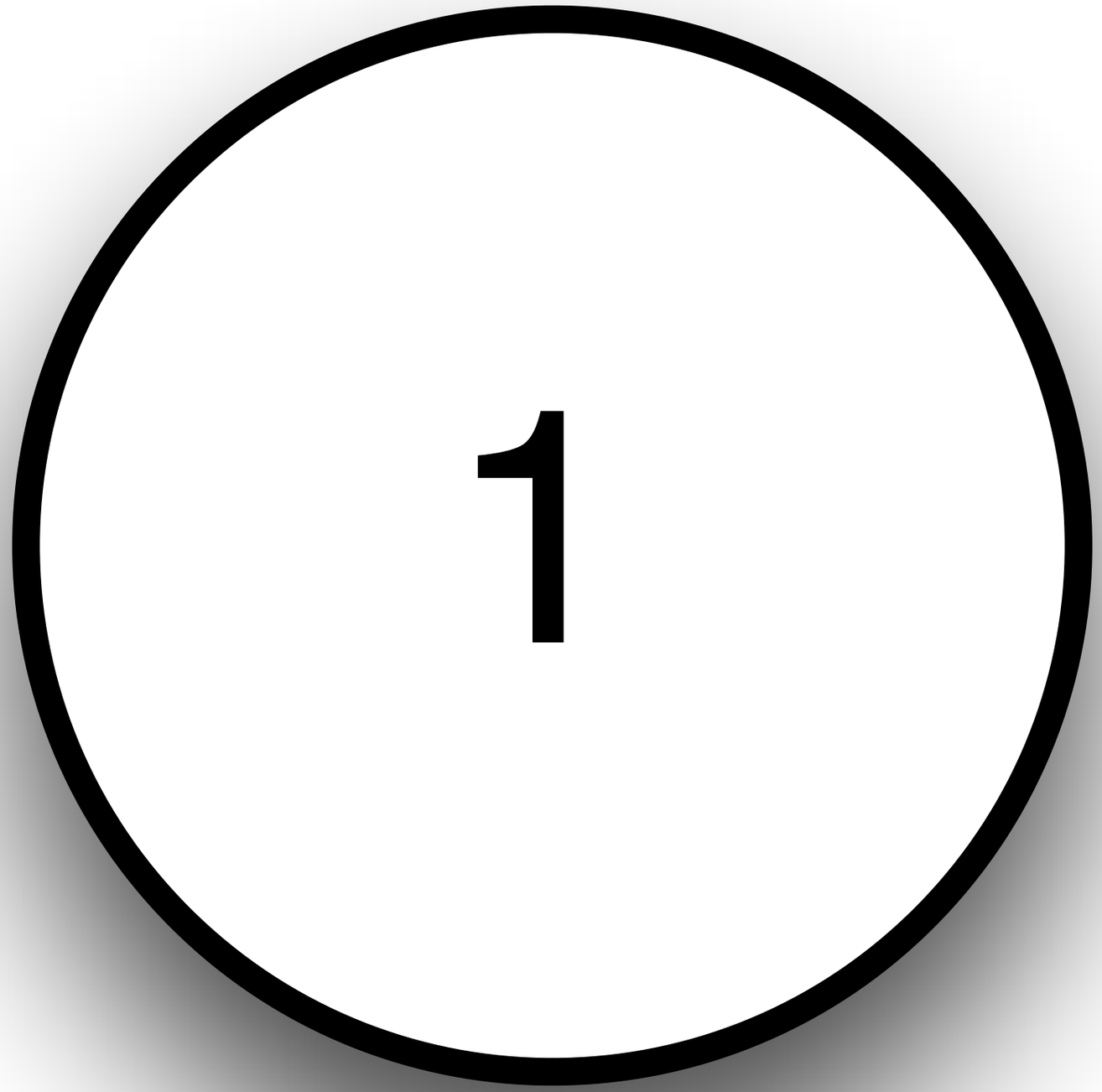
DEMO

MODEL CHECKING

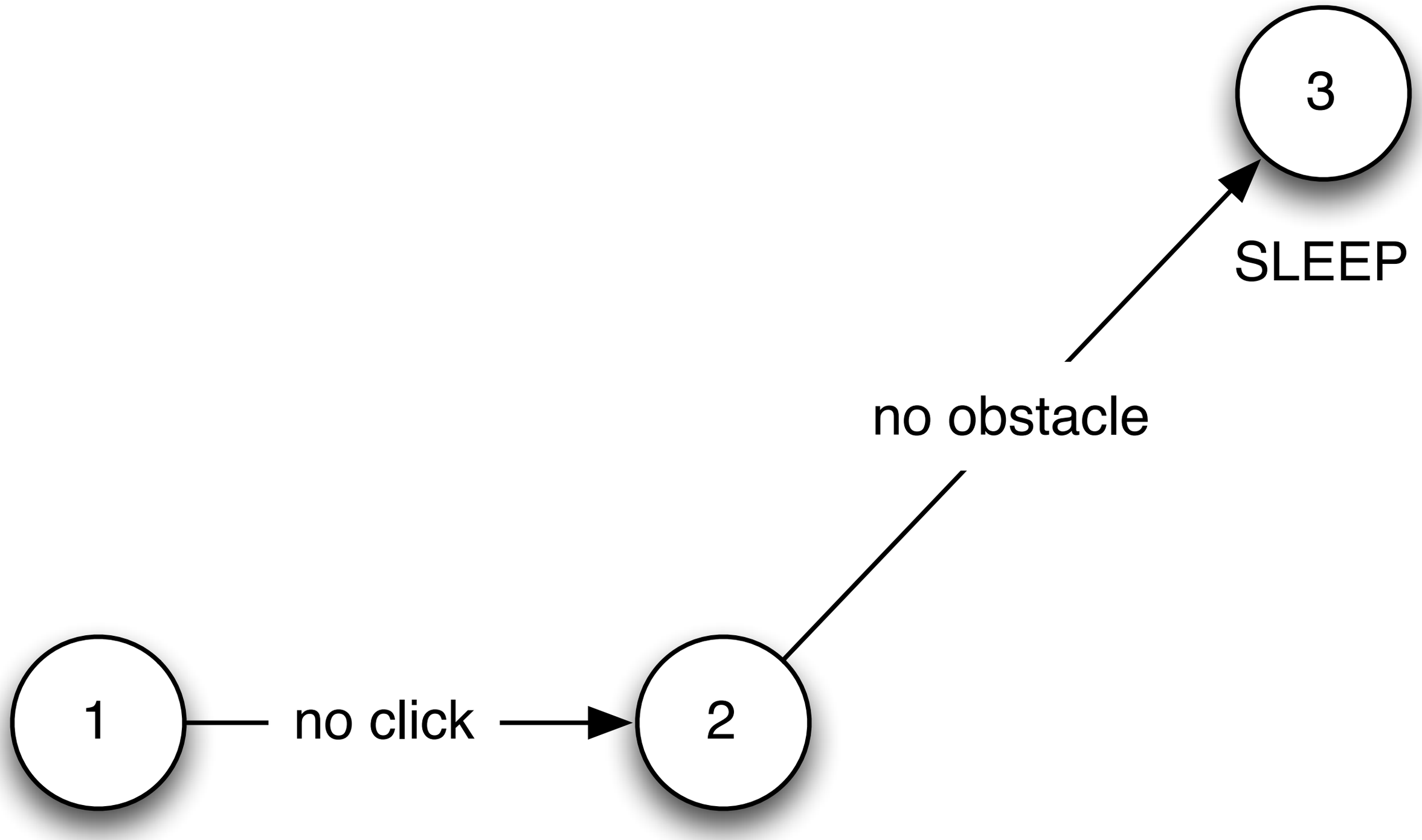
- We explore every choice the robot can make given all possible inputs to its reasoning program.

