USING MAPLE TO EXPLORE THE RUNS OF HEADS COIN TOSSING PROBLEM

Dr. Joel C. Fowler
Mathematics Department, Southern Polytechnic State University
1100 South Marietta Parkway, Marietta, GA 30060-2896
email: jfowler@spsu.edu

Determining the probability of k consecutive heads in n flips of a fair coin is a classic problem. While asymptotic results have been known for some time, exact answers tend to be computationally intractable. This makes the problem a good choice for a Maple project. We consider the theoretical solution and how Maple can be used to explore the problem, variations on it, and the asymptotic results.

The problem is easily modeled by letting sequences of coin flips correspond to binary strings, where a "1" is a head, and a "0" is a tail. Let a(n, k) = the number of binary strings of length n without a run of k consecutive 1's. Then the probability of a run of k heads in n flips = 1 - a(n, k)/2^n For example: a(3, 2) = 5 (000, 001, 010, 100, 101); a(5, 3) = 24 (all 2^5 = 32, except 11111, 11110, 11101, 10111, 01111, 11100, 01110, 00111) . a(n, k) has been extensively studied. A nice introduction to the problem and the literature on its solution can be found in [M. Schilling, The Longest Run of Heads, The College Mathematics Journal, Vol. 21, No. 3 (1990) 196 - 207]. The most elementary approach to the problem involves recurrence relations. We may divide the a(n, k) strings of length n without k consecutive 1's into cases based on the first appearance of 0 , from the left end of the string.

For position 1, the string is: **"0 [any length n-1 string w/o a run of k 1's]"** , hence a(n-1, k),

For position 2, the string is: **"1 0 [any length n-2 string w/o a run of k 1's]"** , hence a(n-2, k),

For position 3, the string is: **"1 1 0 [any length n-3 string w/o a run of k 1's]"** , hence a(n-3, k),

and so on, until the last case, which is k since the string does not contain k consecutive 1's:

for position k : **"1 1 1 1 ... 1 0 [any length n - k string w/o a run of k 1's]"** , hence a(n-k, k) .

Thus we have: a(n, k) = a(n - 1, k) + a(n - 2, k) + a(n - 3, k) + ... + a(n - k, k) , and the initial conditions: a(0, k) = 1, a(1, k) = 2, a(2, k) = 4, ... , a(k - 1, k) = 2^(k-1) . So each k determines a sequence that is recursive only in n. For example, consider k = 2 . Then a(n, 2) = a(n - 1, 2) + a(n - 2, 2); a(0, 2) = 1, a(1, 2) = 2 . This generates a shift of the Fibbonacci Numbers: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, . . .

The standard method for finding closed expressions for recurrence relations results in exponentials. In general, the solution to the p-th order linear recurrence relation: a(n) = c1*a(n - 1) + c2*a(n - 2) + ... + cp*a(n - p) is a linear combination of the roots of the polynomial: x^p = c1*x^(p-1) + c2*x^(p-2) + ... + cp, when those roots are distinct. Apart from exponential solution forms, Maple is efficient at computing exact values for a(n, k) directly from the recurrence relation and initial conditions. For convenience we replace our recurrence relation: a(n, k) = a(n - 1, k) + a(n - 2, k) + a(n - 3, k) + ... + a(n - k, k) , for n > k - 1, where a(0, k) = 1, a(1, k) = 2, a(2, k) = 4, ... , a(k - 1, k) = 2^(k - 1) with the form: a(n, k) = 2*a(n - 1, k) - a(n - k - 1, k) , for n > k, where a(0, k) = 1, a(1, k) = 2, a(2, k) = 4, ... , a(k , k) = 2^k - 1 . A program loads exact values of a(n,k) .

```
> MaxN := 15000: MaxK := 15:
> for k from 1 to MaxK do
      for n from 0 to k-1 do a(n,k) := 2^n end do;
      a(k,k) := 2^k-1:
      for n from k+1 to MaxN do a(n,k) := eval(2*a(n-1,k) -
  a(n-k-1,k))  end do;
  end do: unassign('k'): unassign('n'):
> a(100,7);
```

