جامعة الأخوين

ⵜⴰⵙⴷⴰⵡⵉⵜ ⵏ ⵓⵅⴰⵡⵉⵏ

AL AKHAWAYN UNIVERSITY

Wumpus World Project Report

Introduction To Artificial Intelligence

Summer 2022

Imane Guessous
Nada Laabid

Table of Contents

Introduction: 3
Predicates: 4
Scenarios: 6
Scenario 1: 6
Scenario 2: 8
Scenario 3: 11
Scenario 4: 14
Scenario 5: 17
Limitations and remedies: 19

Introduction:

In the game Wumpus World, an intelligent agent tries to get rid of the Wumpus that inhabits a 4x4 grid. The agent must also avoid stepping into any pits. The global map also shows the location of the gold.

The rooms next to the Wumpus seem stench-filled, whereas the rooms next to the pits seem breezy. The Wumpus can only be shot from an adjacent room.

In this version of the game, instead of being independent, the agent uses commands or predicates asked by a user to know what to do.

Predicates:

```
:- dynamic ([breeze/1,
             stench/1,
             wumpus_location/1,
             pit_location/1,
             gold_location/1,
             agent_location/1]).
```

The variables pit_location, wumpus_location, gold_location, and agent_location are world parameters set at the start of every iteration of the game.

The variables breeze and stench are dynamically asserted to the knowledge base.

adjacentTo([X,Y], L): will give all the adjacent rooms to your room [X,Y]

```
adjacentTo([X,Y],L) :- (X<4)->  Xr is X+1, L=[Xr,Y].
adjacentTo([X,Y],L) :- (X>1)->  Xl is X-1, L=[Xl,Y].
adjacentTo([X,Y],L) :- (Y<4)->  Yt is Y+1, L=[X,Yt].
adjacentTo([X,Y],L) :- (Y>1)->  Yb is Y-1, L=[X,Yb].
```

breeze([X,Y]): will tell you if there is breeze in room [X,Y]

```
breeze([X,Y]) :- (pit_location(PL),adjacentTo([X,Y],PL))->  format('There is a breeze in ~p~n',[[X,Y]]); format('There is no breeze in ~p~n',[[X,Y]]).
```

pit([X,Y]): will tell if there is a breeze in any adjacent rooms and if room [X,Y] has a pit

```
pit([X,Y]) :- forall(adjacentTo([X,Y],L), (breeze(L))), (pit_location([X,Y]))-> format('You fell in the pit!').
```

stench([X,Y]): will tell you if room [X,Y] has a stench

```
stench([X,Y]) :- (wumpus_location(L),adjacentTo([X,Y],L))->  format('there is a stench in ~p~n',[[X,Y]]); format('there is no stench in ~p~n',[[X,Y]]).
```

wumpus([X,Y]): will tell you the adjacent rooms that have a stench and if the wumpus is in room [X,Y]

```
wumpus([X,Y]) :- forall(adjacentTo([X,Y],L), stench(L)),wumpus_location([X,Y]).
```

safe([X,Y]): will tell you if room [X,Y] has no pit nor wumpus.

```
safe([X,Y]):- \+ pit_location([X,Y]), \+ wumpus_location([X,Y]).
```

gold([X,Y]): will tell you if room [X,Y] has gold.

```
gold([X,Y]):- (gold_location([X,Y]))->  format('There is gold in ~p~n',[[X,Y]]); format('There is no gold in ~p~n',[[X,Y]]).
```