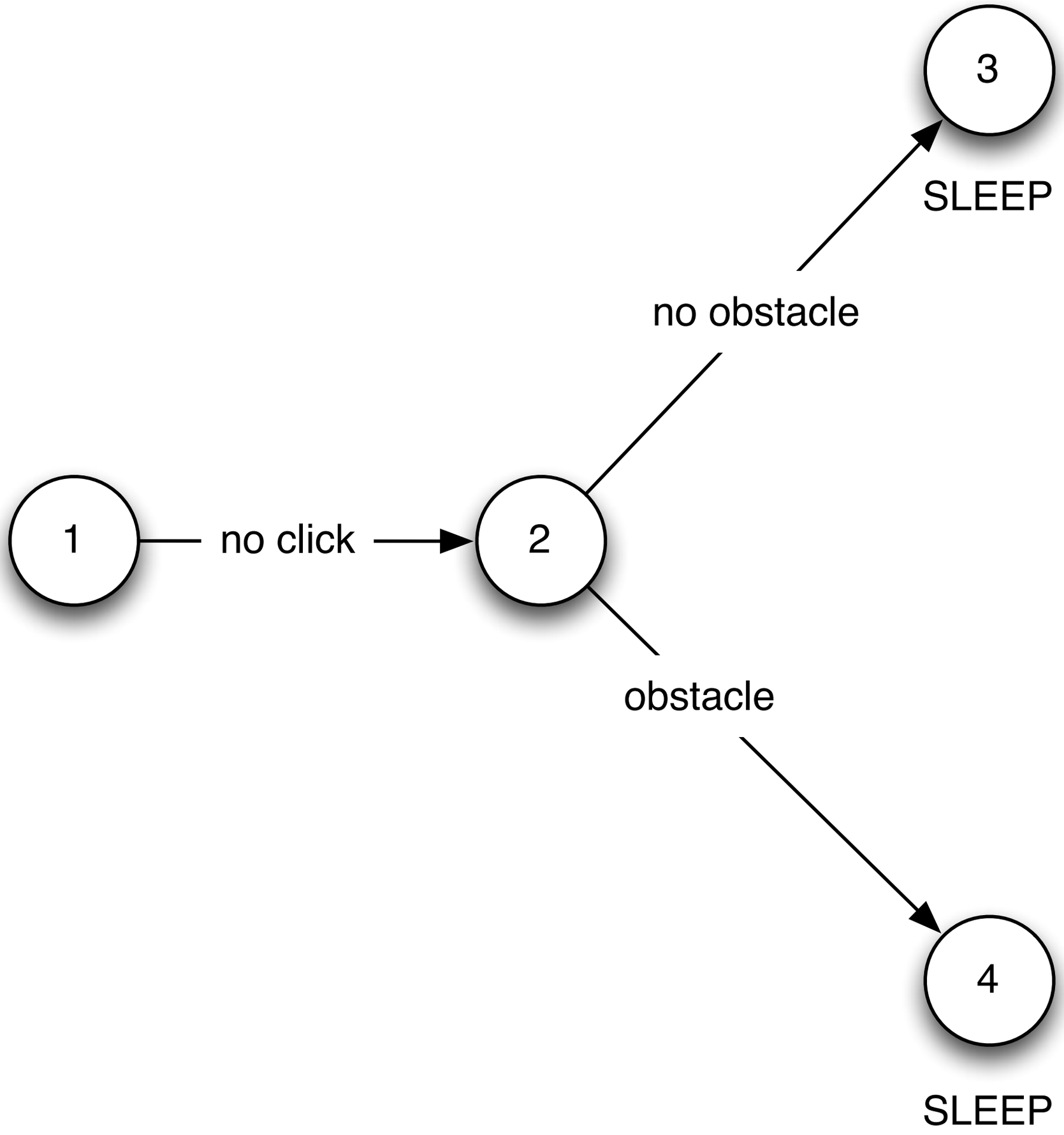
- We check it against some property such as *if dysprosium believes there is an obstacle and it has a goal to keep moving then eventually it will turn right or shut down.*

- B(dysprosium, obstacle) & G(dysprosium, keep_moving) →
  - D(dysprosium, perf(right)) OR B(dysprosium, shutting_down))

