CS267A: Homework #4

Peter Racioppo (103953689)

May 7, 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.
I collaborated with Tianyu Zhang on Qs. 4.

Problem 1

Topics: Logic
Let α, β, γ be Boolean formulae, and let MC denote the model count of a Boolean formula. Select whether the following is a true or false statement about model counts and provide a brief justification for your choice.
1. $MC(\alpha) \leq MC(\alpha \wedge \beta)$.
2. Suppose $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \gamma$. Then, $MC(\alpha \vee \beta \vee \gamma) = MC(\gamma)$.
3. Suppose $\alpha \Rightarrow \beta$ and $\alpha \Rightarrow \gamma$. Then, $MC(\alpha) \leq MC(\gamma \wedge \beta)$.

Solution:
Let S denote the set in which a statement is true, and $MC = |S|$ denote the set's cardinality.
1. True
If both α and β are true, than α is true. Thus,
$S(\alpha \wedge \beta) \subseteq S(\alpha)$
$\Rightarrow MC(\alpha) \geq MC(\alpha \wedge \beta)$

2. True
$S(\alpha \vee \beta) = S(\alpha) + S(\beta) - S(\alpha \wedge \beta)$
$(\alpha \Rightarrow \beta) \Rightarrow S(\alpha) \subseteq S(\beta)$
$\Rightarrow S(\alpha) = S(\alpha \wedge \beta)$.
Thus, $S(\alpha \vee \beta) = S(\beta) \Rightarrow MC(\alpha \vee \beta) = MC(\beta)$.
Now, let $x = (\alpha \vee \beta)$. Then, $x \Rightarrow \gamma$, so $MC(\alpha \vee \beta \vee \gamma) = MC(x \vee \gamma) = MC(\gamma)$, by the previous argument.

3. True
$(\alpha \Rightarrow \beta) \Rightarrow S(\alpha) \subseteq S(\beta)$
$(\alpha \Rightarrow \gamma) \Rightarrow S(\alpha) \subseteq S(\gamma)$
Suppose, by way of contradiction, that $S(\alpha) \not\subset S(\beta \wedge \gamma)$
$\Rightarrow \exists\, x \in S(\alpha)$ s.t. $x \notin S(\beta \wedge \gamma)$
$\Rightarrow x \notin S(\beta) \vee x \notin S(\gamma)$
$\Rightarrow S(\alpha) \not\subset S(\beta) \vee S(\alpha) \not\subset S(\gamma)$.
This is a contradiction, so we have
$S(\alpha) \subseteq S(\beta \wedge \gamma) \Rightarrow MC(\alpha) \leq MC(\beta \wedge \gamma)$

Problem 2

Topics: First-order logic and grounding
Consider the following first-order vocabulary:
• *Friends*(x, y) *is a predicate which says* x *is friends with* y *(one-directional)*
• *Smokes*(x) *is a predicate which says* x *is a smoker.*
Then, from this vocabulary we can build the following first-order sentence:
$\forall x, \forall y, (Smokes(x) \wedge Friends(x, y)) \Rightarrow Smokes(y)$.
Answer the following for the above sentence:
1. Assume that there is a finite domain of people $\{Alice, Bob\}$ *that each x and y variable is drawn from. Compute the propositional grounding* Δ *for the first-order sentence with this finite domain.*
2. How many models of Δ *are there, still assuming there is a finite domain of people* $\{Alice, Bob\}$*?*

Solution:
1. The four possibilities are:
$(Smokes(Alice) \wedge Friends(Alice, Bob)) \Rightarrow Smokes(Bob)$
$(Smokes(Bob) \wedge Friends(Bob, Alice)) \Rightarrow Smokes(Alice)$
$(Smokes(Alice) \wedge Friends(Alice, Alice)) \Rightarrow Smokes(Alice)$
$(Smokes(Bob) \wedge Friends(Bob, Bob)) \Rightarrow Smokes(Bob)$
2. Since there are 2 elements in the domain (Alice and Bob), there are 2 possibilities for x and 2 for y. Thus, there are 4 possible models. In general, if the finite domain of people has n elements, there are n^2 models.

Problem 3

Topics: Modeling with first-order logic
Consider a vocabulary with the following symbols:
• *Occupation(p, o): A predicate which states person p has occupation o.*
• *Customer(p1, p2): A predicate which states p1 is a customer of p2.*
• *Boss(p1, p2): A predicate which states p1 the boss of p2.*
• *Doctor,Person,Lawyer,Actor,Surgeon: constants which denote occupations.*
• *Emily, Joe: constants denoting people.*

Using the above symbols, translate the following sentences into first-order logic:
1. Emily is either a surgeon or an actor.
2. Joe is an actor, but he also has at least one other job.
3. All surgeons are doctors.
4. Emily has a boss who is a lawyer.
5. There exists a lawyer whose customers are all doctors.
6. Every surgeon has a lawyer.