865145690433457063670671045568

Examining the probabilities of at least one run of k heads in n flips is now easy.

```
> Prob := (n,k) -> evalf(1 - a(n,k)/2^n):
> Prob(15000,15);
```

0.2044627046

How large n must be in order to achieve a given likelihood of a run of k heads can be found, brute force.

```
> Flips := proc(m::nonnegint, p::numeric)
           local i,r;
            for i from 1 to MaxN while Prob(i,m)<p do r:=i+1
  end do;
         r; end:
> Flips(10,.5);
```

1421

The brute force results for a(n, k) generated by Maple can be compared with the theoretical asymptotic results. To illustrate we examine the case k = 6.
According to the general theory on the solution of linear recurrences, we have that a(n, 6) is a linear combination of powers of the roots of

x^6 = x^5 + x^4 + x^3 + x^2 + x + 1 ,

since the recurrence is a(n, 6) = a(n - 1, 6) + a(n - 2, 6) + a(n - 3, 6) + ... + a(n - 6, 6). Equivalently, by multiplying by (x - 1) (which introduces the extraneous root of 1), and rearranging, we have x^7 - 2*x^6 + 1 = 0 . It is easy to see that f(x) = x^7 - 2*x^6 + 1 has no repeated roots, since the derivative has only 0 and 12/7 as roots. Maple handles the more difficult task of estimating all of the roots, and their magnitudes.

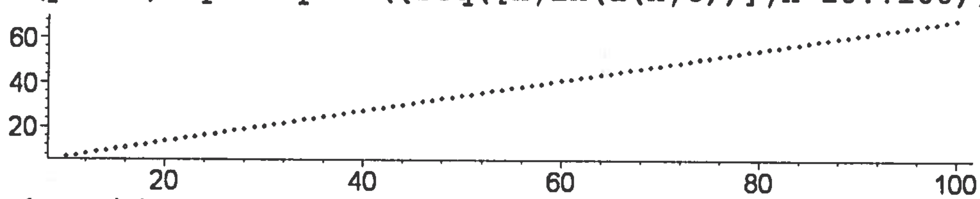
```
> evalf(allvalues(RootOf(x^7-2*x^6+1,x)));
  evalf(allvalues(abs(RootOf(x^7-2*x^6+1,x))));
```

1., 1.983582843, 0.3902920340 + 0.8178616563 *I*, -0.4619289065 + 0.7191443780 *I*, -0.8403090983, -0.4619289065 − 0.7191443780 *I*, 0.3902920340 − 0.8178616563 *I*

1., 1.983582843, 0.9062149638, 0.8547203935, 0.8403090983, 0.8547203935, 0.9062149638

Note that, apart from 1 (which is extraneous), all but one root has modulus less than 1. Hence the one dominant root should provide excellent asymptotic estimates for a(n, 6). That is, we would expect, since powers of all other roots approach 0, to find that a(n, 6) ~ c*b^n, where b is the root 1.983582843 . This can be verified using Maple's plotting and regression features. The exponential form implies the linear relationship: ln(a(n, 6)) ~ ln(c) + n*ln(b) .

```
> with(plots): pointplot({seq([n,ln(a(n,6))],n=10..100)});
```



The graph certainly suggests linearity. We can use the statistical routines to estimate the parameters in the relationship.

```
> with(Statistics): with(stats): Digits := 20:
> EqFit:= fit[leastsquare[[x,y]]]([ [seq(i,i=10..100)],
  [seq(evalf(ln(a(i,6))),i=10..100)]]);
```

EqFit := *y* = 0.034384368190973614397 + 0.68490468003303218847 *x*

```
> Correlation( [seq(i,i=10..100)],
  [seq(evalf(ln(a(i,6))),i=10..100)]);
```