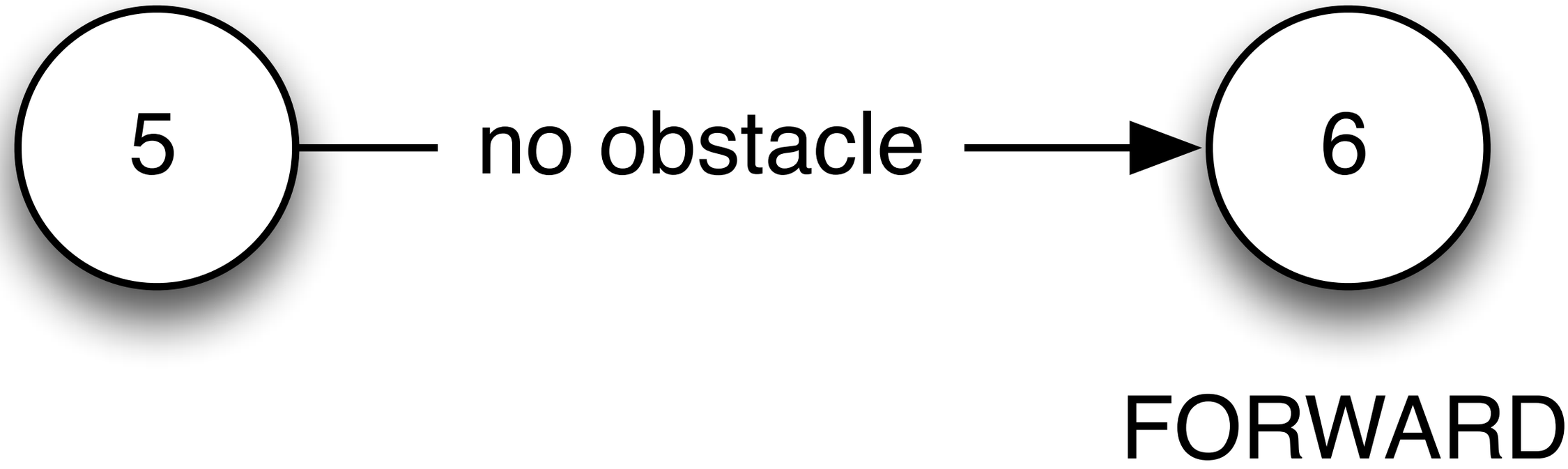

1

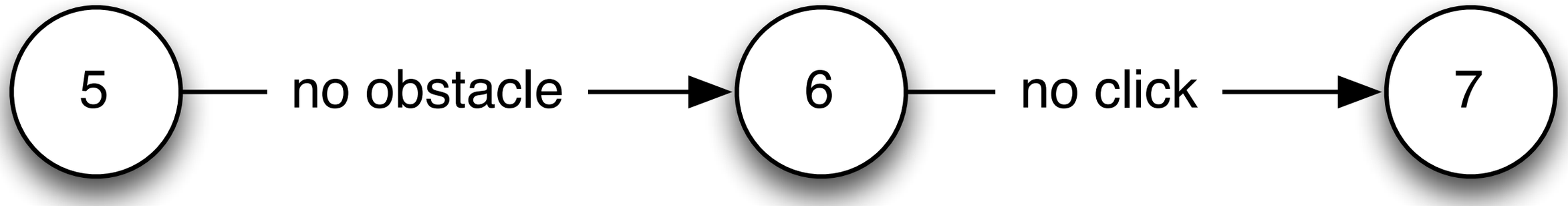

1
no click
2


3
SLEEP
no obstacle
1
no click
2

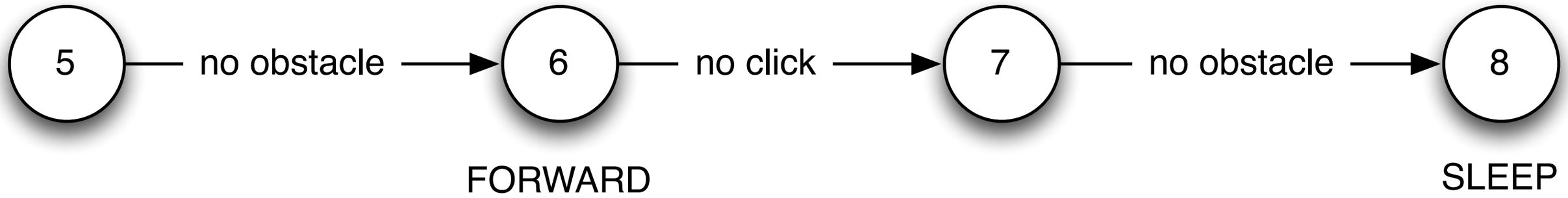

3
SLEEP
no obstacle
1
no click
2
obstacle
4
SLEEP

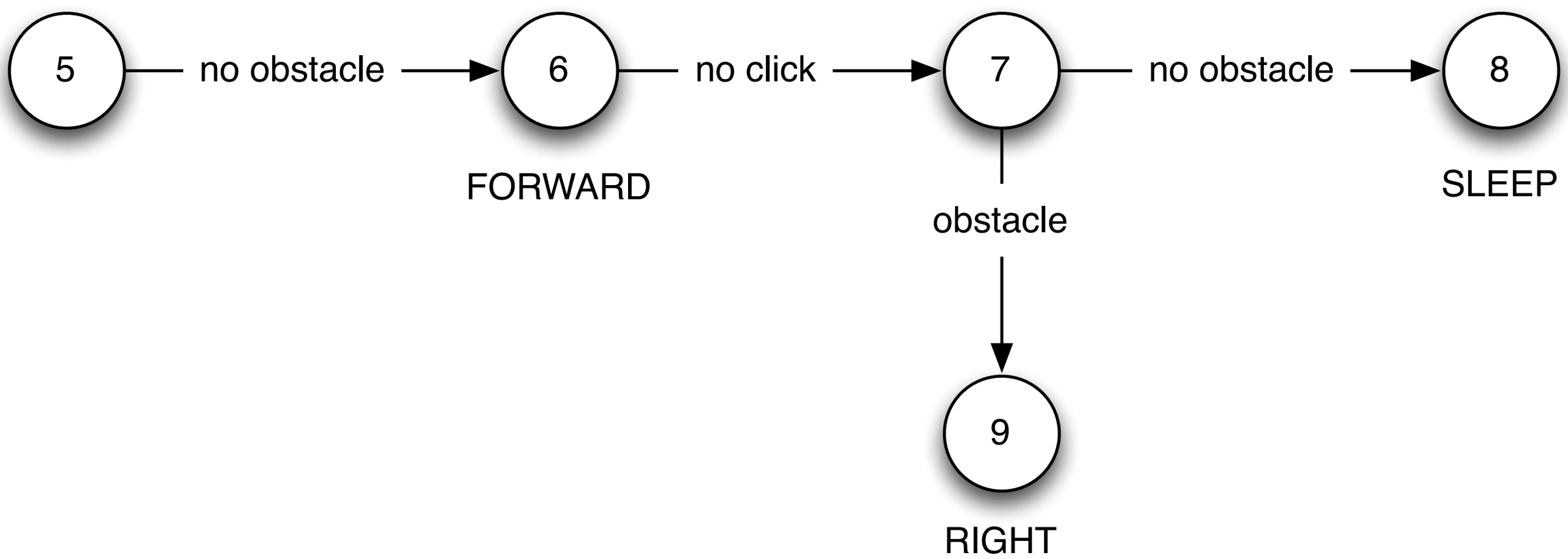

5

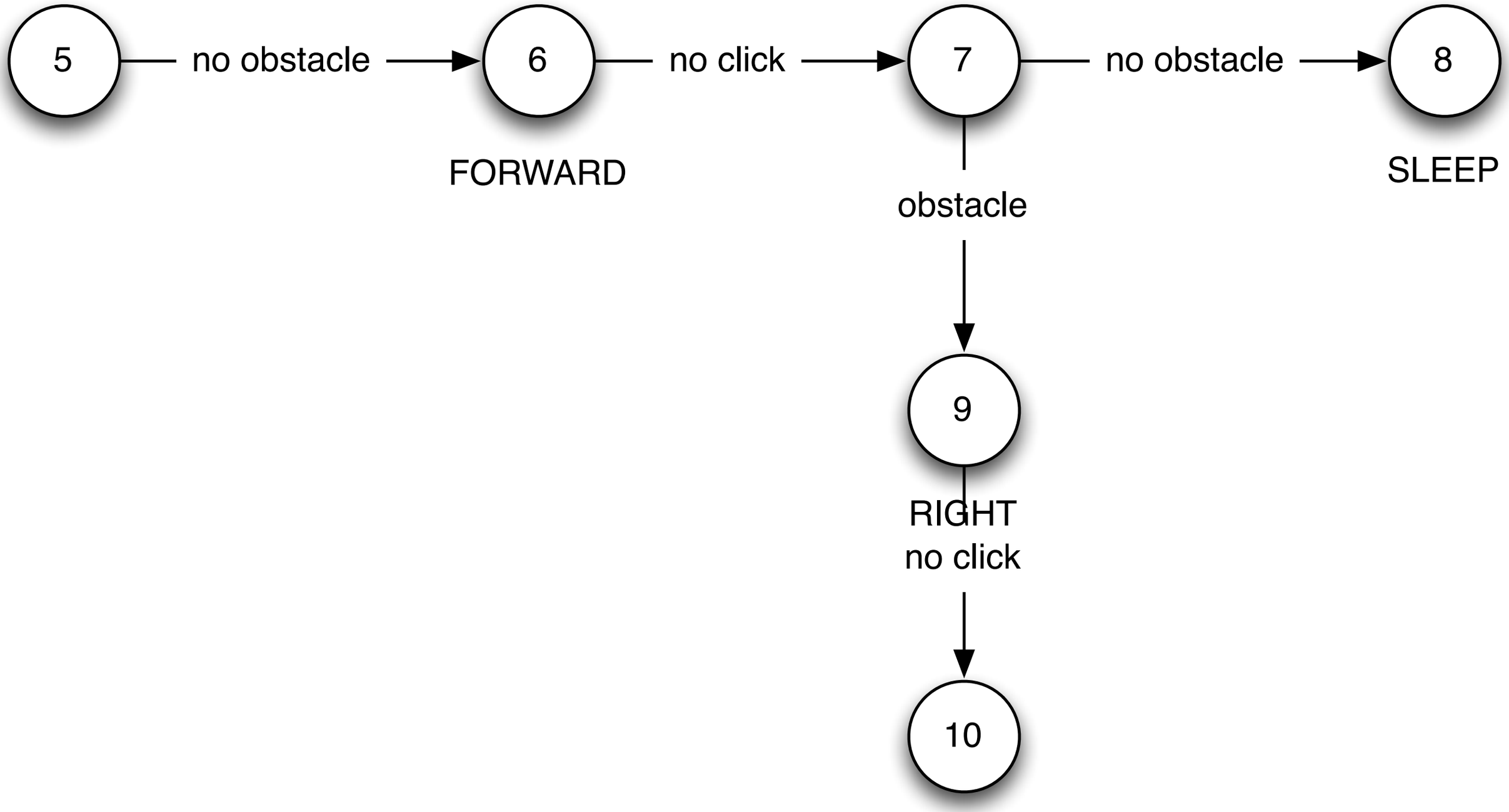

5
no obstacle
6
FORWARD

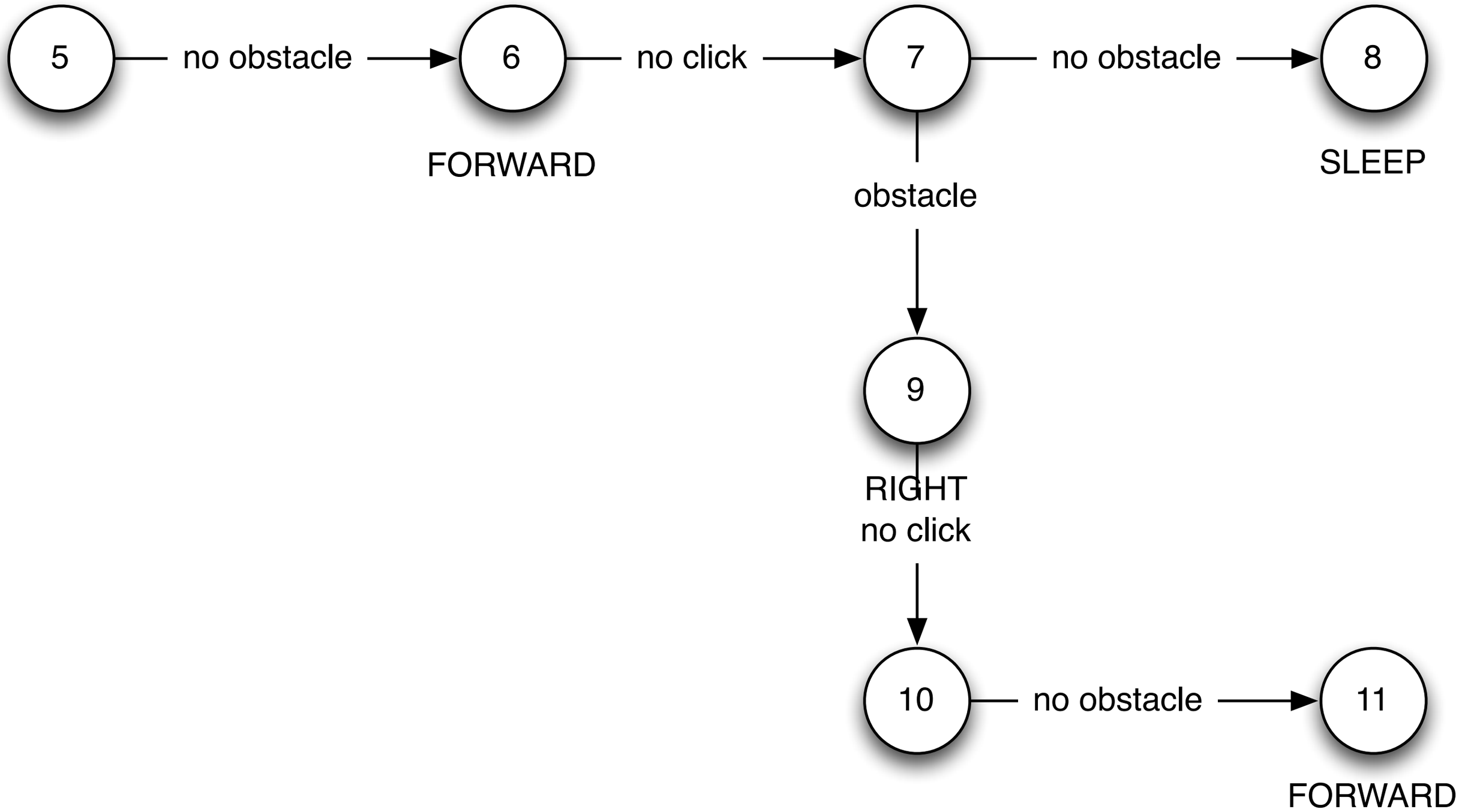


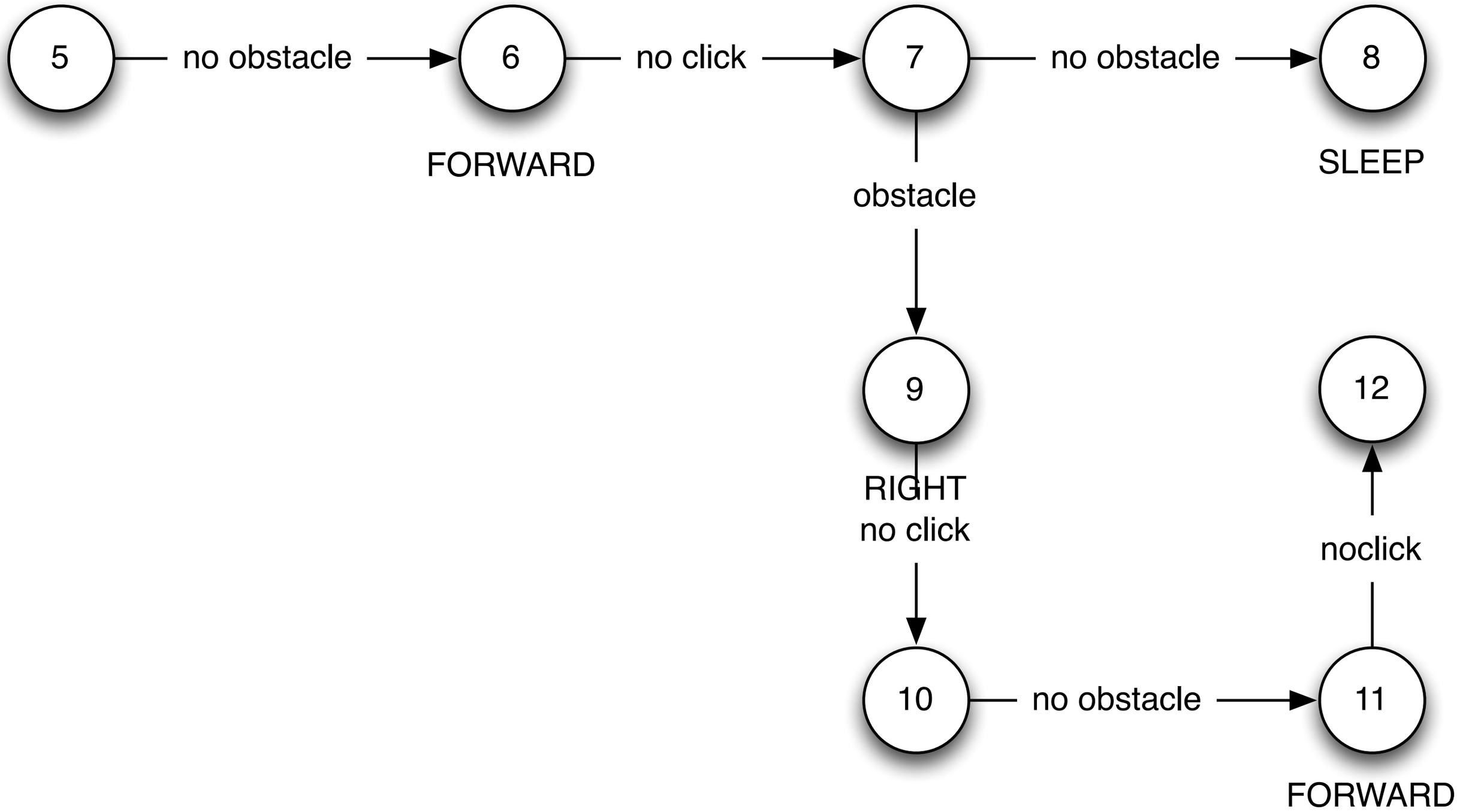
FORWARD

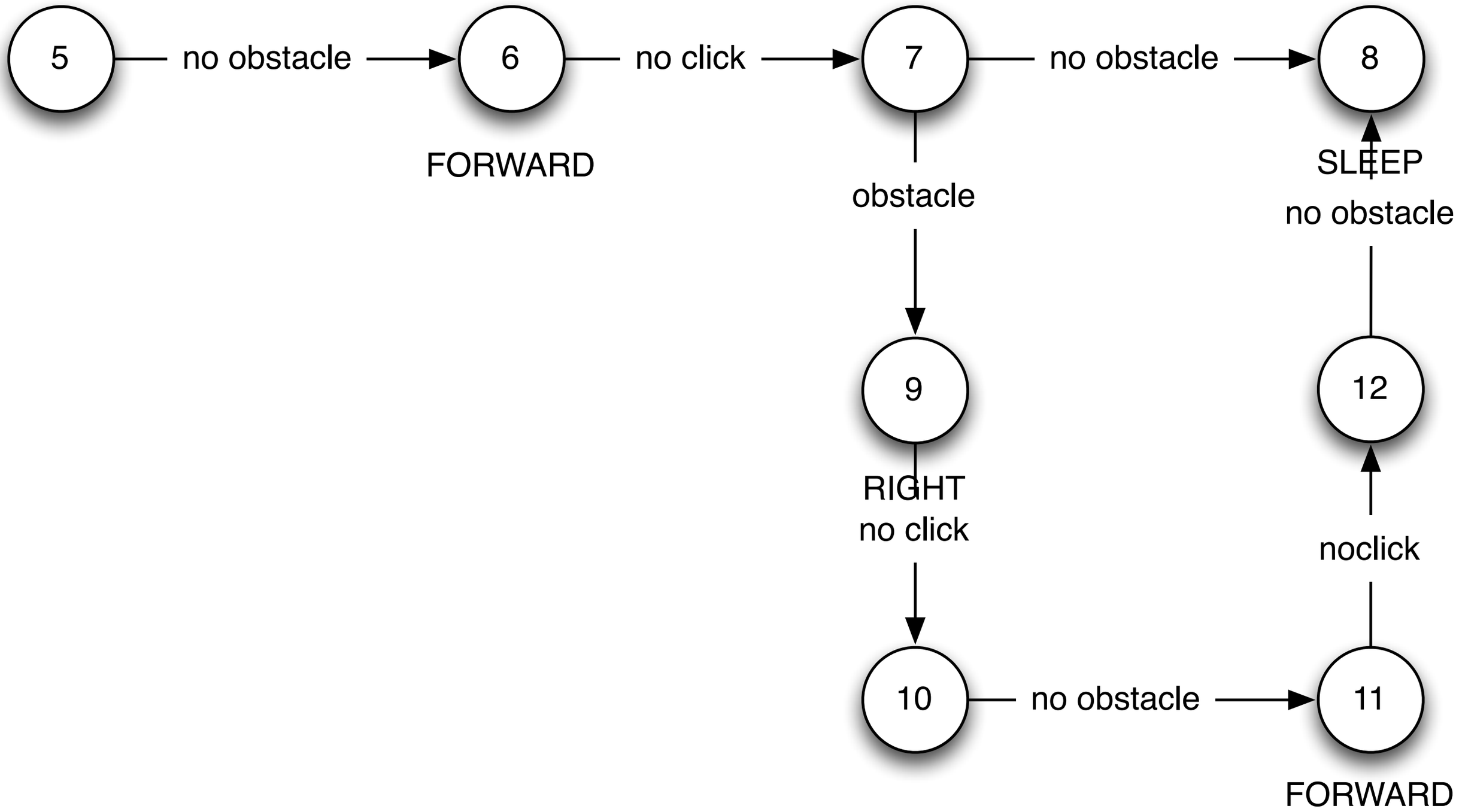

5
no obstacle
6
no click
7
no obstacle
8
FORWARD
SLEEP

