Collaborators : None

Sources : None

Q2 Simulating a biased coin

2-a) Prove that there is a simple subroutine (written in pseudocode using only deterministic computation and "coin flip" operations) that simulates a biased coin, which returns 0 with probability 1/4 and 1 with probability 3/4.

To simulate a biased coin that returns 0 with probability 1/4 and 1 with probability 3/4, we can use the following subroutine:

```
BiasedCoin():
  x = CoinFlip()
  y = CoinFlip()
  if x == Tails and y == Tails:
    return 0
  else:
    return 1
```

This subroutine works because:

- The probability of getting (Tails,Tails) from two coin flips is 1/4
- The probability of getting any other combination is 3/4

Therefore, this subroutine correctly simulates the desired biased coin.

2-b) Prove that for any $\varepsilon > 0$ there is a subroutine that almost simulates a biased coin, which returns a value $r \in \{0, 1, FAILURE\}$ such that $Pr(r = FAILURE) \leq \varepsilon$, and the conditional probability $Pr(r = 0 \mid r \neq FAILURE) = \frac{1}{3}$.

Remark: A subroutine is a program that always halts in $O(1)$ steps. In (b) this $O(1)$ could depend on ε.

$$Pr(r = 0 \mid r \neq FAILURE) = \frac{1}{3} \quad \Rightarrow \quad Pr(r = 1 \mid r \neq FAILURE) = \frac{2}{3}$$

Idea :

- Flip a standard coins (let's say 2 coins) atmost k times
- Each time a coin flips, match the outcomes to 1 or 0 such that their proportion is $2:1$
- Ensure that there is an outcome in each iteration of coins flip which isn't 1 or 0.
- After k iterations, if you didn't return 1 or 0 in any iteration, declare the outcome of the experiment to be $FAILURE$.
- Therefore, in each iteration, let's say if the outcome of 2 flipped coins is (Tails, Tails), the result is 0. If the outcome is (Tails, Heads) or (Heads, Tails), the result is 1. If the outcome is (Heads, Heads), we continue to next iteration of coin flips atmost k times.
- In each iteration, $Pr(outcome = 0) = \frac{1}{4}$ and $Pr(outcome = 1) = \frac{2}{4}$ and $Pr(outcome = ?) = \frac{1}{4}$
- $Pr(r = FAILURE) = \left(\frac{1}{4}\right)^k \leq \varepsilon \quad \Rightarrow \quad k \geq -log_4(\varepsilon)$
- $Pr(r = 0 \mid r \neq FAILURE) = \frac{Pr(outcome=0)}{Pr(outcome=0)+Pr(outcome=1)} = \frac{\frac{1}{4}}{\frac{1}{4}+\frac{2}{4}} = \frac{1}{3}$

To simulate a biased coin that returns 0 with probability 1/3 (conditional on not failing), 1 with probability 2/3 (conditional on not failing), and fails with probability at most ϵϵ, we can use the following subroutine:

```
AlmostBiasedCoin(ε):
  k = ceil(−log_4(ε))
  for i = 1 to k:
    x = CoinFlip()
    y = CoinFlip()
    if x == Tails and y == Tails:
      return 0
    if (x == Tails and y == Heads) or (y == Tails and x == Heads):
      return 1
  return FAILURE
```

This subroutine always halts in $O(1)$ steps because k is a constant for any fixed ϵ. The number of steps depends on ϵ, but for any given ϵ, it's a constant.