جامعة الأخوين

ⵜⴰⵙⴷⴰⵡⵉⵜ ⵉⵏ ⴰⵅⴰⵡⴰⵢⵏ

AL AKHAWAYN
UNIVERSITY

Report of Project #2

Yassine Drafate

Pr. Tajjeeddine Rachidi

CSC 4301

SUMMER 2022

I- Introduction:

This project is about coding a logical agent for the Wumpus game; the goal is to build a single iteration of a logical agent that given a starting point (scenario) and asked about one single iteration (like safety of rooms, shooting Wumpus, picking up gold), the agent should then reply with an answer with respect to the rules of the Wumpus game and corresponding to the pre-defined scenario or initial condition. I used the rules that we discussed in class and SWISH Prolog for the implementation of the agent. This report includes the key predicates and the meaning of the variables, snapshots of three different scenarios along with some of the limitations of this implementation along with some possible solutions.

II- Key predicates and the meaning of the variables:

breeze([X,Y]) : is there a breeze in room(x, y) ?

stench([X,Y]) : is there a stench in room (x, y) ?

pit([X,Y]) : is there a pit in room (x, y) ?

wumpus([X,Y]) : is the Wumpus in room (x, y) ?

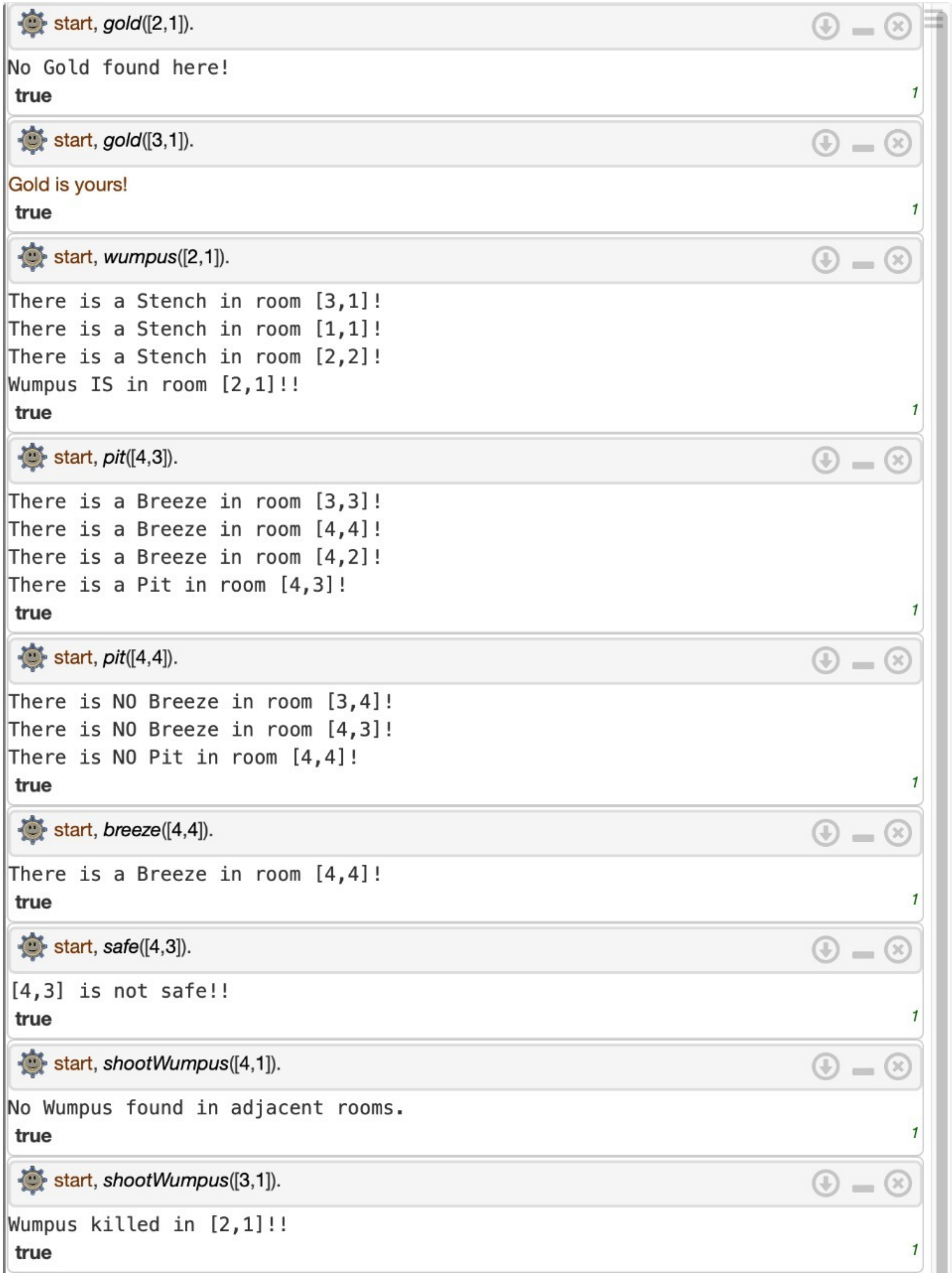
safe([X,Y]): is the room (x, y) safe ?