Problem 3 continued on next page. . .

Solution:
1. Occupation(Emily, Surgeon) $\vee$ Occupation(Emily, Actor).
2. Occupation(Joe, Actor) $\wedge \exists\, x \in$ {Doctor,Person,Lawyer,Actor,Surgeon} s.t. $x \neq$ Actor $\wedge$ Occupation(Joe,x).
3. $\forall x \in$ {Emily, Joe}, Occupation(x, Surgeon) $\Rightarrow$ Occupation(x, Doctor).
4. $\exists\, x \in$ {Emily, Joe} s.t. Boss(x, Emily) $\wedge$ Occupation(x,Lawyer).
5. $\exists\, x \in$ {Emily, Joe} s.t. Occupation(x,Lawyer)
$\wedge$ [$\forall\, y \in$ {Emily, Joe}, Customer(y, x) $\Rightarrow$ Occupation(y,Doctor)]
6. $\forall x \in$ {Emily, Joe}, Occupation(x, Surgeon) $\Rightarrow$ [$\exists\, y$ s.t. Occupation(y,Lawyer) $\wedge$ Customer(x,y)]

Translate the following first-order sentences into English:
i. $\forall x$.Occupation(x,Doctor) $\Rightarrow \exists y$. Customer(x,y)
ii. $\exists x$. Occupation(x,Doctor) $\Rightarrow \forall y$. Customer(x,y)
iii. $\exists x$. $\forall y$. Occupation(x,Lawyer) $\wedge$ Customer(x,y) $\wedge$ Occupation(y,Doctor)

Solution:
i. Every doctor is someone's customer.
ii. There is a person who, if they are a doctor, is everyone's customer.
iii. There is a lawyer who is a customer of every doctor.

Note: I think you may have mistaken the ordering of p1, p2 in your Customer(.,.) symbol.

Problem 4

Flipping coins in ProbLog
Suppose you have an unbiased coin (50% of the time it will show heads) which you flip M times in sequence. What is the probability that N consecutive heads appear at some in this sequence? Implement this as a function in ProbLog. Submit your code, as well as answers to the following queries:

1. 2 consecutive heads in 5 flips
2. 5 consecutive heads in 6 flips
3. 7 consecutive heads in 10 flips

Solution:
1. 0.59375
2. 0.046875
3. 0.01953125

I begin by listing out each possibility:

```
0.5::heads(X).
two_h :- heads(x1), heads(x2).
two_h :- heads(x2), heads(x3).
two_h :- heads(x3), heads(x4).
two_h :- heads(x4), heads(x5).
```

```
query(two_h).

% ---------------------------------

0.5::heads(X).
five_h :- heads(x1), heads(x2), heads(x3), heads(x4), heads(x5).
five_h :- heads(x2), heads(x3), heads(x4), heads(x5), heads(x6).

query(five_h).

% ---------------------------------

0.5::heads(X).
seven_h :- heads(x1), heads(x2), heads(x3), heads(x4), heads(x5), heads(x6),
    heads(x7).
seven_h :- heads(x2), heads(x3), heads(x4), heads(x5), heads(x6), heads(x7),
    heads(x8).
seven_h :- heads(x3), heads(x4), heads(x5), heads(x6), heads(x7), heads(x8),
    heads(x9).
seven_h :- heads(x4), heads(x5), heads(x6), heads(x7), heads(x8), heads(x9),
    heads(x10).

query(seven_h).

% ---------------------------------
% Attempt at a function:

0.5::heads(C,ID).

% This predicate calculates the probability of N consecutive heads.
fact1(0,Result,n) :-
     Result is 1.
fact1(N,Result,n) :-
    N > 0,
    N1 is N-1,
     Result1 is Result, heads(n,N1),
     fact1(N1,Result1,n).

query(fact1(4,1,_)).

% This is an attempt to add the probabilities of rolling N consecutive
% heads (M-N) times.
% A or B = !(!A and !B)
fact2(M,N,M-N-1,Result,n) :-
     Result is 1.
fact2(M,N,i,Result,n) :-
     i > M-N-1,
     i1 is i-1,
     \+ Result1 is (\+ Result, \+ fact1(M,1,n)),
     fact2(M,N,i1,Result1,n).

query(fact2(4,2,4,1,_)).
```

Problem 4 continued on next page...

```
% ------------------------------
% Second attempt at a function:

0.5::flip(coin(1),h,ID); 0.5::flip(coin(1),t,ID).

% This predicate computes all permutations
% of heads and tails of length N. There are
% 2^N such permutations.
permute(N,Coin,[R|Rs]) :-
  N > 0,
  N1 is N-1,
   flip(Coin,R,N),
   permute(N1,Coin,Rs).
permute(0,_,[]).

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

query(permute(5,_)).

% ------------------------------
```

It remains to count up the probabilities of the permutations in which N heads appear in sequence. I wasn't able to finish this.

