15-381 Spring 2007
Assignment 4: Auctions, Probability, and Uncertainty
SOLUTIONS

Questions to Gil (egjones+@cs.cmu.edu)

Spring 2007
Out: March 20
Due: April 3, 1:30pm Tuesday

The written portion of this assignment must be turned in at the beginning of class at 1:30pm on February 20th. Type or write neatly; illegible submissions will not receive credit. Write your name and andrew id clearly at the top of your assignment. If you do not write your andrew id on your assignment, you will lose 5 points.

The code portion of this assignment must be submitted electronically by 1:30pm on April 3rd. To submit your code, please copy all of the necessary files to the following directory:

/afs/andrew.cmu.edu/course/15/381/hw4_submit_directory/yourandrewid

replacing yourandrewid with your Andrew ID.

Late Policy. Both your written work and code are due at 1:30pm on 4/3. Submitting your work late will affect its score as follows:

- If you submit it after 1:30pm on 4/3 but before 1:30pm on 4/4, it will receive 90% of its score.
- If you submit it after 1:30pm on 4/4 but before 1:30pm on 4/5, it will receive 50% of its score.
- If you submit it after 1:30pm on 4/5, it will receive no score.

Collaboration Policy.

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas in the class in order to help each other answer homework questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers
- not to copy answers
- not to allow your answers to be copied

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we ask that you specifically record on the assignment the names of the people you were in discussion with (or "none" if you did not talk with anyone else). This is worth five points: for each problem, you solution should either contained the names of people you talked to about it, or "none." If you do not give references for each problem, you will lose five points. This will help resolve the situation where a mistake in general discussion led to a replicated weird error among multiple solutions. This policy has been established in order to be fair to everyone in the class. We have a grading policy of watching for cheating and we will follow up if it is detected.

Problem 1 (Auctioning a Wii)

Much of this assignment will examine various ways that you go about acquiring a scarce commodity - a Nintendo Wii. In this question you are asked to think about auctions in ways slightly different than what was covered in class - considering things like truthfulness, revenue generation, and susceptibility to collusion. Most of the things you've learned about game theory are true when evaluating auctions, including concepts like Nash equilibria.

Question 1.1

Your first attempt to acquire a Wii involves on-campus auctions held by student entrepreneurs. One auctioneer is holding an English auction: he sets a low price and instructs the participants to stand until they are no longer willing to pay the price. He then slowly raises the price until only one person remains standing, and that person pays the amount at which the last person sat down. The other auctioneer auctions her Wii in a Dutch auction. She sets a high price and instructs the bidders to stand when they are willing to pay that price for the Wii. She then slowly lowers the price. The first person to stand gets the Wii at that price. In the following problems assume that your valuation for the Wii is v.

A dominant strategy for a bidder in an auction is very similar to the dominant strategy concept from game theory: does an agent maximize its payoff (the agent's value for an item less the amount paid for that item) by following a particular strategy no matter what every other agents' strategy is?

1. (3 points) Is it a dominant strategy for the English auction to represent your value truthfully - to stand up until the price of v is reached, and then to sit down? Why or why not?
2. (3 points) Is it a dominant strategy for the Dutch auction to represent your value truthfully - to remain seated until the price of v is reached and then to immediately stand up? Why or why not?
3. (3 points) Suppose that there are people with a set of valuations $v_1, v_2, \ldots, v_n$ where $v_1 \geq v_2 \geq \cdots \geq v_n$ for the Wiis in each auction, and that the participants in the auction intend to represent their valuations truthfully (as described above). If these people participate in each of the two auctions independently, will one Wii auctioneer make more money than the other?
4. (6 points) Suppose you know that your valuation v is the highest among all auction participants in each of the auctions, that your friend's valuation v_2 is the next highest, and that the third highest valuation, v_3, is strictly less than either of yours. Furthermore, you know that all bidders with valuations v_3 and below are intending to represent their valuations truthfully. For each of the auctions describe an arrangement you can make with your friend that will increase your payoffs over truthfully representing your bids - for your arrangements show how each of you and the friend will make more by the arrangement than either would make by truthfully revealing your valuations. Also indicate for the given auction whether under what circumstances, if any, given that you both agreed to cheat that there would be an inducement to not perform the agreed upon action - essentially, for each of the auctions does double-crossing pay?