gold([X,Y]): is there any gold in room (x, y) ?

shootWumpus([X,Y]): attempt to shoot the wumpus in rooms adjacent to room (x, y)

III- Scenario 1:

```
retractall(isGold(_)),
assert(isGold([3,1])),
retractall(isWumpus(_)),
assert(isWumpus([2,1])),
retractall(isPit(_)),
assert(isPit([1,4])),
assert(isPit([4,3])),
retractall(agentSite(_)).
```



```
start, gold([2,1]).

No Gold found here!
true
```

```
start, gold([3,1]).

Gold is yours!
true
```

```
start, wumpus([2,1]).

There is a Stench in room [3,1]!
There is a Stench in room [1,1]!
There is a Stench in room [2,2]!
Wumpus IS in room [2,1]!!
true
```

```
start, pit([4,3]).

There is a Breeze in room [3,3]!
There is a Breeze in room [4,4]!
There is a Breeze in room [4,2]!
There is a Pit in room [4,3]!
true
```

```
start, pit([4,4]).

There is NO Breeze in room [3,4]!
There is NO Breeze in room [4,3]!
There is NO Pit in room [4,4]!
true
```

```
start, breeze([4,4]).

There is a Breeze in room [4,4]!
true
```

```
start, safe([4,3]).

[4,3] is not safe!!
true
```

```
start, shootWumpus([4,1]).

No Wumpus found in adjacent rooms.
true
```

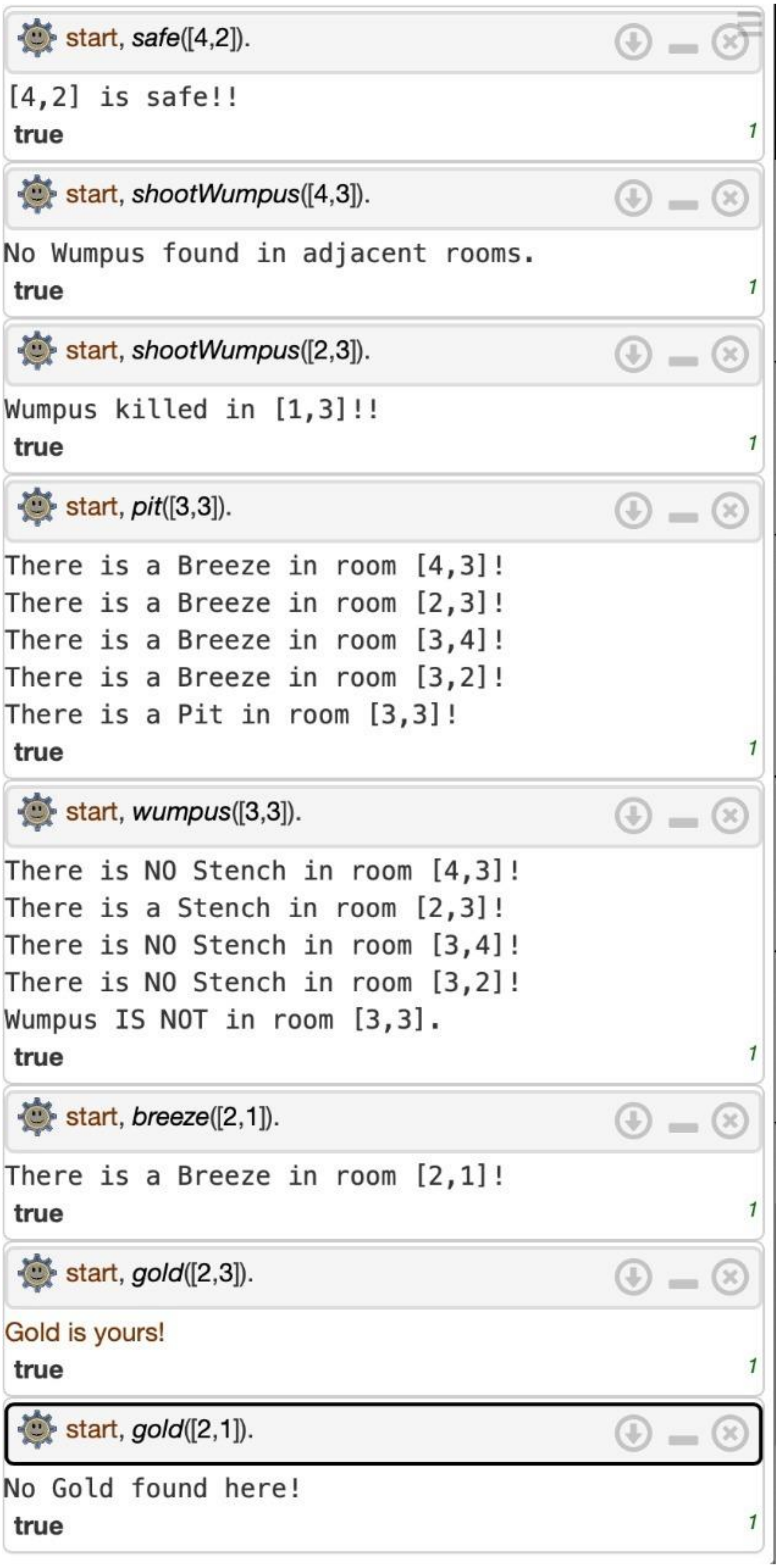
```
start, shootWumpus([3,1]).

Wumpus killed in [2,1]!!
true
```

