Reverse John von Neumann Problem

Steven Jaslar

December 17, 2003

1 Background

Suppose we are a given a coin. Flipping the coin yields two possible events: the coin either lands on heads or tails. We designate the probability that the coin lands on heads as $P(H)$ and the probability that the coin lands on tails as $P(T) = 1 - P(H)$. Because $P(H)$ and $P(T)$ are probabilities, their values lie in the real interval $[0, 1]$. We call the coin *unbiased* if $P(H) = P(T) = 1/2$. Naturally, we call the coin *biased* if $P(H) \neq P(T)$. Furthermore, we assume that repeated flips of the coin are *independent*, that is $P(H)$ and $P(T)$ are fixed.

We say an event *almost always* occurs if it occurs with probability one. For example, suppose we have an experiment where a person flips an unbiased coin until the coin lands on heads. It is possible that the coin always lands on tails. In this single case, the experiment does not end. However, the probability of this event is $\lim_{x\to\infty} P(T)^x$. So, the probability that the experiment ends is $1 - \lim_{x\to\infty} P(T)^x = 1 - \lim_{x\to\infty}(1/2)^x = 1$. Therefore, we say that this experiment almost always ends.

In 1951, John von Neumann is reported to have initially proposed and solved the following question[1]: given a biased coin with $0 < P(H) < 1$ is there a procedure to simulate an unbiased coin that almost always ends? John von Neumann answers in the affirmative. His procedure is as follows: Let $P(H)$ be the probability of a biased coin. At each step i, flip the coin twice and examine the four possible sequences. If the sequence is (Heads, Tails), report Heads. If the sequence is (Tails, Heads), report Tails. If the sequence is either (Heads, Heads) or (Tails, Tails), go to step $i + 1$. Thus, for each step, the Prob{reporting Heads} $= P(H) * P(T) = P(T) * P(H) =$ Prob{reporting Tails}. Furthermore, this algorithm has probability $2 * P(H) * P(T)$ of terminating at each step. So, the probabilty that this algorithm terminates is 1, and the expected running time in terms of number of coin flips is $\frac{1}{P(H)*P(T)}$.

Professor Greenfield proposes the "reverse" of the John von Neumann problem - that is, given an unbiased coin and a number p with $0 < p < 1$, is there a procedure to simulate a biased coin with $P(H) = p$ that almost always ends? In the following, I answer in the affirmative by demonstrating an algorithm that has an expected running time of 2 flips.

2 Reverse John von Neumann Problem

Let $p \ \epsilon \ [0,1]$. If $p = \Sigma_{i=1}^{\infty}(x_i * 2^{-i})$ and where $x_i \ \epsilon \ \{0,1\} \ \forall i \epsilon \mathbb{N}$ then we say the infinite sequence of $x_1, x_2, x_3, ...$ is the *binary representation* of p. Often, for convenience, the binary representation will be written as $.x_1x_2x_3...$

Lemma 1 *Every $p \ \epsilon \ [0,1]$ has a binary representation.*

The following algorithm may be used to recursively determine each x_j:

Step j:
If $p - (\Sigma_{i=1}^{j-1}(x_i * 2^{-i}) + 2^{-j}) \geq 0$, then $x_j = 1$
Otherwise, $x_j = 0$
Go to step $j + 1$

Obviously, this algorithm can always be executed at every step. Suppose, for sake of contradiction, that $p \neq \Sigma_{i=1}^{\infty}(x_i * 2^{-i})$. First observe, $\Sigma_{i=1}^{j}(x_i * 2^{-i}) < p$ $\forall j \epsilon \mathbb{N}$. So, $\exists \epsilon$ such that $\Sigma_{i=1}^{j}(x_i * 2^{-i}) < p - \epsilon \ \forall j \epsilon \mathbb{N}$. Furthermore, it follows from the Archimedean Principle that $\exists n \epsilon \mathbb{N}$ such that $2^{-n} < \epsilon < 2^{-n+1}$. Because $\Sigma_{i=1}^{n}(2^{-i}) = 1 - 2^{-n}$ and $p - 1 + 2^{-n} < \epsilon$, $\exists k \epsilon [n]$ such that $x_k = 0$ and $x_l = 1 \ \forall l \epsilon ([n] \setminus [k])$. Observe that by binary addition, $\Sigma_{i=1}^{n}(x_i * 2^{-i}) + 2^{-n}$ $= \Sigma_{i=1}^{k-1}(x_i * 2^{-i}) + 2^{-k} < p$. Yet this violates our choice of x_k. Hence, we have reached a contradiction and every real in the unit interval has a binary representation.

Example 1 *Construct the binary representation of 5/8.*

Step 1: $5/8 - 1/2 \geq 0$, so $x_1 = 1$;
Step 2: $5/8 - (1/2 + 1/4) < 0$, so $x_2 = 0$;
Step 3: $5/8 - (1/2 + 1/8) = 0$, so $x_3 = 1$;

So, $5/8 = .101$

Example 2 *Construct the binary representation of $1/\pi$.*

Step 1: $1/\pi$ - $(1/2) < 0$, so $x_1 = 0$;
Step 2: $1/\pi$ - $(1/4) \geq 0$, so $x_2 = 1$;
Step 3: $1/\pi$ - $(1/4 + 1/8) < 0$, so $x_3 = 0$;
Step 4: $1/\pi$ - $(1/4 + 1/16) \geq 0$, so $x_4 = 1$;

Step 5: $1/\pi$ - $(1/4 + 1/16 + 1/32) < 0$, so $x_5 = 0$;
Step 6: $1/\pi$ - $(1/4 + 1/16 + 1/64) < 0$, so $x_6 = 0$;

And so forth. $1/\pi$ = .010100...