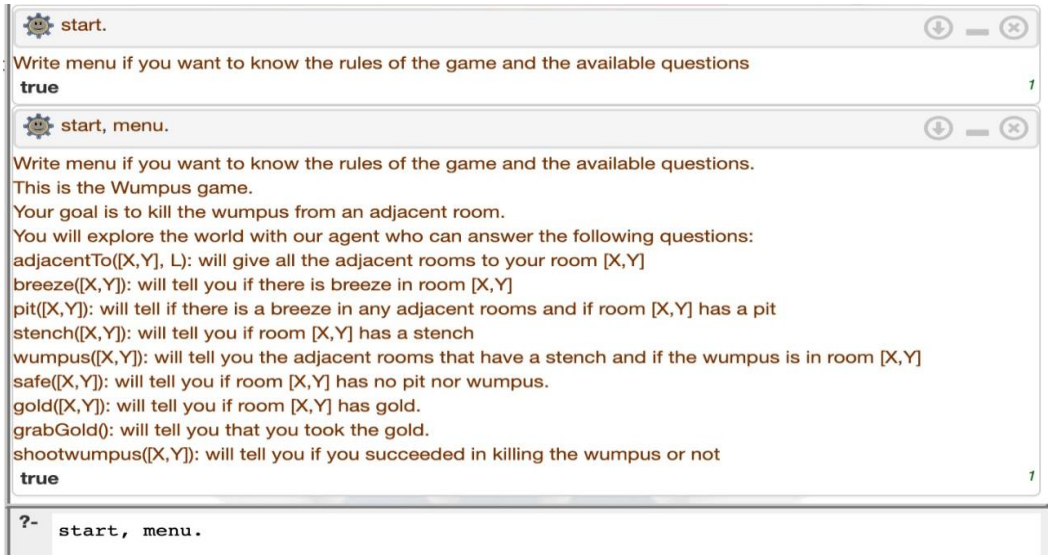
grabGold(): will tell you that you took the gold

```
grabGold():- write('We have the gold!').
```

shootwumpus([X,Y]): will tell you if you succeeded in killing the wumpus or not

```
shootwumpus([X,Y]):- (pit_location([X,Y]))-> format('You fell in a pit, and you missed your shot!'); (wumpus_location(L), adjacentTo([X,Y],L))-> format('Wumpus killed in ~p~n',[L]);
```

```
(wumpus_location([X,Y]))->  format('Wumpus ate you. You failed.');format('Wumpus not found. You failed.').
```



Scenarios:

Scenario 1:

wumpus	stench	breeze	pit
stench	breeze	pit	breeze
		Breeze, Gold	
Agent	breeze	pit	breeze

Code snippet:

```
retractall(gold_location(_)),
assert(gold_location([3,2])),
retractall(wumpus_location(_)),
assert(wumpus_location([1,4])),
retractall(pit_location(_)),
assert(pit_location([3,1])),
assert(pit_location([3,3])),
assert(pit_location([4,4])),
retractall(agent_location(_)).
```

Snapshot of the experiments:

start, *safe*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *safe*([2,1]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *safe*([1,2]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *safe*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *breeze*([1,2]).

Write menu if you want to know the rules of the game and the available questions.
There is no breeze in [1,2]
true 1

start, *breeze*([2,1]).

Write menu if you want to know the rules of the game and the available questions.
There is a breeze in [2,1]
true 1

start, *pit*([2,2]).

Write menu if you want to know the rules of the game and the available questions.
There is a breeze in [3,2]
There is no breeze in [1,2]
There is a breeze in [2,3]
There is a breeze in [2,1]
false

start, *breeze*([3,3]).

Write menu if you want to know the rules of the game and the available questions.
There is no breeze in [3,3]
true 1

start, *safe*([3,3]).

Write menu if you want to know the rules of the game and the available questions.
false

start, *stench*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
there is a stench in [1,3]
true 1

start, *wumpus*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
there is no stench in [2,3]
there is no stench in [1,4]
there is no stench in [1,2]
false

start, *safe*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
false

start, *shootwumpus*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
Wumpus killed in [1,4]

Scenario 2:

stench			breeze
wumpus	stench	Breeze	pit
pit	Breeze		Breeze
Agent	Gold	Breeze	pit

Code snippet:

```
retractall(gold_location(_)),
assert(gold_location([2,1])),
retractall(wumpus_location(_)),
assert(wumpus_location([1,3])),
retractall(pit_location(_)),
assert(pit_location([4,3])),
assert(pit_location([1,2])),
assert(pit_location([4,1])),
retractall(agent_location(_)).
```

Snapshot of the experiments:

start, *safe*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
true

start, *safe*([1,2]).

