Wumpus World - Improvements

- Alexandre Dietrich
- Patrick Chabelski
- Rodolfo Vasconcelos

SCS 3547 – Intelligent Agents & Reinforcement Learning

UNIVERSITY OF TORONTO - SCHOOL OF CONTINUING STUDIES

Instructor: Larry Simon

July, 5th, 2020

GitHub: https://github.com/ravasconcelos/wumpus_world/ (https://github.com/ravasconcelos/wumpus_world/)

About the Project

This Prolog implementation of Wumpus World has some improvements afer the analysis made on the original code.

More details can be found in this other Notebook: https://swish.swi-prolog.org/p/Wumpus%20World%20-%20Explained.swinb (https://swish.swi-prolog.org/p/Wumpus%20World%20-%20Explained.swinb)

The original code is implemented by Richard O. Legendi and can be found at https://github.com/rlegendi/wumpus-prolog/ (https://github.com/rlegendi/wumpus-prolog/)

Main changes:

1. Change the gold location as per Fig. 7.2 (https://raw.githubusercontent.com/ravasconcelos/wumpus_world/master/images/figure_7_2.png)
2. Remove unused code
3. Create a KB for nodes that are "OK", which can be safely trasversed
4. Ask_KB:
    1. Consult preffered moves
    2. Let to revisit a location
    3. Check for adjacent cells

