8 Computability and Introduction to Probability

Computer Science Mentors 70

Oct 26 - Oct 30
Fall 2020

Prepared by: Aishani Sil, Austin Lei, Agnibho Roy, Debayan Bandyopadhyay, Abinav Routhu, Sylvia Jin :ˆ)

Computability

1. Determine the computability of the following tasks. If it's not computable, write a reduction or self-reference proof. If it is, write the program. Hint: we can use the fact that TestHalt can't exist to prove other statements.

   (a) You want to determine whether a program P on input x prints "Hello World!" Is there a computer program that can perform this task? Justify your answer.

   Solution: Not Computable. Define a program that takes a program (you can think of it as a function), and transforms it into a program that runs the input program and then prints "Hello World!":

   ```
   def transform(P):
       def Q(x):
           P(x)
           print("Hello World!")
       return Q
   ```

   Let `testHelloWorld(P, x)` be our procedure that determines whether P eventually prints "Hello World!" when run on x. Now we want to solve the halting problem using this transformation. We write a function that uses `testHelloWorld` and `transform` to solve the halting problem:

   ```
   def halts(P, x):
       return testHelloWorld(transform(P), x)
   ```

   More compactly, we could write the entire thing like this:

   ```
   def halts(P, x):
       def Q(x):
           P(x)
           print("Hello World!")
       return testHelloWorld(Q, x)
   ```

   (b) You want to determine whether a program P prints "Hello World!" before running the kth line of the program.

   Solution: Not Computable. Similar to the last program, we can write a reduction program:

   ```
   def reduce(input):
       execute(input)
       print("Hello World!")
   ```

   (c) You want to determine whether a program P prints "Hello World!" in the first k steps of its execution. Is there a computer program that can perform this task?

   Solution: Computable. You can run the program until k steps are executed. If P has printed "Hello World!" by then, return true. If not, return false.

2. Say that we have a program M that decides whether any input program halts as long as it prints out the string "ABC" as the

first operation that it carries out. Can such a program exist? Prove your answer.

Solution: No. Such a program M can not exist. We proceed as follows: we show that if such a program existed, the halting problem would be computable.

Consider any program P. If we wanted to decide if P halted, we could simply create a new program P' where P' first prints out "ABC", then proceed to do exactly what P would. However, if M existed, we could determine whether any program P halted by converting it to a P' and feeding it into M. This would solve the halting problem - but this is a contradiction, since by diagonalization we can prove that the halting problem is not computable.

Therefore, M can not exist!

Below we have the Pseudocode for our program P.

```
define M'(Program p, input i):
    Construct program p' which will first print('ABC')
        then run p(i)
    run M(p', i)

define M(Program p, input i):
    if p's first command is print('ABC'):
        if p(i) halts:
            return True
        else if p(i) loops forever:
            return False
```

Introduction to Probability

1. **Rolling Die**

   Leanne rolls two fair six-sided die. For parts (c) through (e), imagine that Leanne wishes to compute the probability that the sum of the two die is 5.

   (a) How could you represent the sample space?

   Solution: We can represent the sample space as ordered pairs (i, j), where i represents the first die and j represents the second die.

   (b) What is the size of the sample space?

   Solution: There are 6 choices for i and 6 choices for j, so there are 36 ordered pairs in the sample space.

   (c) How could we represent the event we are looking for?

   Solution: We are looking for ordered pairs (i, j) such that $i + j = 5$.

   (d) What is the size of the event?

   Solution: There are 4 ordered pairs that work; $(1, 4)$, $(2, 3)$, $(3, 2)$, and $(4, 1)$.

   (e) What is the probability that Leanne rolls a 5?

Solution: The probability is the size of the event over the size of the sample space, or $\frac{4}{36} = \frac{1}{9}$.

2. **Probably Poker**

   (a) What is the probability of drawing a hand with four of a kind (four cards of the same rank)?

   Solution: $\frac{13*48}{\binom{52}{5}}$. In this approach, we're working over a sample space where the order in which we draw cards doesn't matter (shown in the denominator), so our numerator is counting the number of distinct combinations where order doesn't matter. We choose a rank and this selects 4 cards (13 ways). A poker hand is 5 cards, so to complete the poker hand, we have 48 cards left to choose from.

   (b) What is the probability of drawing a straight (five cards in numerical order)?

   Solution: $\frac{10*4^5}{\binom{52}{5}}$. There are 10 possible straights to choose from: A-5, 2-6, 3, 7, ..., 10-A. For each of those straights, for each number, there are 4 suits to choose from, so there are 4^5 ways to choose cards.

   (c) What is the probability of drawing a flush (five cards all of the same suit)?

   Solution: $\dfrac{4 * \binom{13}{5}}{\binom{52}{5}}$. First, choose a suit (4 suits). Then choose 5 cards from each suit which has 13 cards $\binom{13}{5}$.

   (d) What is the probability of drawing a straight flush (five cards in numerical order, all of the same suit)?