1,4 P	2,4 B	3,4	4,4 B
1,3 B	2,3	3,3 B	4,3 P
1,2	2,2 S	3,2	4,2 B
1,1 S	2,1 W	3,1 G S	4,1

IV- Scenario 2:

```
retractall(isGold(_)),
assert(isGold([2,3])),
retractall(isWumpus(_)),
assert(isWumpus([1,3])),
retractall(isPit(_)),
assert(isPit([3,3])),
assert(isPit([3,1])),
retractall(agentSite(_)).
```



```
start, safe([4,2]).
[4,2] is safe!!
true
```

```
start, shootWumpus([4,3]).
No Wumpus found in adjacent rooms.
true
```

```
start, shootWumpus([2,3]).
Wumpus killed in [1,3]!!
true
```

```
start, pit([3,3]).
There is a Breeze in room [4,3]!
There is a Breeze in room [2,3]!
There is a Breeze in room [3,4]!
There is a Breeze in room [3,2]!
There is a Pit in room [3,3]!
true
```

```
start, wumpus([3,3]).
There is NO Stench in room [4,3]!
There is a Stench in room [2,3]!
There is NO Stench in room [3,4]!
There is NO Stench in room [3,2]!
Wumpus IS NOT in room [3,3].
true
```

```
start, breeze([2,1]).
There is a Breeze in room [2,1]!
true
```