Write menu if you want to know the rules of the game and the available questions.
false

start, *safe*([2,1]).

Write menu if you want to know the rules of the game and the available questions.
true

start, *stench*([2,1]).

Write menu if you want to know the rules of the game and the available questions.
there is no stench in [2,1]
true

start, *stench*([1,2]).

Write menu if you want to know the rules of the game and the available questions.
there is a stench in [1,2]
true

start, *stench*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
there is no stench in [1,1]
true

start, *safe*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
false

start, *breeze*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
There is a breeze in [1,3]
true

start, *breeze*([1,2]).

Write menu if you want to know the rules of the game and the available questions.
There is no breeze in [1,2]
true

start, *pit*([2,1]).

Write menu if you want to know the rules of the game and the available questions.
There is a breeze in [3,1]
There is a breeze in [1,1]
There is a breeze in [2,2]
false

start, *safe*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
true

start, *stench*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
there is a stench in [1,4]
true

start, *gold*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
There is no gold in [1,4]
true

start, *gold*([2,4]).

Write menu if you want to know the rules of the game and the available questions.
There is no gold in [2,4]
true *1*

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

Write menu if you want to know the rules of the game and the available questions.
There is gold in [2,1]
true *1*

start, grabGold.

Write menu if you want to know the rules of the game and the available questions.
We have the gold!
true *1*

start, *shootwumpus*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
Wumpus killed in [1,3]
true *1*

Scenario 3:

breeze	pit	breeze	
Stench, pit	breeze	breeze	gold
wumpus	stench	pit	breeze
Agent, Stench	breeze	pit	breeze

Code snippet:

```
retractall(gold_location(_)),
assert(gold_location([3,4])),
retractall(wumpus_location(_)),
assert(wumpus_location([1,2])),
retractall(pit_location(_)),
assert(pit_location([3,2])),
assert(pit_location([3,1])),
assert(pit_location([2,4])),
assert(pit_location([1,3])),
retractall(agent_location(_)).
```

Snapshot of the experiments:

start, *safe*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *pit*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
There is a breeze in [2,1]
There is a breeze in [1,2]
false

start, *wumpus*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
there is no stench in [2,1]
there is no stench in [1,2]
false

start, *safe*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
false

start, *safe*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *pit*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
There is no breeze in [2,4]
There is no breeze in [1,3]
false

start, *gold*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
There is no gold in [1,4]
true 1

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

Write menu if you want to know the rules of the game and the available questions.
There is no gold in [1,2]
true 1

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

Write menu if you want to know the rules of the game and the available questions.
There is no gold in [2,1]
true 1

start, *safe*([2,2]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *pit*([2,2]).

Write menu if you want to know the rules of the game and the available questions.
There is a breeze in [3,2]
There is a breeze in [1,2]
There is a breeze in [2,3]
There is a breeze in [2,1]
false

start, *safe*([3,1]).

Write menu if you want to know the rules of the game and the available questions.

false

start, *wumpus*([2,2]).

Write menu if you want to know the rules of the game and the available questions.

```
there is no stench in [3,2]
there is no stench in [1,2]
there is no stench in [2,3]
there is no stench in [2,1]
```

false

start, *wumpus*([1,4]).

Write menu if you want to know the rules of the game and the available questions.

```
there is no stench in [2,4]
there is a stench in [1,3]
```

false

start, *shootwumpus*([1,3]).