Create a Program | Query | Markdown | HTML **cell here**

```
?- start.
```

```
Initializing started...
Let the game begin!



New Round: I am at [1,1] and I have visieted []
make_percept_sentence... _63060,_63066,_63072
smelly=no
```

```
Stench... no
I'm in [1,1], seeing: [no,no,no]
update_KB [no,no,no]
add_wumpus_KB no
Not visited before= [1,1]
KB learn [1,2] - no Wumpus there!
[1,0] is not permitted
KB learn [2,1] - no Wumpus there!
[0,1] is not permitted
add_pit_KB no
Not visited before= [1,1]
KB learn [1,2] - there's no Pit there!
[1,0] is not permitted
KB learn [2,1] - there's no Pit there!
[0,1] is not permitted
add_gold_KB no
KB learn [1,1] - there's no gold here!
add_ok_KB no,no
Not visited before= [1,1]
add_ok_KB_item [1,2]
assume_ok(no,no,L) [1,2]
KB learn [1,2] is OK
add_ok_KB_item [1,0]
[1,0] is not permitted
add_ok_KB_item [2,1]
assume_ok(no,no,L) [2,1]
KB learn [2,1] is OK
add_ok_KB_item [0,1]
[0,1] is not permitted
ask_KB VisitedList=[] - Action=_80878
agent_location=[1,1]
preferred_move - Let's see if [1,1] is OK
preferred_move - L was not visited before= [1,1]
Let's see if [1,1] is OK
[1,1] is the preffered move
preferred_move - Let's see if [1,2] is OK
preferred_move - L was not visited before= [1,2]
Let's see if [1,2] is OK
[1,2] is the preffered move
L=[1,2] is adjacent to AL_=[1,1]
There is no loop
New Agent Location: [1,2]
Assume L as the new location= [1,2]
I'm going to: [1,2]
New time: 1
New score: -1
```

```
VisitedList = [[1,1]]
There's still something to do...



New Round: I am at [1,2] and I have visieted [[1,1]]
make_percept_sentence... _91076,_91082,_91088
smelly=no
Stench... no
I'm in [1,2], seeing: [no,yes,no]
update_KB [no,yes,no]
add_wumpus_KB no
Not visited before= [1,2]
KB learn [1,3] - no Wumpus there!
KB learn [1,1] - no Wumpus there!
KB learn [2,2] - no Wumpus there!
[0,2] is not permitted
add_pit_KB yes
Not visited before= [1,2]
KB learn [1,3] - is it a Pit?
KB learn [1,3] - maybe there is a Pit!
KB learn [1,1] - is it a Pit?
I know there is no Pit at [1,1]!
KB learn [2,2] - is it a Pit?
KB learn [2,2] - maybe there is a Pit!
[0,2] is not permitted
add_gold_KB no
KB learn [1,2] - there's no gold here!
add_ok_KB no,yes
Not visited before= [1,2]
add_ok_KB_item [1,3]
assume_ok(no,maybe,L) [1,3]
KB learn [1,3] is NOT OK
add_ok_KB_item [1,1]
assume_ok(no,no,L) [1,1]
KB learn [1,1] is OK
add_ok_KB_item [2,2]
assume_ok(no,maybe,L) [2,2]
KB learn [2,2] is NOT OK
add_ok_KB_item [0,2]
[0,2] is not permitted
ask_KB VisitedList=[[1,1]] - Action=_111328
agent_location=[1,2]
preferred_move - Let's see if [2,1] is OK
preferred_move - L was not visited before= [2,1]
Let's see if [2,1] is OK
```

```
[2,1] is the preffered move
preferred_move - Let's see if [1,2] is OK
preferred_move - L was not visited before= [1,2]
Let's see if [1,2] is OK
[1,2] is the preffered move
preferred_move - Let's see if [1,1] is OK
preferred_move - Let's see if [2,1] is OK
Let's see if [2,1] is OK
[2,1] is NOT the preffered mov
preferred_move - Let's see if [1,2] is OK
Let's see if [1,2] is OK
[1,2] is NOT the preffered mov
preferred_move - Let's see if [1,1] is OK
Let's see if [1,1] is OK
[1,1] is NOT the preffered mov
L=[1,1] is adjacent to AL_=[1,2]
There is no loop
New Agent Location: [1,1]
Assume L as the new location= [1,1]
I'm going to: [1,1]
New time: 2
New score: -2
VisitedList = [[1,2],[1,1]]
There's still something to do...



New Round: I am at [1,1] and I have visieted [[1,2],[1,1]]
make_percept_sentence... _120970,_120976,_120982
smelly=no
Stench... no
I'm in [1,1], seeing: [no,no,no]
update_KB [no,no,no]
add_wumpus_KB no
Already visited before= [1,1]
add_pit_KB no
Already visited before= [1,1]
add_gold_KB no
KB learn [1,1] - there's no gold here!
add_ok_KB no,no
Already visited before= [1,1]
ask_KB VisitedList=[[1,2],[1,1]] - Action=_128384
agent_location=[1,1]
preferred_move - Let's see if [2,1] is OK