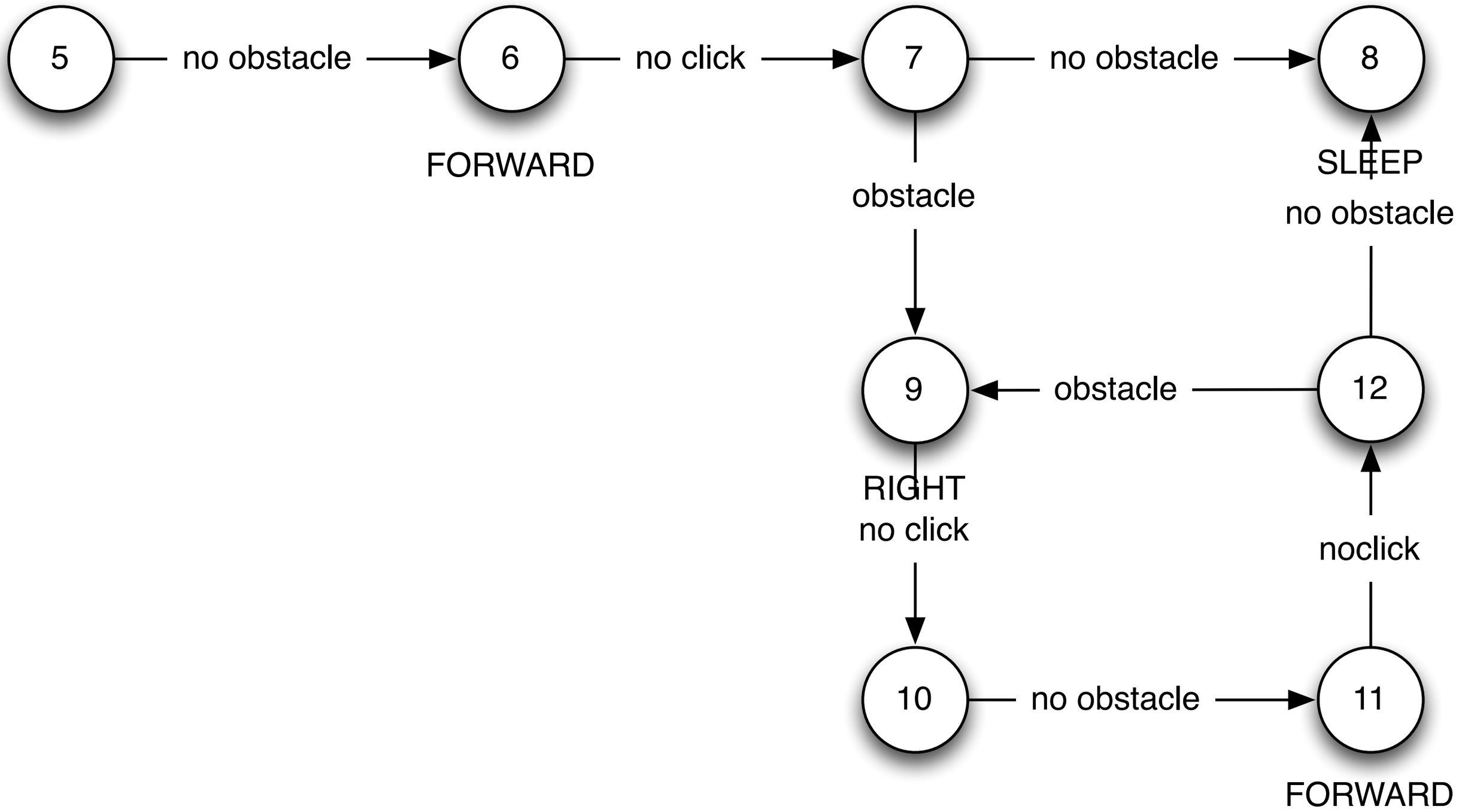

5
no obstacle
6
FORWARD
no click
7
obstacle
9
RIGHT
no obstacle
8
SLEEP


5
no obstacle
6
FORWARD
no click
7
obstacle
9
RIGHT
no click
10
no obstacle
8
SLEEP


5
no obstacle
6
FORWARD
no click
7
obstacle
9
RIGHT
no click
10
no obstacle
11
FORWARD
no obstacle
8
SLEEP


5
no obstacle
6
FORWARD
no click
7
obstacle
no obstacle
8
SLEEP
9
RIGHT
no click
10
no obstacle
11
FORWARD
noclick
12


5
no obstacle
6
FORWARD
no click
7
obstacle
9
RIGHT
no click
10
no obstacle
11
FORWARD
noclick
12
no obstacle
SLEEP
8


5
no obstacle
6
FORWARD
no click
7
obstacle
no obstacle
8
SLEEP
no obstacle
9
obstacle
12
RIGHT
no click
noclick
10
no obstacle
11
FORWARD

JUST TO PROVE IT REALLY WORKS

IN THEORY THEREFORE

NOT G(dysprosium, go_on_a_killing_spree)

NOT D(dysprosium, fire_lasers)

IN CASE YOU WERE WORRIED ABOUT DYSPROSIUM

WHERE DO THESE PROPERTIES COME FROM?

FORMALISING THE RULES OF THE AIR

Work by the Virtual Engineering Centre, Daresbury Labs

"But what does the aircraft do when obeying the rules of the air would cause a crash?"