Problem 1 *Given an unbiased coin and a number p with $0 < p < 1$, construct a procedure to simulate $P(H) = p$ that almost always ends.*

By Lemma 1, we can construct a binary representation for p. Let x_1, x_2, x_3... be a binary representation of p. I claim the following algorithm simulates an unfair coin with $P(H) = p$.

Step j:
  Flip unbiased coin and:
    If coin lands on heads and $x_j = 1$, report heads.
    If coin lands on tails and $x_j = 0$, report tails.
    Otherwise go to step $j + 1$.

First, this algorithm terminates with probability 1/2 at each step. This coincides with the geometric distribution with $p = 1/2$ and thus has expected running time equal to $1/(1/2) = 2$.

Notice that the probability of heads is the sum of the mutually exclusive events that we report heads on the i^{th} toss. Furthermore, we only report heads on the i^{th} toss if the procedure did not end in the previous $i - 1$ steps, the i^{th} toss is heads, and $x_i = 1$. So let $X' := \{x_i | x_i = 1\}$. Clearly, $P(H) = \Sigma_{i \epsilon X'} (x_i * 2^{-i}) = \Sigma_{i=1}^{\infty} (x_i * 2^{-i}) = p$.

Example 3 *Simulate a biased coin with P(H)=1/π by using an unbiased coin.*

¿From Example 2, we know that the binary representation of $1/\pi$ = .010100... We flip the unbiased coin. If the coin lands on tails, we report tails because $x_1 = 0$. If the coin lands on heads, we flip again. Suppose on the second flip, the coin lands on heads. Because $x_2 = 1$, we report heads. If the second flip is tails, we would flip again.

$$
\begin{aligned}
\text{So, } P(H) = & \ \text{Prob}\{\text{Flipping Heads with fair coin, then Heads again}\} \\
& + \text{Prob}\{\text{Flipping Heads, then Tails, then Heads, then Heads}\} \\
& + ... \\
= & \ 1/4 + 1/16 + ...
\end{aligned}
$$

$= .010100...$
$= 1/\pi$

Problem 2 *Given an biased coin and numbers $p_1 \neq p_2$ with $0 < p_1, p_2 < 1$, construct a procedure to simulate a biased coin with $P(H) = p_2$ from a biased coin with probability $P(H) = p_1$.*

First, use the John Von Neumann algorithm to simulate a fair coin from the biased coin. Then use the Reverse John Von Neumann algorithm to simulate the new biased coin from the simulated fair coin.

Due to the independence of coin flips, this procedure has expected running time $\frac{1}{P(H)*P(T)}*2 = \frac{2}{P(H)*P(T)}$.

Problem 3 *Given an unbiased coin and a rational number p in (0,1) construct a procedure to simulate a biased coin with $P(H) = p$ that does not use the binary representation of p.*

Let be p is a rational in (0,1). Then, $\exists q, r \epsilon \mathbb{N}$ such that $p = \frac{q}{r}$ and $\frac{q}{r}$ is in lowest terms. Let $x \epsilon \mathbb{N}$ such that $2^{x-1} < r \leq 2^x$. Observe that x flips of an unbiased coin produce 2^x distinct sequences of heads and tails that are equally likely. Mark q of these as *report heads*, $r - q$ as *report tails*, and the remaining $2^x - r$ as *repeat*.

Then, for each step i:

Flip the coin x times and call the resulting sequence X.

If X is marked as *report heads*, report heads.
If X is marked as *report tails*, report tails.
If X is marked as *repeat*, go to step $i + 1$.

Given that we report heads or tails, the probability that heads is reported is clearly the proportion $q/r = p$. The probability that we report heads or tails for some step i, is $1 - \lim_{i \to \infty}[\ \frac{r}{2^x}\ (\ \frac{2^x - r}{2^x})^{i-1}] = 1$. So, this algorithm successfully simulates a coin with $P(H) = p$.

The expected running time of this algorithm depends on r. In the best case, $\exists x \epsilon \mathbb{N}\ 2^x = r$, and, thus, there are no *repeat* sequences. In the worst case, $\exists x \epsilon \mathbb{N}\ 2^{x-1} + 1 = r$, and, thus, there are $2^{x-1} - 1$ *repeat* sequences. Observe that $\lim_{x \to \infty} \frac{2^{x-1}-1}{2^x} = \frac{1}{2}$. The expected running time of this algorithm is in the half-open interval $[x, 2 * x)$. However, the worst case running time is either x or infinity.

Example 4 *Given an unbiased coin, construct a procedure to simulate a biased coin with $P(H) = 2/3$ that does not use the binary representation of 2/3.*

Observe that $2^1 < 3 \leq 2^2$, so $x = 2$. Mark (Heads, Heads) and (Heads, Tails) as *report heads.* Mark (Tails, Heads) as *report tails.* Finally, mark (Tails, Tails) as *repeat.* For each step i flip the coin twice and report heads, report tails, or repeat for their respective sequence(s). The probability of reporting heads at each step is $\frac{2}{4} = \frac{1}{2}$. The probability of going to the next step is $\frac{1}{4}$. So, the probability of reporting heads is $\frac{2}{3}\Sigma_{i=1}^{\infty}[\frac{3}{4}*(\frac{1}{4})^{i-1}] = \frac{2}{3}$. The expected running time is $2 * \Sigma_{i=1}^{\infty}[i * (\frac{3}{4}) * (\frac{1}{4})^{i-1}] = \frac{8}{3} \leq 4 = 2 * x$.

3 References

.

[1] L. Fortnow, http://people.cs.uchicago.edu/ fortnow/talks/nvti.pdf, August 2002, 15 - 18.

[2] S. Greenfield, http://www.math.rutgers.edu/ greenfie/gs2003/pdfstuff/lect5.pdf, June 2003, 40-42.