preferred_move - L was not visited before= [2,1]
Let's see if [2,1] is OK
```

```
[2,1] is the preffered move
L=[2,1] is adjacent to AL_=[1,1]
There is no loop
New Agent Location: [2,1]
Assume L as the new location= [2,1]
I'm going to: [2,1]
New time: 3
New score: -3
VisitedList = [[1,1],[1,2],[1,1]]
There's still something to do...



New Round: I am at [2,1] and I have visieted [[1,1],[1,2],[1,1]]
make_percept_sentence... _22520,_22526,_22532
smelly=yes
Stench... yes
I'm in [2,1], seeing: [yes,no,no]
update_KB [yes,no,no]
add_wumpus_KB yes
Not visited before= [2,1]
KB learn [2,2] - is it a Wumpus?
I know there is no Wumpus at [2,2]!
[2,0] is not permitted
KB learn [3,1] - is it a Wumpus?
KB learn [3,1] - maybe there is a Wumpus!
KB learn [1,1] - is it a Wumpus?
I know there is no Wumpus at [1,1]!
add_pit_KB no
Not visited before= [2,1]
KB learn [2,2] - there's no Pit there!
[2,0] is not permitted
KB learn [3,1] - there's no Pit there!
KB learn [1,1] - there's no Pit there!
add_gold_KB no
KB learn [2,1] - there's no gold here!
add_ok_KB yes,no
Not visited before= [2,1]
add_ok_KB_item [2,2]
assume_ok(no,no,L) [2,2]
KB learn [2,2] is OK
add_ok_KB_item [2,0]
[2,0] is not permitted
add_ok_KB_item [3,1]
assume_ok(maybe,no,L) [3,1]
KB learn [3,1] is NOT OK
```

```
add_ok_KB_item [1,1]
assume_ok(no,no,L) [1,1]
KB learn [1,1] is OK
ask_KB VisitedList=[[1,1],[1,2],[1,1]] - Action=_42840
agent_location=[2,1]
preferred_move - Let's see if [1,2] is OK
preferred_move - Let's see if [2,1] is OK
preferred_move - L was not visited before= [2,1]
Let's see if [2,1] is OK
[2,1] is the preffered move
preferred_move - Let's see if [2,2] is OK
preferred_move - L was not visited before= [2,2]
Let's see if [2,2] is OK
[2,2] is the preffered move
L=[2,2] is adjacent to AL_=[2,1]
There is no loop
New Agent Location: [2,2]
Assume L as the new location= [2,2]
I'm going to: [2,2]
New time: 4
New score: -4
VisitedList = [[2,1],[1,1],[1,2],[1,1]]
There's still something to do...



New Round: I am at [2,2] and I have visieted [[2,1],[1,1],[1,2],[1,1]]
make_percept_sentence... _53118,_53124,_53130
smelly=no
Stench... no
I'm in [2,2], seeing: [no,no,no]
update_KB [no,no,no]
add_wumpus_KB no
Not visited before= [2,2]
KB learn [2,3] - no Wumpus there!
KB learn [2,1] - no Wumpus there!
KB learn [3,2] - no Wumpus there!
KB learn [1,2] - no Wumpus there!
add_pit_KB no
Not visited before= [2,2]
KB learn [2,3] - there's no Pit there!
KB learn [2,1] - there's no Pit there!
KB learn [3,2] - there's no Pit there!
KB learn [1,2] - there's no Pit there!
add_gold_KB no
KB learn [2,2] - there's no gold here!
```

```
add_ok_KB no,no
Not visited before= [2,2]
add_ok_KB_item [2,3]
assume_ok(no,no,L) [2,3]
KB learn [2,3] is OK
add_ok_KB_item [2,1]
assume_ok(no,no,L) [2,1]
KB learn [2,1] is OK
add_ok_KB_item [3,2]
assume_ok(no,no,L) [3,2]
KB learn [3,2] is OK
add_ok_KB_item [1,2]
assume_ok(no,no,L) [1,2]
KB learn [1,2] is OK
ask_KB VisitedList=[[2,1],[1,1],[1,2],[1,1]] - Action=_72424
agent_location=[2,2]
preferred_move - Let's see if [1,1] is OK
preferred_move - Let's see if [2,2] is OK
preferred_move - L was not visited before= [2,2]
Let's see if [2,2] is OK
[2,2] is the preffered move
preferred_move - Let's see if [2,3] is OK
preferred_move - L was not visited before= [2,3]
Let's see if [2,3] is OK
[2,3] is the preffered move
L=[2,3] is adjacent to AL_=[2,2]
There is no loop
New Agent Location: [2,3]
Assume L as the new location= [2,3]
I'm going to: [2,3]
New time: 5
New score: -5
VisitedList = [[2,2],[2,1],[1,1],[1,2],[1,1]]
There's still something to do...



New Round: I am at [2,3] and I have visieted [[2,2],[2,1],[1,1],[1,2],[1,1]]
make_percept_sentence... _82724,_82730,_82736
smelly=no
Stench... no
I'm in [2,3], seeing: [no,yes,no]
update_KB [no,yes,no]
add_wumpus_KB no
Not visited before= [2,3]
KB learn [2,4] - no Wumpus there!
```

```
KB learn [2,2] - no Wumpus there!