0.99999999999994484242

A related problem is determining the number of binary strings without a run of k consecutive 0's or 1's. Let

b(n, k) = the number of length n binary strings without a run of k 0's or 1's

We use a matrix approach for the case k = 3, which eventually leads to a recurrence relation for all values of k . Define the column vector **v**(n) = <v0(n), v1(n), v2(n), v3(n) > , where v0(n) = the number of length n strings ending in 00 without a run of 3 zeros or ones; v1(n) = the number of length n strings ending in 01 without a run of 3 zeros or ones; v2(n) = the number of length n strings ending in 10 without a run of 3 zeros or ones; and v3(n) = the number of length n strings ending in 11 without a run of 3 zeros or ones. We then have, by considering each case in turn: v0(n+1) = v2(n), since such a string of length n+1 would have to end in 100; v1(n+1) = v0(n) + v2(n), since such a string of length n+1 would have to end in 001 or 101; v2(n+1) = v1(n) + v3(n), since such a string of length n+1 would have to end in 010 or 110; v3(n+1) = v1(n), since such a string of length n+1 would have to end in 011 . Hence we have the matrix relationship **v**(n+1) = **B** ***v**(n) with

```
> B := <<0,1,0,0>|<0,0,1,1>|<1,1,0,0>|<0,0,1,0>>;
```

$$B := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Note that since **v**(2) = <1,1,1,1>, **v**(n) = **B** ^(n-2) <1,1,1,1>. Also, b(n, 3) = v0(n) + v1(n) + v2(n) + v3(n) = [1 1 1 1] **v**(n) = [1 1 1 1] **B** ^(n-2) <1,1,1,1>. Thus the minimal polynomial for **B** yields a linear recurrence relation that the sequence must satisfy.

```
> with(LinearAlgebra): MinimalPolynomial(B,x);
```

$$-1 - 2x - x^2 + x^4$$

Hence we have the linear recurrence: b(n, 3) = b(n - 2, 3) + 2*b(n - 3, 3) + b(n - 4, 3), for n > 5, which, along with the initial conditions: b(0, 3) = 1, b(1, 3) = 2, b(2, 3) = 4, b(3, 3) = 6, b(4, 3) = 10, and b(5, 3) = 16, determines the sequence. Use of the minimal polynomial generates a linear recurrence, but not always one of lowest order. Using Maple, we can interate the powers of the matrix A to hunt for a pattern.

```
> seq(B^i .<1,1,1,1>, i=1..10);
```

$$\begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ 3 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 5 \\ 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 5 \\ 8 \\ 8 \\ 5 \end{bmatrix}, \begin{bmatrix} 8 \\ 13 \\ 13 \\ 8 \end{bmatrix}, \begin{bmatrix} 13 \\ 21 \\ 21 \\ 13 \end{bmatrix}, \begin{bmatrix} 21 \\ 34 \\ 34 \\ 21 \end{bmatrix}, \begin{bmatrix} 34 \\ 55 \\ 55 \\ 34 \end{bmatrix}, \begin{bmatrix} 55 \\ 89 \\ 89 \\ 55 \end{bmatrix}, \begin{bmatrix} 89 \\ 144 \\ 144 \\ 89 \end{bmatrix}$$

The vector **v**(n) is composed of essentially one sequence of numbers that are repeated and shifted as the powers of **B** increase. That is **v**(n) = <d(n-1),d(n),d(n),d(n-1)> . Applying the relationship **v**(n) = **B*****v**(n-1) , then shows that d(n) must satisfy the recurrence d(n) = d(n-1) + d(n-2). Since b(n, 3) is the sum of the entries in **v**(n), it too must satisfy that recurrence. Once the initial cases are disposed of, we have that b(n, 3) is determined by:

b(n, 3) = b(n-1, 3) + b(n-2, 3), for n > 2; b(0, 3) = 1, b(1, 3) = 2, b(2, 3) = 4

Note that this is the same recurrence relation for no 2 consecutive ones, i.e. a(n, 2). In fact, by comparing the initial conditions for each sequence we have:

b(n, 3) = 2 * a(n - 1, 2) , for n > 1 . This surprising result holds in general. That is,

b(n, k) = 2 * a(n - 1, k - 1) , for n > 1 .

