Introduction to Artificial Intelligence for Non Computing

Practical 4 (weeks 7 - 8)

Theory Questions

1.Symbolize the following proposition and discuss the truth.

1. Everyone has black hair.
2. Some people boarded the moon.
3. No one has boarded Jupiter
4. Students studying in the US are not necessarily Asians.

your answer here...

---

1. False:
    - A(x) : x is a people.
    - B(x) : x with black hair.
    - x(A(x)⇒B(x)) is false .
    - because there are somebody have a gold hair.
    - so M(a)⇒F(a) is false,
    - Therefore Propositional is false.
2. Ture:
    - G(x) : x boarded the moon.
    - x(A(x)∧G(x)) is Ture.
    - Neil Armstrong completed the Apollo program on the moon in 1969.
    - so A(a)∧G(a) is true.
    - Therefore Propositional is true.
3. Ture
    - H(x) : x boarded the Jupiter.
    - ¬x(M(x)∧H(x)).
    - No one has boarded Jupiter so far.
    - assume anyone is a,then M(a) ∧H(a) of all is false.
    - Therefore x(M(x)∧H(x)) is false.
    - so Propositional is true.
4. Ture
    - F(x) : x is a student studying in the US.
    - G(x) : x is Asian.
    - ¬x(F(x)⇒G(x)).
    - Therefore Propositional is true.

2.Judge the following formula, which is tautology? What is the contradiction?

1. ∀xF(x)⇒(∃x∃yG(x,y))⇒∀xF(x))
2. ¬(∀xF(x)⇒∃yG(y))∧∃yG(y)
3. ∀x(F(x)⇒G(y))

your answer here...

---

1. tautology :
   - It is the same with p⇒(q⇒p)
2. contradiction:
   - It is the same with ¬p⇒(q⇒p)
3. - 1: if F(x):x>2, G(x):x>1, formula is true.

- 2:

```
if F(x):x<2, G(x):x<1, formula is false.
```

- in conclusion:Non permanent satisfaction

3.Which of the following are correct?

1. False |=True.
2. (A ∧ B) |= (A ⇔ B).
3. (A ∧ B) ⇒ C |= (A ⇒ C) ∨ (B ⇒ C).
4. (A ∨ B) ∧ (¬C ∨¬D ∨ E) |= (A ∨ B).
5. (A ∨ B) ∧ (¬C ∨¬D ∨ E) |= (A ∨ B) ∧ (¬D ∨ E).

your answer here...

---

1. True:
   - False has no models
2. True:
   - (A ∧ B) has exactly one model that is one of the two models of (A ⇔ B).
3. True:
   - because (A ∧ B) ⇒ C is false only when both disjuncts are false
4. True:
   - it removing a conjunct only allows more models.
5. False:
   - removing a disjunct allows fewer models.

4.Conjunctive normal form.link:https://baike.baidu.com/item/%E5%90%88%E5%8F%96%E8%8C%83%E5%BC%8F/2459360

1. Obtaining conjunctive paradigm: P∧(Q⇒R)⇒S

Basic steps to find a conjunctive normal form.

1. Cut redundant connectives，Reserved {∨，∧，¬}
2. Move or remove the negation ~

3. distribution rates

your answer here...

- $P \land (Q \Rightarrow R) \Rightarrow S$
- $= \neg(P \land (\neg Q \lor R)) \lor S$
- $= \neg P \lor \neg(\neg Q \lor R) \lor S$
- $= \neg P \lor (\neg\neg Q \land \neg R) \lor S$
- $= \neg P \lor (Q \land \neg R) \lor S$
- $= \neg P \lor S \lor (Q \land \neg R)$
- $=(\neg P \lor S \lor Q) \land (\neg P \lor S \lor \neg R)$

5.Arithmetic assertions can be written in first-order logic with the predicate symbol <,the function symbols + and ×, and the constant symbols 0 and 1. Additional predicates can also be defined with biconditionals. (Chapter 8.20)

1. Represent the property “x is an even number.”
2. Represent the property “x is prime.”
3. Goldbach’s conjecture is the conjecture (unproven as yet) that every even number is equal to the sum of two primes. Represent this conjecture as a logical sentence.

your answer here...