   Solution: $\frac{4*10}{\binom{52}{5}}$. First choose a suit (4 ways). Then choose a straight (10 ways).

   (e) What is the probability of drawing a hand with exactly one pair (two cards of matching rank)?

   Solution: $\dfrac{13 * \binom{4}{2} * \left(\binom{12}{3} * 4^3 + 12 * \binom{4}{3}\right)}{\binom{52}{5}}$. Choose a rank for the pair (13 ways) and 2 suits ($\binom{4}{2}$ ways). Now we need 3 other cards to complete our poker hand. There are 2 cases:

   Case 1: All three are from different ranks. There are 12 ranks left to choose 3 ranks ($\binom{12}{3}$ ways) and for each rank there are 4 ways to choose a suit (4^3 ways).
   Case 2: All three are the same rank. There are 12 ways to choose a rank and $\binom{4}{3}$ ways to choose the 3 suits.

   These are the only two cases that satisfy the criterion of having exactly one pair.

3. Oski is rolling 10 six-sided dice; what is the probability that he gets exactly two 4s?

   Solution: $\binom{10}{2}\left(\frac{1}{6}\right)^2\left(\frac{5}{6}\right)^8$

4. Eric is looking at the weather forecast for the next few days. On Monday, there is a 30% chance it will rain, and a 70% chance it will be sunny. On Tuesday, there is a 20% chance it will rain, and a 80% chance it will be sunny. The weather on Monday does not affect the weather on Tuesday. What is the probability that there will be rain on at least one day?

Solution:
Solution 1: We apply the Principle of Inclusion-Exclusion. Let A be the event that it rains on Monday and let B be the event that it rains on Tuesday. We wish to compute $P(A \cup B)$, which by PIE is $P(A) + P(B) - P(A \cap B)$. We know that $P(A) = 0.3$, $P(B) = 0.2$, and $P(A \cap B) = 0.3 \cdot 0.2 = 0.06$, since A and B are independent events. Thus, $P(A \cup B) = 0.3 + 0.2 - 0.06 = 0.44$.
Solution 2: Let's compute the probability of the complement of the event we want; the probability that it will not rain at all. That occurs with probability $0.7 \cdot 0.8 = 0.56$. Thus, the probability that there is rain is $1 - 0.56 = 0.44$.

5. For each permutation σ of 1 through n, let $\sigma(i)$ denote the value at position i. For example, if the permutation is $2, 4, 1, 3$ we have $\sigma(1) = 2$ and $\sigma(2) = 4$ (Source: CS 70 SP19 MT2)

   (a) For a fixed $1 \leq k \leq n$, what is the probability that a permutation σ of 1 through n satisfies the property that fo all $i < k, \sigma(i) < \sigma(k)$? Express your answer in terms of n and k.

   Solution: Consider the first k elements in the permutation. The biggest element of these k elements must be somewhere in the first k positions. By symmetry, $\frac{1}{k}$ of the permutations have $\sigma(k)$ being the largest of the first k elements in $\sigma(\cdot)$

   (b) What is the probability that a permutation of 1 through n satisfies the property that for each i, $\sigma(\sigma(i)) = i$ and $\sigma(i) \neq i$? (For example, the permutation $3, 4, 1, 2$ is such a permutation, since for example $\sigma(\sigma(1)) = \sigma(3) = 1$. You may assume n is even.

   Solution: We essentially want all sequences where two indices are swapped, which is equivalent to pairing off two indices at a time to be switched with each other. Thus, we can write the total combinations as:

$$\binom{n}{2}\binom{n-2}{2}\binom{n-4}{2}\cdots = \frac{n!}{2^{n/2}}$$

   We have over-counted here, however, since there is no labeling to each of the "pairs" of indices, thus, we need to divide by the number of ways to order the pairs, which is $\left(\frac{n}{2}\right)!$, leading to a total of:

$$\frac{n!}{2^{n/2}\left(\frac{n}{2}\right)!}$$

   Now we need to divide by the total number of permutations:

$$\frac{1}{2^{n/2}\left(\frac{n}{2}\right)!}$$

6. Eve is playing a game with Steve where she guesses the value (head or tail) of two coins that Steve has just flipped. She is told that at least one coin is facing heads. Given this information, what is the probability that both coins are facing heads?

Solution:
Intuitively... Our original sample space given the situation is $\{HH, HT, TH, TT\}$. However, since she knows one of the coins is heads, this eliminates the last event, so the new sample space is $\{HH, HT, TH\}$. Since all events are equally likely, one event out of 3 is HH so our probability is $\frac{1}{3}$.
Using the definition of conditional probability... we know that $P(A|B) = \frac{P(A \cap B)}{P(B)}$. We are trying to find P(both heads | at least one head). P(both heads) = $\frac{1}{4}$, and P(at least one head) = $\frac{3}{4}$, so plugging into the formula gives us $\frac{1}{3}$.