Solution 1.1

1. The English auction is equivalent to the second price auction, and it is a dominant strategy to represent your value truthfully. If you remain standing past your value v and there is someone else standing then even if you win the auction you will pay more than v for the item, which is a bad outcome. If you sit down before you reach your value v then you might not win an auction that you would have won and paid a value less than your value v, thus making zero profit instead of a positive profit.

2. The Dutch auction is equivalent to the first price auction, and it is not a dominant strategy to represent your value truthfully. Suppose you do represent your value truthfully - standing up when the value reaches v. In one case, someone stands up before you, and you don't win the auction - but you wouldn't have wanted the item a price greater than v, so being truthful gained you zero profit instead of a negative profit. In the other case, you stand up at exactly v, and win the item for v, but this also nets you zero profit. If you had remained seated until a number slightly lower than v and then stood up you could have received a positive profit - and if someone else stands up before you can, then you make zero profit anyway.

3. In every case except one the auctioneer in the Dutch auction makes more money than the auctioneer in the English auction. The Dutch auctioneer gets paid v_1, and the English auctioneer gets paid v_2. If $v_1 = v_2$ then the auctioneers get paid the same amount.

4. (If you assume you know v_3) In either case if you win the auction your friend will gain zero profit, so any amount of money you pay him should induce him to cheat. In the English auction an arrangement you can make with your friend is to have him sit down at the value v_3. You would have paid v_2 for the item, and now pay only v_3. You should keep $v_3 - v_2 - \epsilon$ of the extra profit for yourself, and give your friend ϵ for some low ϵ. In the Dutch auction the agreement you should make is to not stand until $v_3 + \epsilon$, and to have your friend remained sitting entirely. This generates an extra profit of $v_2 - v_3$ profit for you, as you would have stood before at $v_2 + \epsilon$. You can keep $v_2 - v_3 - \epsilon$ for yourself and pay ϵ to your friend. In the English auction neither of you has inducement to double cross - if your friend violated your agreement and remained standing passes v_3 you would realize it and remain standing until v_2 - if he remained standing at that point he would make negative profit instead of ϵ. There is inducement for your friend to double-cross in the Dutch auction. If he stands up at $v_3 + 2 * \epsilon$, for instance, then he makes profit of $v_2 - v_3 - 2 * \epsilon$, which may be greater than ϵ. This inducement to double-cross means that the Dutch auction is less susceptible to this form of bidder collusion than the English auction. If you don't assume you know v_3 in the English it doesn't matter, as the person with v_3 will sit down at v_3 in any case. In the Dutch auction you can get your friend to tell you his valuation, and then can stand at v_2, paying him some small ϵ to induce him not to stand at v_2 and get the Wii for zero profit. It was acceptable in this case to say there was no inducement for him to cheat. But it was better to point out that the friend could lie about his valuation, claiming it was lower than it was, and could then stand at some value $v_2 - \beta$, where $\beta > \epsilon$. In this case you both might lose to the v_3 player, but he has a chance of making more money.

Question 1.2 (10 points)

Your second attempt to acquire a Wii comes in the form of a different kind of auction. One of your friends has tired of his Wii and has offered to auction it to you and one other person by the following method - each of you submits a sealed bid to him, the auctioneer. You each value the object at v, and if your bids are the same you get dual-custody of the Wii, which is worth $\frac{v}{2}$ to each of you. The higher bid wins the Wii, and both of you pay the amount of your bids to him no matter which one of you won. Is there a pure strategy Nash equilibrium in this game? If so, describe it. If not, tell why not.

Solution 1.2

There is no pure equilibrium by the following argument:

There's no pair of bids (x, x) with $x < v$ that constitute a Nash equilibrium because either of you can increase your payoff considerably (winning the whole Wii instead of dual custody) by increasing your bid slightly.

(v, v) is not an equilibrium as either of you can increase your payoff from $\frac{-v}{2}$ to 0 by reducing your bid to zero or increase your payoff to $-\epsilon$ by increasing your bid to $v + \epsilon$.

No pair of actions (x_1, x_2) with $x_1 \neq x_2$ is a Nash equilibrium because the one of you with the higher bid can increase your payoff by reducing your bid to be slightly above the other bid, and the player with the lower bid, if that bid is positive, can increase his payout by reducing his bid to zero or making his bid slightly higher than your bid. If $x_1 > v - \epsilon$, he would be better off bidding $x_1 + \epsilon$; otherwise he'd be better off reducing his bid to 0.