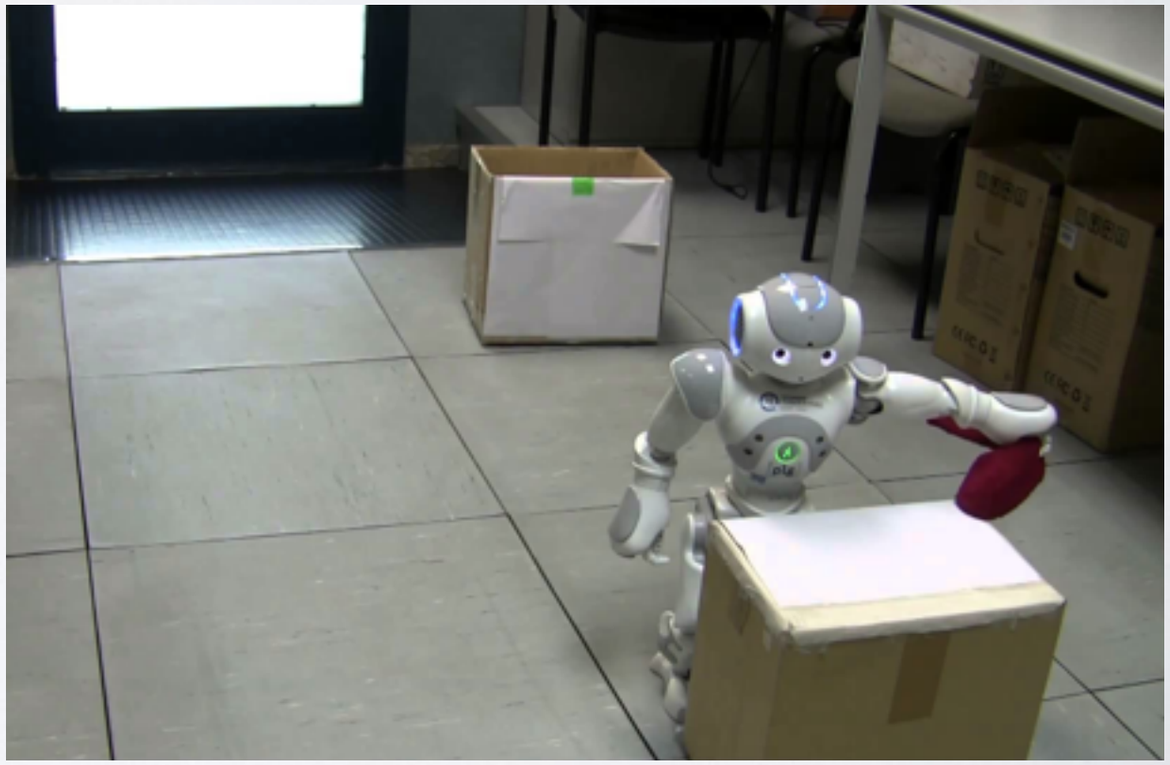
–We will be coming back to this

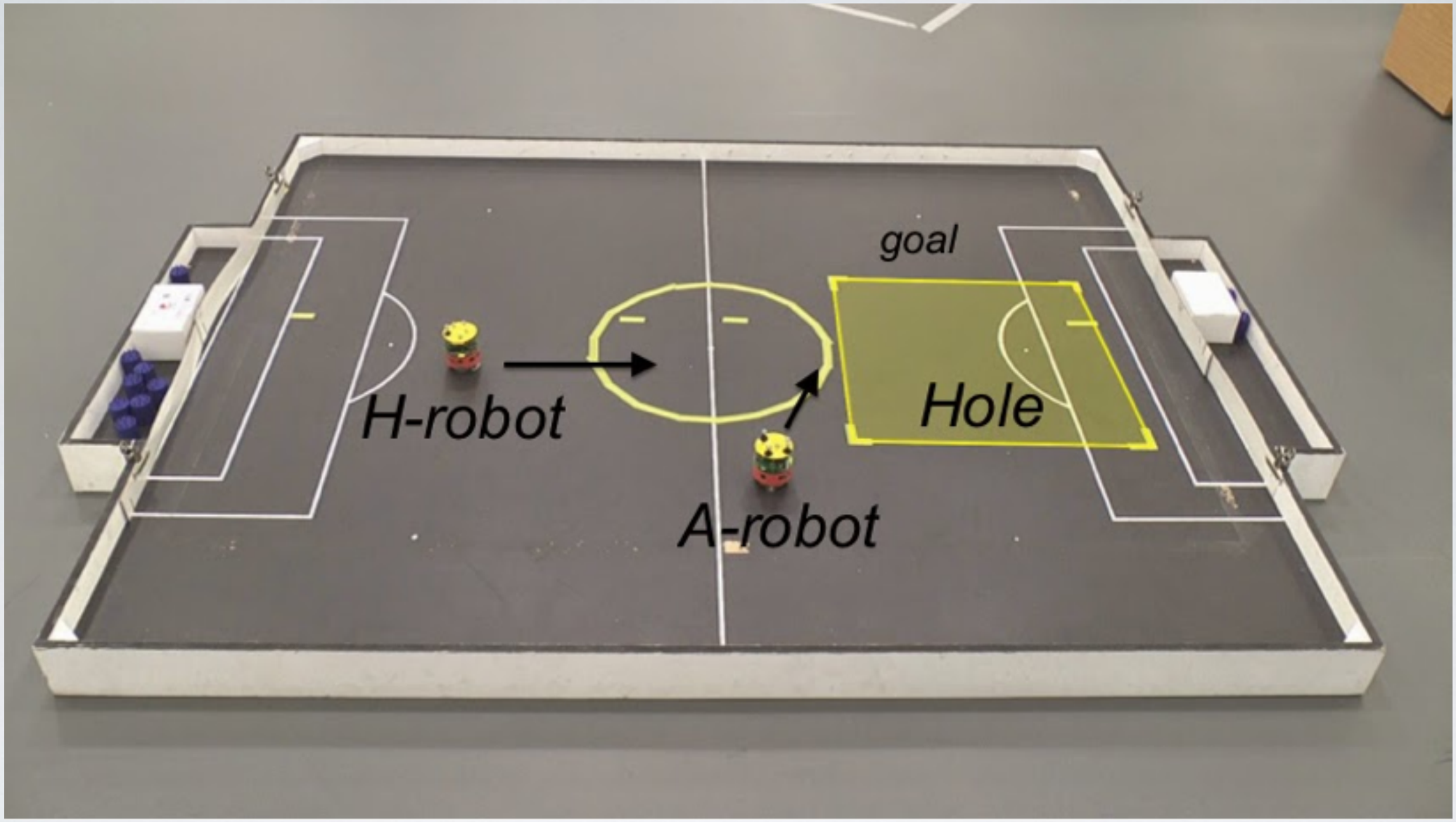
An Alternative to Agents: Planning

initial state: at(A), adjacent(A,B)
goal state: at(B)

actions:

move(X, Y)
preconditions: at(X), adjacent(X,Y)
postconditions: not at(X), at(Y)