To see why this is true in general, first note that the number of strings, without k consecutive ones or zeros, that end in a 1 is the same as the number that end in 0. This is because the strings ending in 0 may be mapped, one to one, to the strings ending in 1 by simple complementation. Thus, for these strings:

(# of length n ending in 1) = (# of length n ending in 0) = b(n, k) / 2

Now note that we may divide all of the strings in b(n, k) into cases depending on the length of the first run of consecutive identical bits, from the right:

If the run is of length 1, then the string is:

"[a length n - 1 string ending in 1] 0 " or **"[a length n - 1 string ending in 0] 1 "**, hence (1/2)*b(n - 1, k) in either case . If the run is of length 2, then the string is:

"[a length n - 2 string ending in 1] 0 0" or **"[a length n - 2 string ending in 0] 1 1"**, hence (1/2)*b(n - 2, k) in either case. If the run is of length 3, then the string is

"[a length n - 3 string ending in 1] 0 0 0" or **"[a length n - 3 string ending in 0] 1 1 1"**, hence (1/2)*b(n-3, k) in either case. The last case is k - 1, since the string doesn't contain k consecutive ones or zeros. If the run is of length k - 1, then the string is:

"[a length n-(k-1) string ending in 1] 0 0...0" or **"[a length n-(k-1) string ending in 0] 1 1...1"**, hence (1/2)*b(n-(k-1), k) in either case..

Summing over these cases results in b(n, k) = b(n - 1, k) + b(n - 2, k) + ... + b(n - (k - 1), k) , the same recurrence as that for a(n , k - 1). Finally we note that, in general: b(1, k) = 2 = 2*1 = 2*a(0, k - 1) , b(2, k) = 4 = 2*2 = 2*a(1, k - 1) , b(3, k) = 8 = 2*4 = 2*a(2, k - 1) , and so on, until b(k, k) = 2^k - 2 = 2*(2^(k - 1) - 1) = 2*a(k - 1, k - 1) . Thus, since 2*a(n - 1, k - 1) satisfies the same recurrence and initial conditions as b(n, k), it is b(n, k). Hence, we can easily find the probability of, for example, at least one run of 6 heads or tails in 200 flips of a fair coin as

```
> evalf(1 - 2*a(200-1,6-1)/2^200);
```

$$0.9653128011576403047$$

Last, we note that this matrix approach, with Maple, is adaptable to enumerating strings with various pattern avoidances. As an example, consider the strings of length n that not containing the patterns "00" or "111". Let c(n) = # strings of length n without 00 or 111 . Then as before, define the column vector **v**(n) with components: v0(n) = # of length n strings ending in 00 without 00 or 111; v1(n) = # of length n strings ending in 01 without 00 or 111; v2(n) = # of length n strings ending in 10 without 00 or 111; and v3(n) = # of length n strings ending in 11 without 00 or 111 . We then have, by considering each case in turn: v0(n+1) = 0, since such a string would immediately contain 00; v1(n+1) = v2(n), since such a string of length n+1 would have to end in 101; v2(n+1) = v1(n) + v3(n), since such a string of length n+1 would have to end in 010 or 110; v3(n+1) = v1(n), since such a string of length n+1 would have to end in 011 . Thus we have the transition matrix from **v**(n) to **v**(n+1), for n > 1, with the initial vector **v**(2) = <0,1,1,1> and its minimal polynomial:

```
> C:=<<0,0,0,0>|<0,0,1,1>|<0,1,0,0>|<0,0,1,0>>:MinimalPolyno
  mial(C,x);
```

$$-x - x^2 + x^4$$

This yields c(n) = c(n-2) + c(n-3) for n > 3; c(0) = 1, c(1) = 2, c(2) = 3, c(3) = 4 . In this case, iteration produces the same recurrence relation.