It is not valid to say that any equilibrium for a symmetric game must take the form of equal actions (in this case that the only possible equilibria are for $x_1 = x_2$). You can state that if there is an equilibrium (x_1, x_2) then that implies that there is also an equilibrium at (x_2, x_1), but not that $(x_1 = x_2)$.

Problem 2 - Waiting for a Wii

Imagine that you are no longer pursuing the auction route for acquiring a Wii but are instead going straight to the source - the Waterfront Target early on Sunday morning. You still have a valuation v for the Wii - representing your desire for the Wii monetarily. Once you get to Target you know that you can buy a single Wii for the advertised price c, which we assume is strictly less than v (you are always willing to pay more for the Wii than it actually costs). You also know that to get a Wii at that price you must be one of the first m people in line when the store opens at 8 AM, where m is the number of Wiis that Target has in stock. Thus it pays to arrive early, but you also recognize that you don't want to wait too long in the cold on Sunday morning. The desire to wait as little as possible is represented by an opportunity cost o for each minute before 8 AM that you have to wait in line to get the Wii - you could be spending that time sleeping, and depending on what your Saturday night looked like that opportunity cost can be pretty high. Finally, there's a cost k that represents getting to the store, which combines the opportunity cost of the time it takes to get to the Waterfront and the gas and munchies you had to bribe your friend with to drive you. Your goal is to minimize your waiting time and the associated cost while still obtaining a Wii.

Question 2.1 (10 points)

Suppose that you know that m is 20 for a given Sunday, and that there are 100 people that want a Wii. Suppose furthermore that all participants know the values of all other participants, with $v_1 > v_2 > \cdots > v_{100} > c$. Also suppose that $k = 0$ (no cost to get to the store) and that $o = 1$ for all participants. Finally, assume that if many people arrive in line simultaneously that the people with the highest valuation get the first spots in line. What is the Nash Equilibrium for this situation?

Notes: A Nash equilibrium for this case follows the same model we've learned about in game theory: given that all other participants are going to arrive at a particular time, when does a single participant want to arrive? At an equilibrium all participants will be satisfied with their strategy given the strategy of all other individuals. Note: you can state your solution and justify it informally.

Solution 2.1

The equilibrium is that all people from v_1 to v_{21} will simultaneously arrive at the line at $v_{21} - c$ minutes (as waiting for one minute is cost 1) before 8 AM. v_{21} will leave if he sees that he will lose the Wii in the value tie-breaker, and stay if he can get a Wii (making zero profit). All others will stay at home, or drive-by at any arbitrary point between their zero times and 8 AM. All the people with v_1 through v_{20} have positive expectation in this equilibrium, and given the strategy of all other participants none of them would prefer to arrive earlier in the line; by arriving later they will sacrifice their spots in line to the person with v_{21}. All other people would have zero or negative expectation from arriving at this time, and thus would prefer to stay at home or drive-by later - if they do drive-by, they should not do it before the time at which they would expect to make zero profit.

In question 2.1 you found an equilibrium for a simple version of the problem of when to arrive at Target to maximize your payoff. The simplification came from the fact that all valuations were common knowledge. This is not generally the case, the this question examines relaxing that constraint - when should you get to the store given that you don't know anyone's value or the strategy they are using?

This section involves coding. Download the files at

`http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15381-s07/www/hw4/hw4.tar`

These files simulate the line in front of Target on a series of Sunday mornings. You will be changing a single function in a C++, and must write your code in such a way that it will compile in this language. Please come and talk to me (Gil) if this is going to be a significant problem - all that should be required is C style coding - just filling out a single function.

For this problem assume that every Sunday morning each agent that wants a Wii is told both its valuation v for the Wii for that week - the price that it thinks it can get on Ebay for it, say - as well as that week's opportunity cost o. The agents are also told the amount in their bank accounts, and the sorted arrival times of all agents the previous week. Each agent must then use this information to determine if they will go to Target to wait for a Wii, and if so, how many minutes before 8 AM they will arrive in the line.