---

1. $\forall x \; Even(x) \Leftrightarrow \exists y \; x=y+y$.
2. $\forall x \; Prime(x) \Leftrightarrow \forall y, z \; x=y \times z \Rightarrow y=1 \lor z=1$.
3. $\forall x \; Even(x) \Rightarrow \exists y, z \; Prime(y) \land Prime(z) \land x=y+z$.

Programming Excercises

1. Take the multiagent folder from assignment 2 and copy into a new directory for this practical. We will implement a version of minimax for Ghost Agents to make very smart ghosts.

First look at the file ghostAgents.py. Try to play classic pacman against the directional ghost and the random ghost. Use the following option:

-g TYPE, --ghosts=TYPE the ghost agent TYPE in the ghostAgents module to use [Default: RandomGhost]

Now implement a new ghost agent called MinimaxGhost. You will need to create a new class and methods in the file ghostAgent (you can ask the tutor if you need help to create the method stub).

This will involve implementing the following methods:

`class MinmaxGhost ( GhostAgent ):`

```
def __init__( self, index, ...):
    ...
```

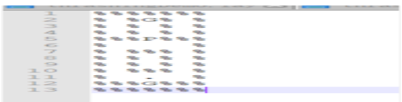
```
def getAction(self, state):
    ...

def getDistribution(self, state):
    ...
```

To test whether or not you have implemented correctly you will need to compare the behaviour of your new ghost agent with the random ghost agent. This has to be done over multiple tests (e.g. 10 runs).

- You can turn off the graphical display to use options -t or -q
- You can perform multiple runs (e.g. 10) by setting the -numGames=10
- You can fix the random seed as well.

Try to use different layouts to test. A simple layout might be good for testing initially as you are developing your algorithm. New layout files can be created in the layout directory (must be saved with end of file name ".lay"). For example: smallClassic.lay `%%%%%%%%%%%%%%%%%%%% %......%G G%......% %.%%...%% %%...%%.% %.%o.%........%.o%.% %.%%.%.%%%%%%.%.%%.% %........P.........% %%%%%%%%%%%%%%%%%%%%` newLayout.lay: `%%%%%%% % %G% % % % % % % % % % %%%P%%% % % % %%% % % % % % % % % % % %%% % % . % %%%G%%% %%%%%%%`

Please try to make some different layouts such as in this example,(the ghosts are marked by G, walls by % and pacman starting position by P,"." for pellet)



2. In this question we will test the new ghostAgent with different pacman agents. In your assignment you were asked to complete a Minimax pacman a version of Expectimax for pacman was provided. If you have not completed the assignment just use the provided expectimax version.

We will perform some experiments to compare the performance of pacman against different types of ghost agent:

- a random (not smart) ghost vs a pacman that assumes optimal play from ghosts (ie minimax pacman).
- a smart (minmax) ghost vs a pacman that assumes optimal play from ghosts
- random ghosts vs pacman that assumes ghosts may not always do an optimal move
- smart (minimax) ghosts vs pacman that assumes that ghosts may do suboptimal moves.

This will result in a table similar to the following:

	Adversarial Ghost	Random Ghost
Minimax Pacman	Won 5/5 Avg. Score: 483	Won 5/5 Avg. Score: 493
Expectimax Pacman	Won 1/5 Avg. Score: -303	Won 5/5 Avg. Score: 503

	Adversarial Ghost	Random Ghost
Minimax Pacman	Won 2/10 Avg.Score: -60.2	Won 4/10 Avg.Score: 292.2
Expectimax Pacman	Won 3/10 Avg.Score: 101.6	Won 7/10 Avg.Score: 706.5