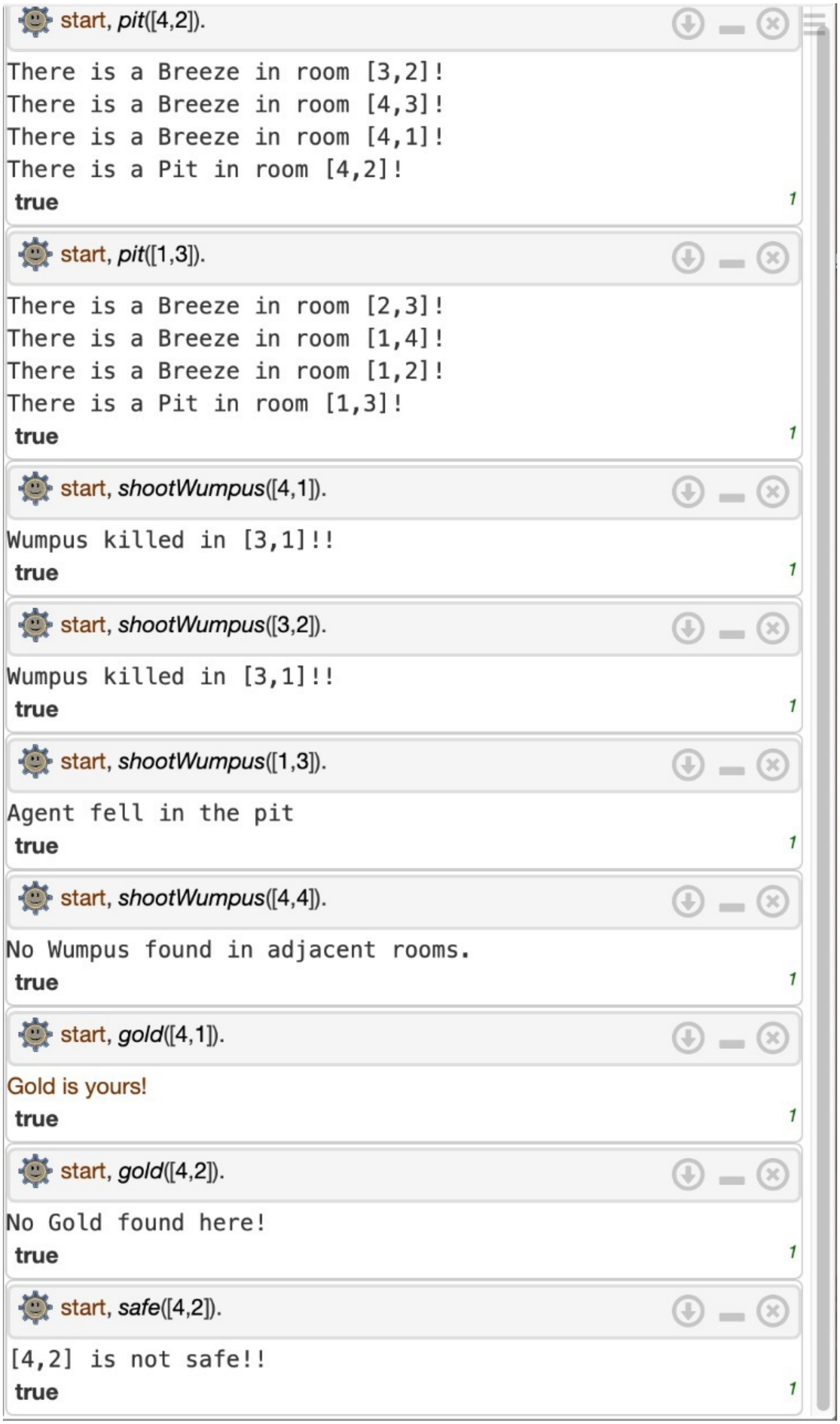
```
start, gold([2,3]).
Gold is yours!
true
```

```
start, gold([2,1]).
No Gold found here!
true
```

1,4 S	2,4	3,4 B	4,4
1,3 W	2,3 S B G	3,3 P	4,3 B
1,2 S	2,2	3,2 B	4,2
1,1	2,1 B	3,1 P	4,1 B

V- Scenario 3:

```
retractall(isGold(_)),
assert(isGold([4,1])),
retractall(isWumpus(_)),
assert(isWumpus([3,1])),
retractall(isPit(_)),
assert(isPit([4,2])),
assert(isPit([4,3])),
assert(isPit([3,4])),
assert(isPit([1,3])),
retractall(agentSite(_)).
```



```
start, pit([4,2]).

There is a Breeze in room [3,2]!
There is a Breeze in room [4,3]!
There is a Breeze in room [4,1]!
There is a Pit in room [4,2]!
true                                                        1

start, pit([1,3]).

There is a Breeze in room [2,3]!
There is a Breeze in room [1,4]!
There is a Breeze in room [1,2]!
There is a Pit in room [1,3]!
true                                                        1

start, shootWumpus([4,1]).

Wumpus killed in [3,1]!!
true                                                        1

start, shootWumpus([3,2]).

Wumpus killed in [3,1]!!
true                                                        1

start, shootWumpus([1,3]).

Agent fell in the pit
true                                                        1

start, shootWumpus([4,4]).

No Wumpus found in adjacent rooms.
true                                                        1

start, gold([4,1]).

Gold is yours!
true                                                        1

start, gold([4,2]).

No Gold found here!
true                                                        1

start, safe([4,2]).

[4,2] is not safe!!
true                                                        1
```

1,4 B	2,4 B	3,4 P	4,4 B
1,3 P	2,3 B	3,3 B	4,3 P B
1,2 B	2,2	3,2 S B	4,2 P B
1,1	2,1 S	3,1 W	4,1 G S B

VI- Limitations and remedies:

The first limitation of this implementation is that the agent location is hard coded (agent always starts in room (1, 1)), which can be fixed by either randomizing it or by taking input from the user but in both solutions, we should account for safety of the room where the agent starts; otherwise, the game would end before it even starts. The second limitation is because of the single iteration approach used in this implementation; we only won in 1 out of 3 scenarios and this can be fixed by changing the implementation so that the agent can change its location safely until it reaches a room where it can shoot the Wumpus.