In this exercise you will write a function that takes your valuation for the week, your opportunity cost for the week, your bank account balance, and the arrival times of all other agents the previous week and returns the number of minutes t before 8 AM that you wish to get in line at the Target. If you choose to go to Target, get in line, and are one of the first m people in line for the week you will buy a Wii and your account will be credited with your profit P as determined by:

$$P = v - c - (t * o) - k$$

Otherwise you will have waited in line until 8 AM in vain and your account will receive a negative profit of

$$P = -(o * t) - k$$

If you choose to stay home your profit is 0.

Your task is to replace the function in `better_ArrivalTime.h` with your own code. See the `readme.txt` file included with the code for more information on the particulars of the coding assignment.

Question 2.2.1

First implement a strategy that does well against the not-so-smart random players in `random_ArrivalTime.h`. To implement this function you should only need to reason about your value and opportunity cost (and about your bank account balance a little bit as described in the `readme.txt` file).

Comment your code and turn in a brief description of what your function does with the write-up for this question (described below).

Question 2.2.2

For this problem you will implement a strategy to compute arrival times given that your peers will also be implementing strategies of their own. After you've turned in your code we will compile all of your functions together into the `wiiLine` program and run them for 5000 cycles. You must create a function that will maximize your profit given that your classmates will all trying to be doing the same. The vector of arrivalTimes (described in the `readme.txt` file) might be useful for this, as it gives information about the trends of arrival times of your peers.

There is no right answer for this problem - we just want you to think about strategies for a competitive real world situation where other agents have similar objectives to your own.

You will get credit for this question by implementing a function and describing your design decisions in a write-up to be submitted with the written portion of your homework.

You can submit different functions for 2.2.1 and 2.2.2 - if you want to do this turn in `better_ArrivalTime.h` with two versions of the function, with the version for 2.2.2 commented out.

Point values and what to turn in:

- (5 points) You turn in your commented code - in your directory please include all code necessary to compile `wiiLine` (almost all of which I've given you), and a compiled wiiLine executable which runs your code versus random participants (more info in the `readme.txt`).
- (5 points) Your code performs reasonably against a number of random opponents - we will run this in your submission directory.
- (5 points) Your code performs reasonably against other people's code - we will be extracting your function and running it against other class members'.
- (5 bonus points) Your code performs exceptionally against other peoples' - you finish in the top of the class after a large number of cycles (5000).
- (10 points) You write up and turn it in with the written of your homework (not a README) an analysis of how you made the design decisions you made in your arrival time for both questions 2.2.1 and 2.2.2.

Solution 2.2

Full points were awarded for 2.2.1 as long as your solution beat the random solution and you did a good job of explaining your solution.

For 2.2 I evaluated your reasonable performance against your peers as follows: if you gained money versus your baseline you got full credit. If you lost money versus your baseline but remained positive you lost either 2 or 3 points. If you went negative you lost either 4 or 5 points for reasonable performance.

Your explanation/justification of your solution were evaluated independently from your performance - you could get full credit even if you lost money.

Bonus points were awarded as follows: 1-2 place got +5 points, $3-6$ place got +3, and 7-15 place got +1.

Problem 3 - Probability (8 points)

Your Doritos are gone, and you have two apartment mates as suspects - Marc and Steve. You know the following things:

1. In previous chip-swiping incidents Marc was implicated 85% of the time, and Steve only 15% of the time.
2. Your across-the-hall neighbor believes she saw Marc eating your Doritos last night, but her eyesight is notoriously poor. You estimate that her rate of Marc/Steve differentiation is only 80% (i.e. 80% of the time she thinks she sees Marc it's actually Marc, and 20% of the time it's actually Steve. And if she thinks she sees Steve she's correct 80% of the time).

Question 3.1 (4 points)

Given the above data what is the probability that Marc purloined the Doritos? Show your work.

Solution 3.1

WM means the witness reported Marc, DM means doritoes were taken by Mac, DS means the Doritoes were taken by Steve. We are trying to determine $p(DM|WM)$:

$$
\begin{aligned}
p(DM|WM) &= \frac{p(WM|DM)p(DM)}{p(WM)} && \text{by Bayes' rules} \\
&= \frac{.85 * .8}{p(WM, DM) + p(WM, DS)} \\
&= \frac{.68}{p(DM|WM)p(DM) + p(DS|WM)p(DS)} \\
&= \frac{.68}{.85 * .8 + .2 * .15} \\
&= \frac{.68}{.71} \\
&= .96
\end{aligned}
$$

So there's a 96% chance that Marc took the Doritos.

