CS267A: Homework #4

Tianyu Zhang

May 6, 2020

Acknowledgements: If you have discussed with other students in the class regarding this homework, please acknowledge their names. See the syllabus for detailed policies about collaboration and academic honesty.

Problem 1

1. **Solution:** True

   First let's define $\alpha = A$ and $\beta = B$

   According to the Inclusion and exclusion principle: $A \vee B = A + B - A \wedge B$

   Then we have $A \wedge B = A + B - A \vee B$ and $A \vee B = B + (A - B)$

   Then we have $A \wedge B = A - (A - B)$ The Model count is used to find the number of model that satisfy the formula to be true. So we have $MC(A \wedge B) = MC(A) - MC(A - B)$

   Since the MC(A-B) can not be negative (≥ 0), then $MC(A) \geq MC(A \wedge B)$

2. **Solution:** True

   $S(\alpha\beta) = S(\alpha) + S(\beta) - S(\alpha \wedge \beta)$

   $(\alpha \implies \beta) \implies S(\alpha) \subseteq S(\beta)$

   $S(\alpha) = S(\alpha \wedge \beta)$

   $S(\alpha \vee \beta) = S(\beta))$ means $MC(\alpha \vee \beta) = MC(\beta)$

   in that case $(\alpha \vee \beta) \implies \gamma$

   similarly we can conclude that $MC(\alpha \vee \beta \vee \gamma) = MC(\gamma)$

3. **Solution:** True

   $(\alpha \implies \beta) \implies S(\alpha) \subseteq S(\beta)$

   $(\alpha \implies \gamma) \implies S(\alpha) \subseteq S(\gamma)$

   First we assume that $S(\alpha) \not\subset S(\beta \vee \gamma)$

   Then there is an x from $S(\alpha)$ but not from $S(\beta \wedge \gamma)$

   which means $x \notin S(\beta) \vee x \notin S(\gamma)$

   $\implies S(\alpha) \not\subset S(\beta) \vee S(\alpha) \not\subset S(\gamma)$ Contradict the given rules.

   So $S(\alpha) \subseteq S(\beta \vee \gamma) \implies MC(\alpha) \leq MC(\beta \vee \gamma)$

Problem 2

1.

Solution:

propositional grounding: Since the Friends(x,y) is one directional, then Friends(x,y) and Friends(y,x) are four different groundings.

$Alice.Bob.(Smokes(Alice) \wedge Friends(Alice, Bob)) \implies Smokes(Bob)$

$Bob.Alice.(Smokes(Bob) \wedge Friends(Bob, Alice)) \implies Smokes(Alice)$

$Alice.Alice.(Smokes(Alice) \wedge Friends(Alice, Alice)) \implies Smokes(Alice)$

$Bob.Bob.(Smokes(Bob) \wedge Friends(Bob, Bob)) \implies Smokes(Bob)$

2.

Solution:

There are two groundings and for first one:

1. Smokes(Alice) = True, Friends(Alice,Bob) =True, Smokes(Bob) = True
2. Smokes(Alice) = False, Friends(Alice,Bob) = True, Smokes(Bob) = False
3. Smokes(Alice) = True, Friends(Alice,Bob) = False, Smokes(Bob) = False
4. Smokes(Alice) = True, Friends(Alice,Bob) = False, Smokes(Bob) = True
5. Smokes(Alice) = False, Friends(Alice,Bob) = True, Smokes(Bob) = True
6. Smokes(Alice) = False, Friends(Alice,Bob) = False, Smokes(Bob) = False
7. Smokes(Alice) = False, Friends(Alice,Bob) = False, Smokes(Bob) = True

There are 7 models in total.

For second one, the model count are same since only the names are swapped.

For third and fourth one, there are four models in total.

So there are 18 models in total.

Problem 3

1.