KB learn [3,3] - no Wumpus there!
KB learn [1,3] - no Wumpus there!
add_pit_KB yes
Not visited before= [2,3]
KB learn [2,4] - is it a Pit?
KB learn [2,4] - maybe there is a Pit!
KB learn [2,2] - is it a Pit?
I know there is no Pit at [2,2]!
KB learn [3,3] - is it a Pit?
KB learn [3,3] - maybe there is a Pit!
KB learn [1,3] - is it a Pit?
KB learn [1,3] - maybe there is a Pit!
add_gold_KB no
KB learn [2,3] - there's no gold here!
add_ok_KB no,yes
Not visited before= [2,3]
add_ok_KB_item [2,4]
assume_ok(no,maybe,L) [2,4]
KB learn [2,4] is NOT OK
add_ok_KB_item [2,2]
assume_ok(no,no,L) [2,2]
KB learn [2,2] is OK
add_ok_KB_item [3,3]
assume_ok(no,maybe,L) [3,3]
KB learn [3,3] is NOT OK
add_ok_KB_item [1,3]
assume_ok(no,maybe,L) [1,3]
KB learn [1,3] is NOT OK
ask_KB VisitedList=[[2,2],[2,1],[1,1],[1,2],[1,1]] - Action=_104370
agent_location=[2,3]
preferred_move - Let's see if [1,1] is OK
preferred_move - Let's see if [2,1] is OK
preferred_move - Let's see if [3,2] is OK
preferred_move - L was not visited before= [3,2]
Let's see if [3,2] is OK
[3,2] is the preffered move
preferred_move - Let's see if [1,2] is OK
preferred_move - Let's see if [2,3] is OK
preferred_move - L was not visited before= [2,3]
Let's see if [2,3] is OK
[2,3] is the preffered move
preferred_move - Let's see if [2,2] is OK
preferred_move - Let's see if [1,1] is OK
Let's see if [1,1] is OK
[1,1] is NOT the preffered mov
```

```
preferred_move - Let's see if [2,1] is OK
Let's see if [2,1] is OK
[2,1] is NOT the preffered mov
preferred_move - Let's see if [3,2] is OK
Let's see if [3,2] is OK
[3,2] is NOT the preffered mov
preferred_move - Let's see if [1,2] is OK
Let's see if [1,2] is OK
[1,2] is NOT the preffered mov
preferred_move - Let's see if [2,3] is OK
Let's see if [2,3] is OK
[2,3] is NOT the preffered mov
preferred_move - Let's see if [2,2] is OK
Let's see if [2,2] is OK
[2,2] is NOT the preffered mov
L=[2,2] is adjacent to AL_=[2,3]
There is no loop
New Agent Location: [2,2]
Assume L as the new location= [2,2]
I'm going to: [2,2]
New time: 6
New score: -6
VisitedList = [[2,3],[2,2],[2,1],[1,1],[1,2],[1,1]]
There's still something to do...



New Round: I am at [2,2] and I have visieted [[2,3],[2,2],[2,1],[1,1],[1,2],[1,1]]
make_percept_sentence... _114066,_114072,_114078
smelly=no
Stench... no
I'm in [2,2], seeing: [no,no,no]
update_KB [no,no,no]
add_wumpus_KB no
Already visited before= [2,2]
add_pit_KB no
Already visited before= [2,2]
add_gold_KB no
KB learn [2,2] - there's no gold here!
add_ok_KB no,no
Already visited before= [2,2]
ask_KB VisitedList=[[2,3],[2,2],[2,1],[1,1],[1,2],[1,1]] - Action=_121480
agent_location=[2,2]
preferred_move - Let's see if [1,1] is OK
preferred_move - Let's see if [2,1] is OK
preferred_move - Let's see if [3,2] is OK
```

```
preferred_move - L was not visited before= [3,2]
Let's see if [3,2] is OK
[3,2] is the preffered move
L=[3,2] is adjacent to AL_=[2,2]
There is no loop
New Agent Location: [3,2]
Assume L as the new location= [3,2]
I'm going to: [3,2]
New time: 7
New score: 994
VisitedList = [[2,2],[2,3],[2,2],[2,1],[1,1],[1,2],[1,1]]
AGENT FOUND THE GOLD!!WON!
Score: 994,
 Time: 7
```

true *1*

0.128 seconds cpu time

```
%------------------------------------------------------------------------------
% A Prolog Implementation of the Wumpus World based on the
% work developed by Richard O. Legendi
% https://github.com/rlegendi/wumpus-prolog/
%
% - Alexandre Dietrich
% - Patrick Chabelski
% - Rodolfo Vasconcelos
%
% SCS 3547 – Intelligent Agents & Reinforcement Learning
%
% UNIVERSITY OF TORONTO - SCHOOL OF CONTINUING STUDIES
%
% Instructor: Larry Simon
%
% July, 5th, 2020
%
% Usage:
% consult this file
% ?-start.
%
%------------------------------------------------------------------------------


%------------------------------------------------------------------------------
% Declaring dynamic methods

:- dynamic ([
```