Write menu if you want to know the rules of the game and the available questions.

```
You fell in a pit, and you missed your shot!
```

true

1

Scenario 4:

wumpus	breeze		
breeze	pit	breeze	
gold	breeze		
Agent	pit	breeze	

Code snippet:

```
retractall(gold_location(_)),
assert(gold_location([1,2])),
retractall(wumpus_location(_)),
assert(wumpus_location([1,4])),
retractall(pit_location(_)),
assert(pit_location([2,3])),
assert(pit_location([2,1])),
retractall(agent_location(_)).
```

Snapshot of the experiments:

start, *safe*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *pit*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
There is no breeze in [2,1]
There is no breeze in [1,2]
false

start, *wumpus*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
there is no stench in [2,1]
there is no stench in [1,2]
false

start, *breeze*([1,1]).

Write menu if you want to know the rules of the game and the available questions.
There is a breeze in [1,1]
true 1

start, *safe*([1,2]).

Write menu if you want to know the rules of the game and the available questions.
true 1

start, *safe*([2,1]).

Write menu if you want to know the rules of the game and the available questions.
false

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

Write menu if you want to know the rules of the game and the available questions.
there is no stench in [2,2]
there is a stench in [1,3]
there is no stench in [1,1]
false

start, *safe*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
true 1

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

Write menu if you want to know the rules of the game and the available questions.
There is no gold in [1,3]
true 1

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

Write menu if you want to know the rules of the game and the available questions.
There is gold in [1,2]
true 1

start, grabGold.

Write menu if you want to know the rules of the game and the available questions.
We have the gold!
true 1

start, *safe*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
false

start, *safe*([1,4]).

Write menu if you want to know the rules of the game and the available questions.
false

start, *shootwumpus*([1,3]).

Write menu if you want to know the rules of the game and the available questions.
`Wumpus killed in [1,4]`
true

Scenario 5:

	Breeze	Breeze	pit
Breeze	pit	pit	gold
pit	Breeze, stench	wumpus	Breeze, stench
Agent, breeze	pit	Breeze, stench	pit

Code snippet:

```
retractall(gold_location(_)),
assert(gold_location([4,3])),
retractall(wumpus_location(_)),
assert(wumpus_location([3,2])),
retractall(pit_location(_)),
assert(pit_location([4,4])),
assert(pit_location([4,1])),
assert(pit_location([3,3])),
assert(pit_location([2,3])),
assert(pit_location([2,1])),
assert(pit_location([1,2])),
retractall(agent_location(_)).
```

Snapshot of the experiments:

start, *safe*([1,1]).

Write menu if you want to know the rules of the game and the available questions.

true 1

start, *pit*([1,1]).

Write menu if you want to know the rules of the game and the available questions.

```
There is no breeze in [2,1]
There is no breeze in [1,2]
```

false

start, *wumpus*([1,1]).

Write menu if you want to know the rules of the game and the available questions.

```
there is no stench in [2,1]
there is no stench in [1,2]
```

false

start, *breeze*([1,1]).

Write menu if you want to know the rules of the game and the available questions.

```
There is a breeze in [1,1]
```

true 1

start, *pit*([1,2]).

Write menu if you want to know the rules of the game and the available questions.

```
There is a breeze in [2,2]
There is a breeze in [1,3]
There is a breeze in [1,1]
You fell in the pit!
```

true 1

Limitations and remedies:

After running the code for 5 different configurations, we won 3 out of 5 times, as you can see in the scenarios and snapshots above. The reasons for the limitations are:

The human brain:

- our agent takes questions from the users and the user is the one who decided what to do next.
- Users make mistakes as we can see in scenario 3, where the mistake was from the human actions.

The configuration is impossible to solve:

- In the fifth scenario, both adjacent rooms and only options for the agent contains pits. It was therefore impossible to win in this case.

Eventual future remedies to the limitations:

- Having the intelligent agent make the best action instead of relying on the human to do so.
- As for the configuration problem, it would have been difficult and impossible to solve either way.