ETHICAL ROBOTS

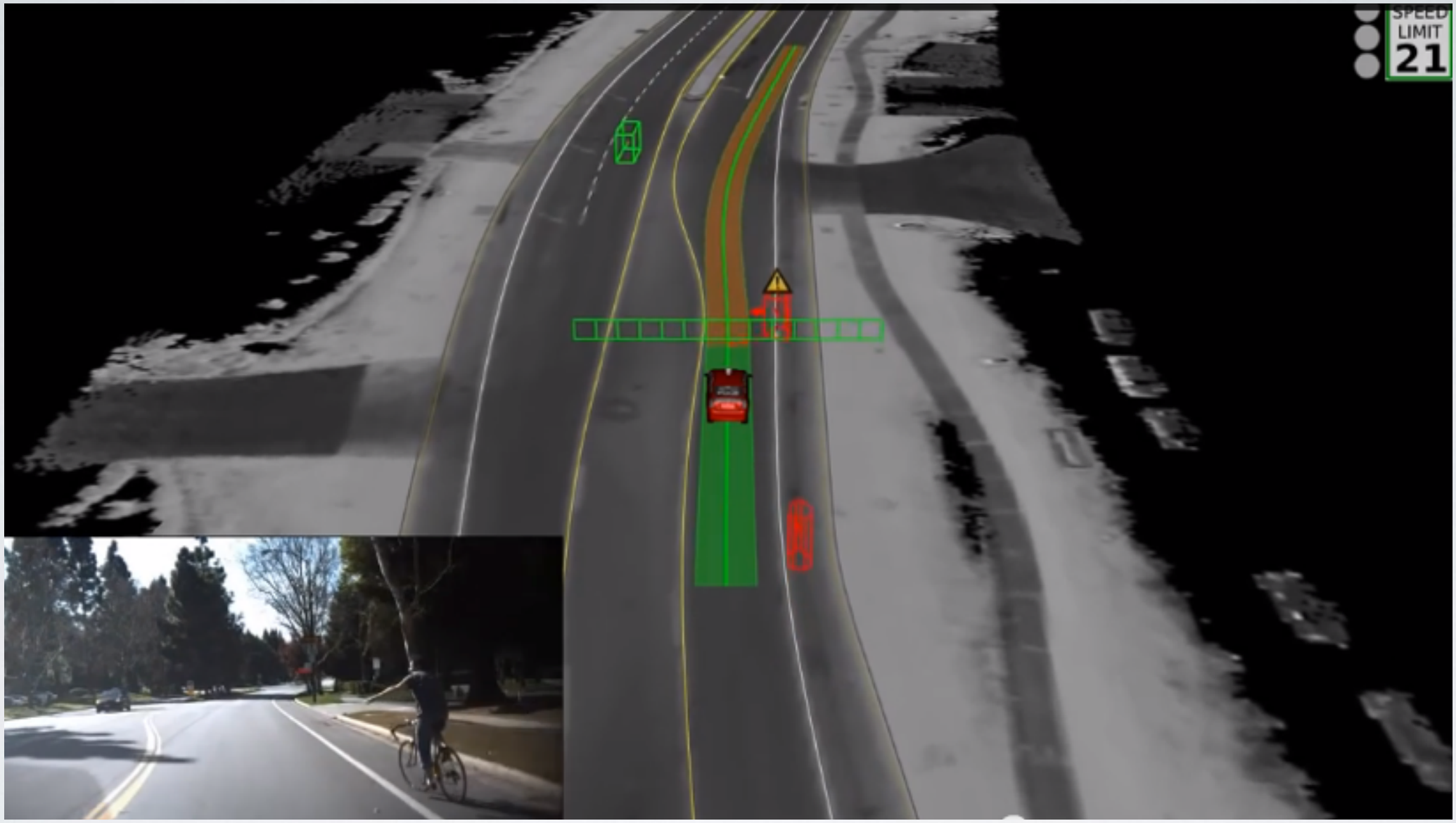
Bristol Robotics Lab

Trial 2

Trial 3

AN ETHICAL GOVERNOR

We can verify the operation of the Governor

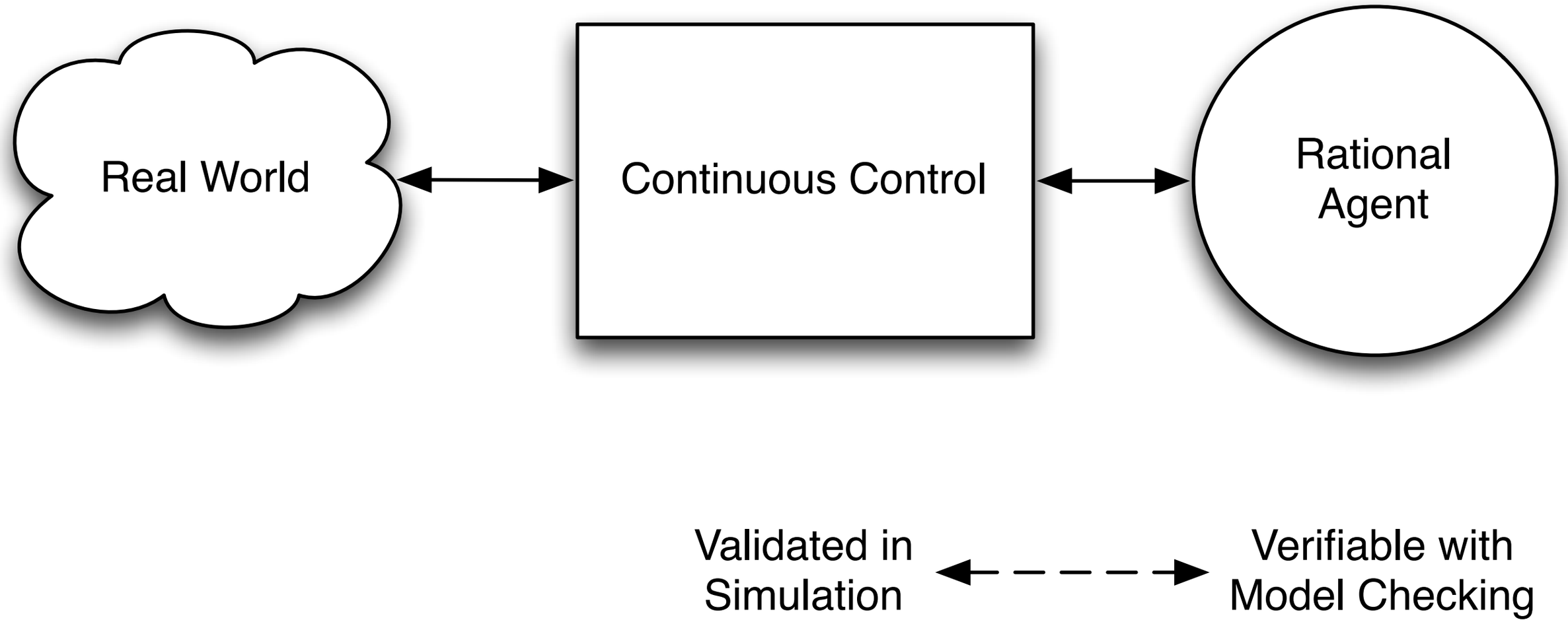


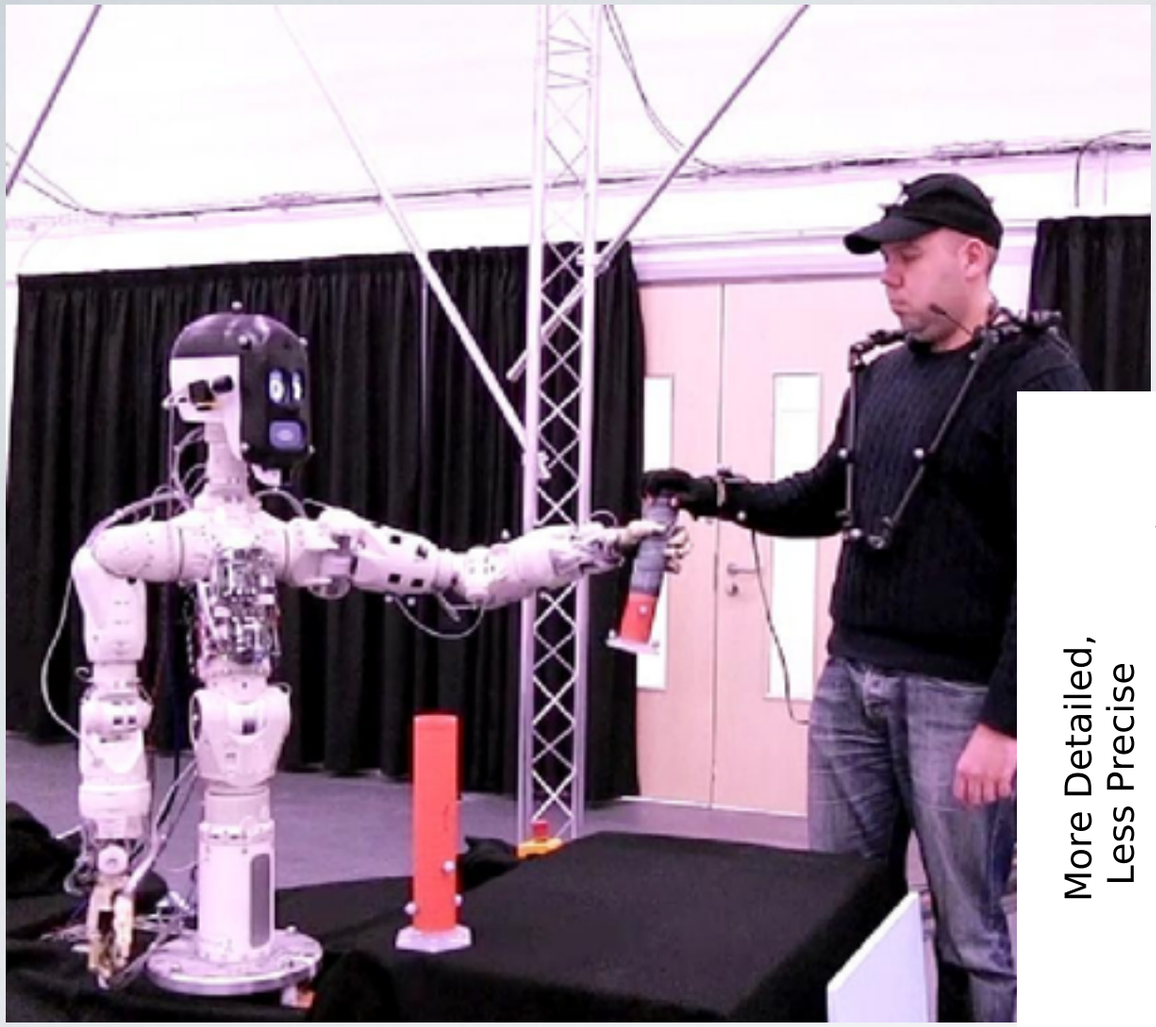
THIS IS NOT THE END OF THE STORY

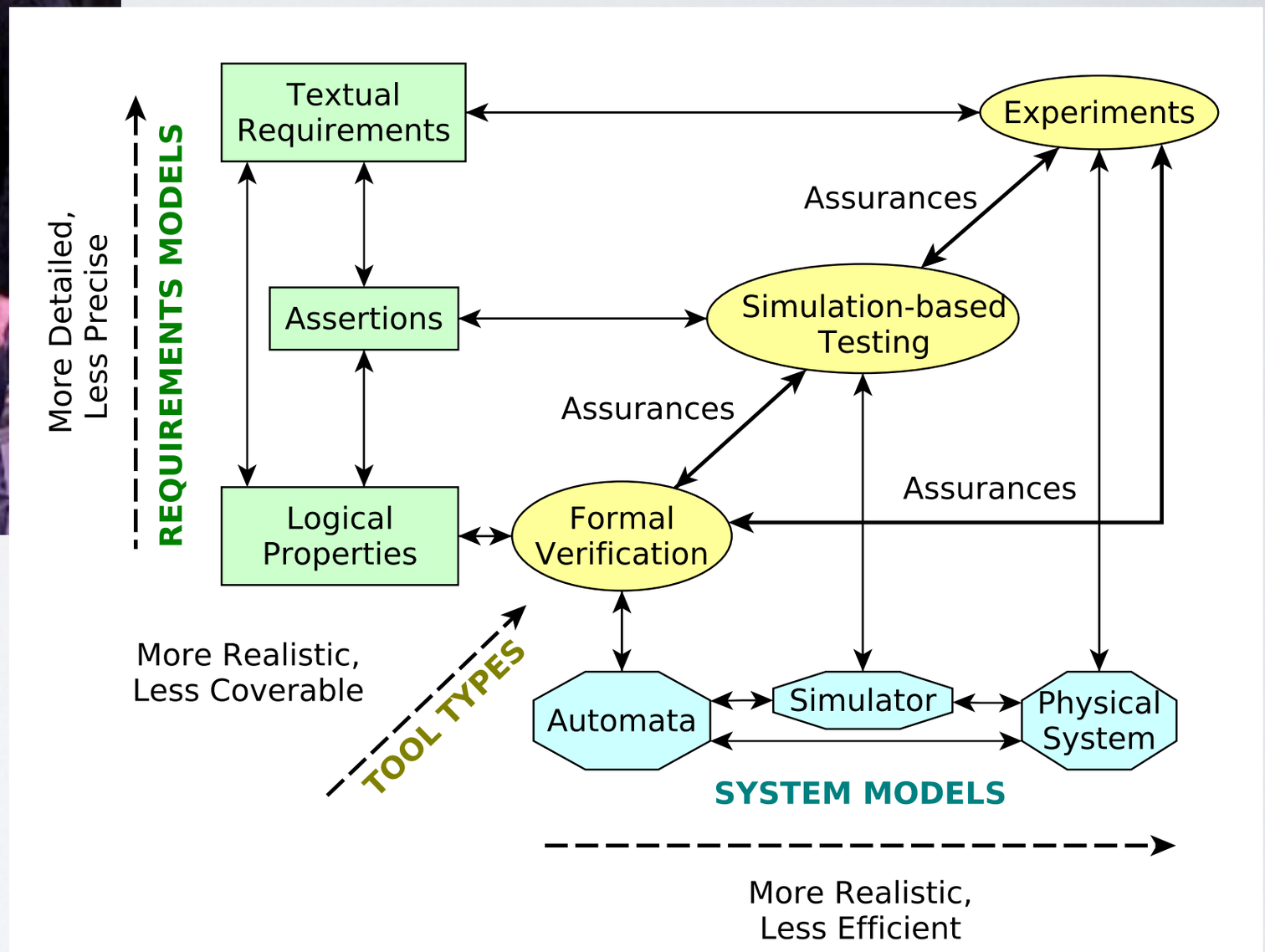
Autonomous Systems need smarter sensors, actuators and control.

AUTONOMOUS CONVOYS

The lead car is driven by a human, remaining cars drive very closely together to minimize congestion and improve fuel efficiency


Real World
Continuous Control
Rational Agent
Validated in Simulation
Verifiable with Model Checking





TRUSTWORTHY ROBOTIC ASSISTANTS