Question 3.2 (4 points)

What would the probability be that Marc stole the chips if your neighbor thought she saw Steve? Show your work.

Solution 3.2

$$
\begin{aligned}
p(DM|WS) &= \frac{p(WS|DM)p(DM)}{p(WS)} \\
&= \frac{.2 * .85}{p(DM|WS)p(DM) + p(DS|WS)p(DS)} \\
&= \frac{.2 * .85}{.2 * .85 + .8 * .15} \\
&= \frac{.17}{.17 + .12} \\
&= .58
\end{aligned}
$$

So even though our neighbor saw Steve, there's still a greater chance that Marc took the doritoes.

Problem 4 - Bayes Nets (22 points)

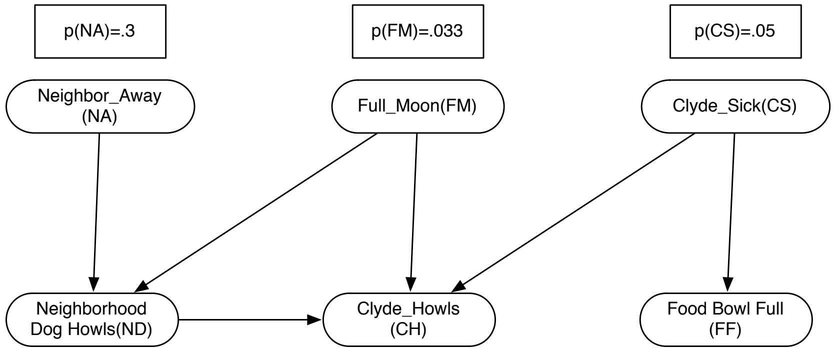
Your loyal dog Clyde has been howling for the last three hours and you want to decide whether or not to take him to the vet or just to put in ear plugs and go back to sleep. You know that Clyde often howls when there's a full moon, when he's genuinely sick, or occasionally when a particular neighborhood dog starts howling. That neighborhood dog sometimes howls at the full moon and sometimes howls when her owner isn't home, but is not affected by Clyde's howls. If Clyde's really sick he probably won't have eaten very much and should have a bunch of food left in his bowl - but he sometimes just isn't very hungry despite not being sick.

Question 4.1 (10 points)

Create a Bayesian network for the scenario described above - use single letter names for each boolean variable (explaining what they mean of course). You can make up the exact numbers in the conditional probability tables (CPTs), but both the CPT values and the causal topology of the network should be reasonable. Briefly explain why you are setting up the topology as you are.

Solution 4.1

Here's mine.



FM	NA	p(ND)
T	T	.8
T	F	.4
F	T	.5
F	F	0

FM	CS	ND	p(CH)
T	T	T	.99
T	T	F	.90
T	F	T	.65
T	F	F	.4
F	T	T	.75
F	T	F	.5
F	F	T	.2
F	F	F	0

CS	p(FF)
T	.6
F	.1

In terms of the CPTs, you just needed to make sure that adding multiple causes increased the probability of a thing - for instance, if the neighbor's dog is howling AND Clyde is sick it should be more probable that Clyde is howling than if only a single one of those causes was true. Additionally, you should have included something around $\frac{1}{28}$ for the prior probability of Full Moon (assuming you're on Earth, which is a reasonable assumption).

Question 4.2 (4 points)

Identify two conditional independencies in your Bayesian network - express them using the notation $A \perp B|C$. Explain why it makes sense that these variables are conditionally independent given the structure of the problem.

Solution 4.2

The easiest way to do this problem is to use the first fact on conditional independence in the book on page 499 - "A node is conditionally independent of its non-descendants given its parents". So for my graph, this means $(CH \perp NA|ND, FM, CS)$ and $(CH \perp FF|ND, FM, CS)$, $(FF \perp \text{Everything else}|CS)$, $(ND \perp CS|NA, FM)$, and so on. There can be additional conditional indepencies, but showing them generally involves using d-seperation style proofs, which we didn't talk about very much. Note that conditional independence is not the same things as real independence - obviously, the neighbor being away has nothing to do with the full moon (unless he's a werewolf) - it would be reasonable to say that those are truly independent events. But they are not conditionally independent, as once they are connected through the Neighbor's Dog Howling they become related. If the Neighbor's home, for instance, and the Neighbor's Dog is Howling it becomes a virtual certainty that there's a Full Moon given the structure of the problem.