3. Describe the performance (in terms of the distribution) of Pacman in each case.

In which cases is the Pacman agent implementing the correct assumption of the ghosts behaviour?

```
(py2) C:\Users\WEIChuan\Documents\GitHub\AI-for-Non-Computing\weeks7-8\prac4_Pacman>python pacman.py -p MinimaxAgent -l
smallClassic -g MinimaxGhost -q -n 10
Pacman died! Score: -355
Pacman died! Score: -381
Pacman emerges victorious! Score: 1141
Pacman died! Score: -359
Pacman died! Score: -371
Pacman died! Score: -124
Pacman died! Score: -64
Pacman emerges victorious! Score: 664
Pacman died! Score: -229
Pacman died! Score: -524
Average Score: -60.2
Scores:        -355.0, -381.0, 1141.0, -359.0, -371.0, -124.0, -64.0, 664.0, -229.0, -524.0
Win Rate:      2/10 (0.20)
Record:        Loss, Loss, Win, Loss, Loss, Loss, Loss, Win, Loss, Loss

(py2) C:\Users\WEIChuan\Documents\GitHub\AI-for-Non-Computing\weeks7-8\prac4_Pacman>python pacman.py -p MinimaxAgent -l
smallClassic -g RandomGhost -q -n 10
Pacman died! Score: -285
Pacman emerges victorious! Score: 1364
Pacman died! Score: 261
Pacman emerges victorious! Score: 1131
Pacman died! Score: -249
Pacman died! Score: -140
Pacman died! Score: -425
Pacman died! Score: -394
Pacman emerges victorious! Score: 684
Pacman emerges victorious! Score: 975
Average Score: 292.2
Scores:        -285.0, 1364.0, 261.0, 1131.0, -249.0, -140.0, -425.0, -394.0, 684.0, 975.0
Win Rate:      4/10 (0.40)
Record:        Loss, Win, Loss, Win, Loss, Loss, Loss, Loss, Win, Win
```

```
(py2) C:\Users\WEIChuan\Documents\GitHub\AI-for-Non-Computing\weeks7-8\prac4_Pacman>python pacman.py -p ExpectimaxAgent
-l smallClassic -g MinimaxGhost -q -n 10
Pacman emerges victorious! Score: 867
Pacman died! Score: -190
Pacman died! Score: -135
Pacman died! Score: -395
Pacman died! Score: -266
Pacman emerges victorious! Score: 821
Pacman died! Score: -349
Pacman emerges victorious! Score: 718
Pacman died! Score: 120
Pacman died! Score: -175
Average Score: 101.6
Scores:        867.0, -190.0, -135.0, -395.0, -266.0, 821.0, -349.0, 718.0, 120.0, -175.0
Win Rate:      3/10 (0.30)
Record:        Win, Loss, Loss, Loss, Loss, Win, Loss, Win, Loss, Loss

(py2) C:\Users\WEIChuan\Documents\GitHub\AI-for-Non-Computing\weeks7-8\prac4_Pacman>python pacman.py -p ExpectimaxAgent
-l smallClassic -g RandomGhost -q -n 10
Pacman emerges victorious! Score: 1117
Pacman emerges victorious! Score: 905
Pacman emerges victorious! Score: 1607
Pacman emerges victorious! Score: 1155
Pacman emerges victorious! Score: 692
Pacman died! Score: -355
Pacman emerges victorious! Score: 972
Pacman died! Score: 90
Pacman died! Score: -188
Pacman emerges victorious! Score: 1070
Average Score: 706.5
Scores:        1117.0, 905.0, 1607.0, 1155.0, 692.0, -355.0, 972.0, 90.0, -188.0, 1070.0
Win Rate:      7/10 (0.70)
Record:        Win, Win, Win, Win, Win, Loss, Win, Loss, Loss, Win
```

4. Describe why the ghosts seem as if they are cooperating when using minimax even though they are not sharing information with each other.

- Because their actions are all based on the Pacman's actions, and each ghost can impact the Pacman.