HOW CAN I KNOW A ROBOT IS SAFE

- It is going to involve a combination of proof, simulation and testing.
- Much like certification of existing safety-critical systems.
- Meanwhile robots will also be engineered to follow legal, ethical and social rules.

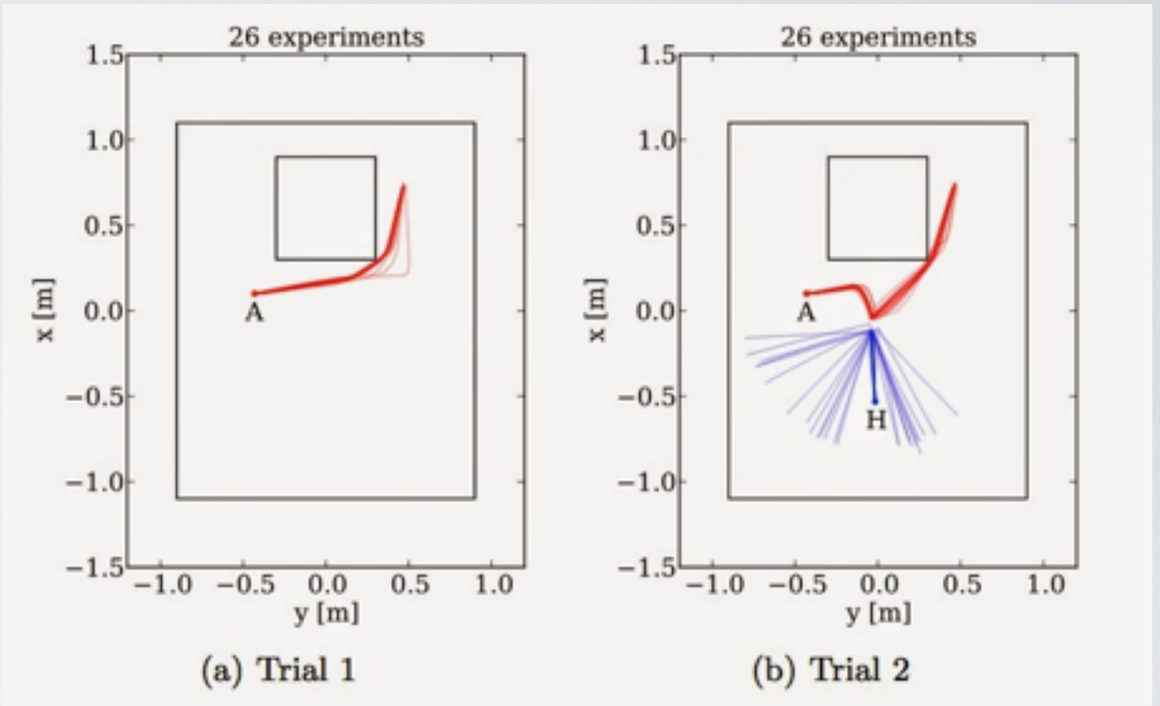
Looking Further Ahead

QUESTIONS?

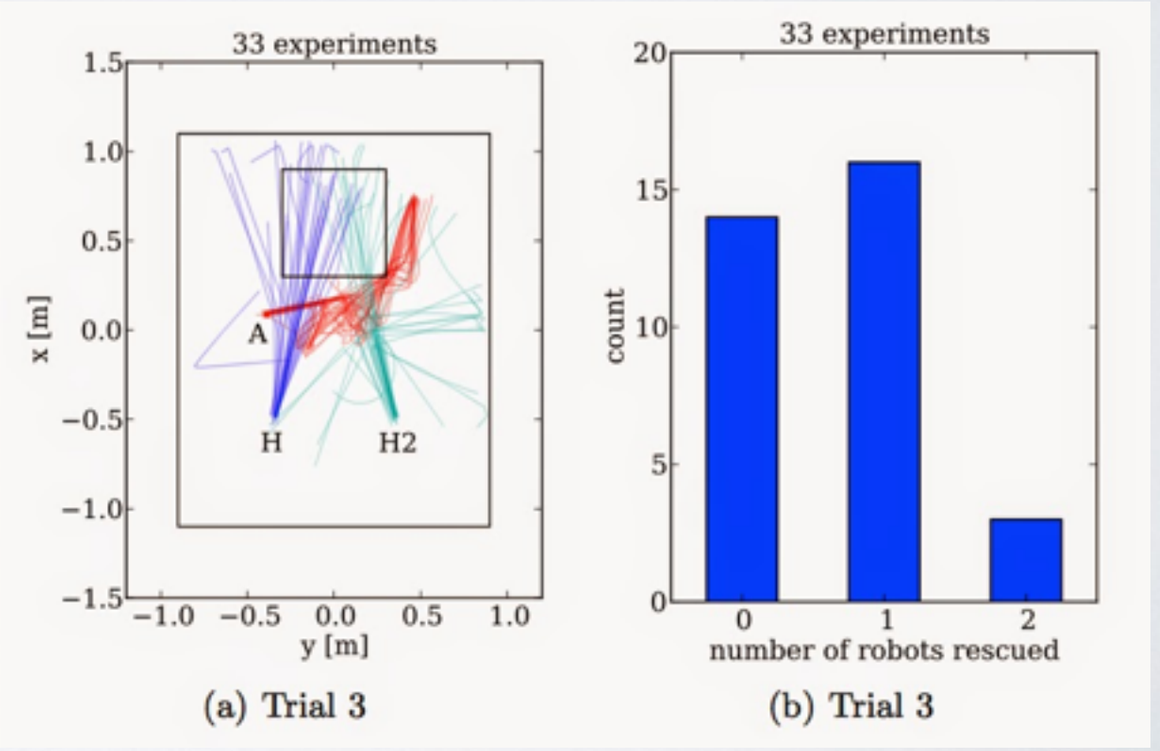
WANT TO KNOW MORE?

- Slides, web links and references for this talk can be found at http://wordpress.csc.liv.ac.uk/va/2015/04/02/eastercon-slides-and-references/.

- I'll tweet that link using the #DY66 tag as soon as I can after this talk.



(a) Trial 1 (b) Trial 2



(a) Trial 3 (b) Trial 3