Question 4.3 (4 points each)

For each of the following, write the expression you would need to compute the probabilities using your Bayes Net from 4.1 for inference. You don't need to compute the actual probabilities using your CPT values - just give the expression. Use the notation $p(A|B)$ to indicate the probability that A is true given that B is true - use $\neg A$ to indicate when A is false. Be sure to exploit the structure in your graphs to make the computation easier.

1. What is the probability that Clyde is sick given that there's no full moon, the neighbor's dog is howling, and that his food bowl is full?
2. What is the probability that Clyde is sick given that there's a full moon and you know your neighbor is away, but you have no other information?

Solution 4.3.1

First we will write the expression for Clyde being sick. Note that we don't need to consider the value of Neighbor Away, as having Neighbor Dog as evidence makes Clyde Sick conditionally independent.

$$\begin{aligned} P(CS|\neg FM, ND, FF, CH) &= \alpha P(CS, \neg FM, ND, FF, CH) \\ &= \alpha * P(CS)P(\neg FM)P(CH|CS, ND, \neg FM)P(FF|CS) \end{aligned}$$

I'm actually subbing in my values here, though you didn't have to

$$\begin{aligned} &= \alpha * .05 * .967 * .75 * .6 \\ &= \alpha * .02176 \end{aligned}$$

Now for Clyde Not Sick we have

$$\begin{aligned} P(\neg CS|\neg FM, ND, FF, CH) &= \alpha P(\neg CS, \neg FM, ND, FF, CH) \\ &= \alpha * P(\neg CS)P(\neg FM)P(CH|\neg CS, ND, \neg FM)P(FF|\neg CS) \\ &= \alpha * .95 * .967 * .2 * .1 \\ &= \alpha * .01837 \end{aligned}$$

Which gives us

$$\begin{aligned} P(CS) &= \frac{.02176}{.02176 + .01837} \\ &= 54.26\% \end{aligned}$$

Solution 4.3.2

First for true again - lowercase means that we don't know the value:

$$\begin{aligned} P(CS|FM, NA, CH) &= \alpha P(CS, FM, NA, CH, nd, ff) \\ &= \alpha \sum_{nd} \sum_{ff} P(CS)P(FM)P(CH|CS, FM, nd)P(nd|NA, FM)P(ff|CS) \\ &= \alpha P(CS)P(FM) \sum_{nd} P(CH|CS, FM, nd)P(nd|NA, FM) \sum_{ff} P(ff|CS) \end{aligned}$$

I'm actually subbing in my values here, though you didn't have to

$$\begin{aligned} &= \alpha * .05 * .033 * (.99 * .8 * (1) + .9 * .8 * (1)) \\ &= \alpha * .0024948 \end{aligned}$$

Same thing for Clyde Not Sick:

$$\begin{aligned}P(\neg CS|FM,NA,CH) &= \alpha P(\neg CS,FM,NA,CH,nd,ff)\\ &= \alpha \sum_{nd}\sum_{ff} P(\neg CS)P(FM)P(CH|\neg CS,FM,nd)P(nd|NA,FM)P(ff|\neg CS)\\ &= \alpha P(\neg CS)P(FM)\sum_{nd} P(CH|\neg CS,FM,nd)P(nd|NA,FM)\sum_{ff} P(ff|\neg CS)\end{aligned}$$

I'm actually subbing in my values here, though you didn't have to

$$\begin{aligned}&= \alpha * .95 * .033 * (.65 * .8 * (1) + .4 * .8 * (1))\\ &= \alpha * .026334\end{aligned}$$

$$\begin{aligned}P(CS|FM,NA,CH) &= \frac{.002498}{.0024948 + .026334}\\ &= 9.47\%\end{aligned}$$

Thus Clyde is probably healthy - though Clyde being sick could have caused his howling, if the neighbor is gone then the neighbor's dog is likely howling, and the full moon could also cause the howling. Being sick is thus Explained Away as a cause of the howling.