Problem 5

Topics: Modeling with ProbLog
Suppose you sample N classes from the following list of classes:
- *CS267A: 0, 1, and 2 midterms with probability 0.25, 0.7, and 0.05, respectively.*
- *CS31: 0, 1, and 2 midterms with probability 0.1, 0.1, and 0.8, respectively.*
- *Math61: 0, 1, and 2 midterms with probability 0.01, 0.1, and 0.89, respectively.*
- *History101: 0, 1, and 2 midterms with probability 0.49, 0.5, and 0.01, respectively.*
- *English101: 0, 1, and 2 midterms with probability 0.7, 0.3, and 0, respectively.*

What is the probability that there are at least M midterms in a sampled set of size N? Implement this as a function in ProbLog. Submit your code as well as answers to the following queries:

1. at least 3 midterms in 2 classes
2. at least 7 midterms in 4 classes

Solution:
1. The probability of 3 midterms is 0.238 and the prob. of 4 is 0.1225. The prob. of at least is therefore $0.238 + 0.1225 = 0.3605$.
2. The probabilities of 7 and 8 midterms are, respectively, 0.05831 and 0.01500625, so the probability of at least 7 is $0.05831 + 0.01500625 = 0.07331625$.

Note: My solutions are not functions. Furthermore, I assume sampling with replacement, whereas it should be without replacement.
Part 1:

```
0.25::flip(class(1),0,ID); 0.70::flip(class(1),1,ID); 0.05::flip(class(1),2,ID
    ).
0.10::flip(class(2),0,ID); 0.10::flip(class(2),1,ID); 0.80::flip(class(2),2,ID
    ).
0.01::flip(class(3),0,ID); 0.10::flip(class(3),1,ID); 0.89::flip(class(3),2,ID
    ).
0.49::flip(class(4),0,ID); 0.50::flip(class(4),1,ID); 0.01::flip(class(4),2,ID
    ).
0.70::flip(class(5),0,ID); 0.30::flip(class(5),1,ID); 0.00::flip(class(5),2,ID
    ).

% Sample a class, with uniform distribution.
0.20::sample(s(1),1,ID); 0.20::sample(s(1),2,ID); 0.20::sample(s(1),3,ID);
0.20::sample(s(1),4,ID); 0.20::sample(s(1),5,ID).

flip(class(1),1).
flip(class(2),1).
flip(class(3),1).
flip(class(4),1).
flip(class(5),1).

sample(s(1),1).
sample(s(1),2).

sum(S) :-
    sample(s(1),X,1),
    sample(s(1),Y,2),
    flip(class(X),A,1),
    flip(class(Y),B,2),
    S is A+B,
    S > 2.

query(sum(_)).
```

Part 2:

```
0.25::flip(class(1),0,ID); 0.70::flip(class(1),1,ID); 0.05::flip(class(1),2,ID
    ).
0.10::flip(class(2),0,ID); 0.10::flip(class(2),1,ID); 0.80::flip(class(2),2,ID
    ).
0.01::flip(class(3),0,ID); 0.10::flip(class(3),1,ID); 0.89::flip(class(3),2,ID
    ).
0.49::flip(class(4),0,ID); 0.50::flip(class(4),1,ID); 0.01::flip(class(4),2,ID
    ).
0.70::flip(class(5),0,ID); 0.30::flip(class(5),1,ID); 0.00::flip(class(5),2,ID
    ).

% Sample a class, with uniform distribution.
0.20::sample(s(1),1,ID); 0.20::sample(s(1),2,ID); 0.20::sample(s(1),3,ID);
0.20::sample(s(1),4,ID); 0.20::sample(s(1),5,ID).

flip(class(1),1).
```

```
flip(class(2),1).
flip(class(3),1).
flip(class(4),1).
flip(class(5),1).

sample(s(1),1).
sample(s(1),2).
sample(s(1),3).
sample(s(1),4).

sum(S) :-
    sample(s(1),X,1),
    sample(s(1),Y,2),
    sample(s(1),Z,3),
    sample(s(1),W,4),
    flip(class(X),A,1),
    flip(class(Y),B,2),
    flip(class(Z),C,3),
    flip(class(W),D,4),
    S is A+B+C+D,
    S > 6.

query(sum(_)).
```