Solution:

1. $Occupation(Emily, Surgeon) \vee Occupation(Emily, actor)$
2. $Occupation(Joe, Actor) \wedge \exists x.[(Occupation(Joe, x) \wedge \neg(x = actor))]$
3. $\forall x[(Occupation(x, Surgeon)) \implies (Occupation(x, doctor))]$
4. $\exists x.[Boss(x, Emily) \wedge Occupation(x, Lawyer)]$
5. $\exists x.\forall y Occupation(x, Lawyer) \wedge [Customer(y, x) \implies Occupation(y, Doctor)]$
6. $\forall x Occupation(x, Surgeon) \implies \exists y.[Occupation(y, Lawyer) \wedge Customer(x, y)]$

2.

Solution:

1. Every Doctor has a customer.
2. There is a doctor that he/she is everyone's customer.
3. There is a lawyer that he/she is every doctor's customer.

Problem 4

1. **Solution:** The idea behind my code is I write down the all possible permutations with their probability and sum up all the permutation which satisfies the rule. 2 consecutive heads in 5 flips 19*0.03125 = 0.59375

2. Please find the following my code that generates the solution.

```
0.5::msw(coin(1),ID,head); 0.5::msw(coin(1),ID,tail).

dcoin(N,Rs) :-
   dcoin(N,coin(1),Rs).

dcoin(N,Coin,[R|Rs]) :-
  N > 0,
  msw(Coin,N,R),
  NextCoin = coin(1),
  N1 is N-1,
  dcoin(N1,NextCoin,Rs).
dcoin(0,_,[]).


query(dcoin(5,_)).
```

3. **Solution:** The idea behind my code is I write down the all possible permutations with their probability and sum up all the permutation which satisfies the rule. 5 consecutive heads in 6 flips 3 * 0.015625 = 0.046875

4. Please find the following my code that generates the solution.

```
0.5::msw(coin(1),ID,head); 0.5::msw(coin(1),ID,tail).

dcoin(N,Rs) :-
   dcoin(N,coin(1),Rs).

dcoin(N,Coin,[R|Rs]) :-
  N > 0,
  msw(Coin,N,R),
  NextCoin = coin(1),
  N1 is N-1,
  dcoin(N1,NextCoin,Rs).
dcoin(0,_,[]).


query(dcoin(6,_)).
```

5. **Solution:** The idea behind my code is I write down the all possible permutations with their probability and sum up all the permutation which satisfies the rule. 7 consecutive heads in 10 flips 20 * 0.0009765625 = 0.01953125

6. Please find the following my code that generates the solution.

```
0.5::msw(coin(1),ID,head); 0.5::msw(coin(1),ID,tail).

dcoin(N,Rs) :-
   dcoin(N,coin(1),Rs).

dcoin(N,Coin,[R|Rs]) :-
  N > 0,
  msw(Coin,N,R),
   NextCoin = coin(1),
  N1 is N-1,
   dcoin(N1,NextCoin,Rs).
dcoin(0,_,[]).


query(dcoin(10,_)).
```

Problem 5

1. **Solution:**

   At least 3 midterms in 2 classes : 0.35799.

   at least 7 midterms in 4 classes : 0.0072588

2. Please find the following my code that generates the solution.

```
0.25::midterm(1,0); 0.7::midterm(1,1); 0.05::midterm(1,2).
0.1::midterm(2,0); 0.1::midterm(2,1); 0.8::midterm(2,2).
0.01::midterm(3,0); 0.1::midterm(3,1); 0.89::midterm(3,2).
0.49::midterm(4,0); 0.5::midterm(4,1); 0.01::midterm(4,2).
0.7::midterm(5,0); 0.3::midterm(5,1); 0.0::midterm(5,2).

sample(_,0,[]).
sample([X|L], N, [X|S]):-
     N > 0, sample_now([X|L],N),
     N2 is N-1, sample(L,N2,S).
sample([H|L], N, S) :-
     N > 0, \+ sample_now([H|L],N),
     sample(L,N,S).

P::sample_now(L,N) :- length(L, M), M >= N, P is N/M.

sum(_,0).
sum([X|L],S):-
     midterm(X,A),
     sum(L,S1),
     S is S1+A.

ans(N,M) :- sample([1,2,3,4,5], N, X), sum(X, S), S >= M.
```

```
query(ans(2,3)).
query(ans(4,7)).
```