1 Recurrence Relations

They are recursive definitions of a sequence: expressions of the form \( a_n = G(a_{n-1}, \ldots, a_{n-k+1}) \) together with \textit{initial conditions} \( a_1 = \beta_1, \ldots, a_k = \beta_k \) (the \( \beta_i \) are constants – elements of \( \mathbb{R} \)). We can produce the sequence by applying \( G() \) first to the first \( k \) \( a \)'s to get \( a_{k+1} \), and computing successive elements. They are a kind of recursive program to compute elements of the sequence.

1.1 Fibonacci Sequence

The original motivation was the following:

Consider a pair of rabbits They are immortal, and produce pairs of off-springs every two months. The number of new pairs in month \( i + 2 \) is the number of pairs in month \( i \). Starting with 1 pair in month 1 we have the table of (month, number of pairs) below:

<table>
<thead>
<tr>
<th>Month</th>
<th>Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>34</td>
</tr>
</tbody>
</table>
It is easy to see that the description of this sequence is given by the recurrence
\[ f_n = f_{n-1} + f_{n-2} \]
with initial conditions \( f_1 = f_2 = 1 \).
Note: the sequence was defined by the Pisan mathematician Fibonacci (Filius Bonacci—the son of Bonacci) in 1202.

1.2 Tower of Hanoi

Read about it in Rosen – including a nice myth about it bringing about the end of the world when done (for some Buddhists, this is supposed to be a good thing!)

For us, the important part is that the solution for \( n \) requires a number of moves, \( H_n \), that is given by

- moving \( n-1 \) disks from peg 1 to peg 3 (using \( H_{n-1} \) moves)
- moving the \( n \)-th disk from peg 1 to peg 2 (1 move)
- moving \( n-1 \) disks from peg 3 to peg 2 (using \( H_{n-1} \) moves)

This yields the recurrence
\[ H_n = 2H_{n-1} + 1 \]
with initial condition \( H_1 = 1 \)
that has \( H_n = 2^n - 1 \) as a solution.

For those with time on their hands, read the relevant SF short story ‘The nine billion names of god’ by Arthur C. Clarke.

1.3 Strings of length \( n \) with no 00

We want to count the number of binary strings of length \( n \) that do not have a substring of 2 consecutive 0s. Note that the ‘obvious’ first attempt, namely consider 00 as a new symbol, and count the number of ways in could be in each of the \( 2^{n-2} \) binary strings of length \( n - 2 \) (it is easy to see that there are \( n - 1 \) possibilities) does not work! It is clear that this count is wrong, since for \( n > 4 \) the number we get, \( (n-2)2^{n-2} \) is greater than \( 2^n \), the number of binary strings of length \( n \). Can you see why the count is wrong?

We use recurrence relations as follows: let \( a_n \) denote the number of binary string with no 00 as a substring.

Clearly, \( a_1 = 2 \), and \( a_2 = 3 \).
Consider a string $x_1x_2 \cdots x_n$ in the set. There are 2 possible values for the last bit $x_n$; it can be 0 or 1.

Let us count the number of strings that end in 1. The first $n - 1$ bits, $x_1x_2 \cdots x_{n-1}$ can be any binary string of length $n - 1$ that has no 00 substring. The number of such strings is $a_{n-1}$.

Now let us count the number of strings of length $n$ with no 00 substring, that end in 0. The bit in position $n - 2$ must be a 1 (otherwise we have a 00). But then we know that $x_1 \cdots x_{n-2}$ can be any binary string of length $n - 2$ with no 00 substring. The number of such strings is $a_{n-2}$.

So the number of strings, $a_n$ satisfies the recurrence relation

$$a_n = a_{n-1} + a_{n-2} \text{ for } n > 2.$$ 

Our joy in discovering that this is exactly the recurrence relation for Fibonacci numbers is soon tempered by the realization that the initial conditions are different (for Fibonacci $f_1 = 1$, while we have $a_1 = 2$.)

A closer look yields that in fact $a_1 = 2 = f_3$, and $a_2 = 3 = f_4$. Since the recurrence is identical, we conclude that

$$a_n = f_{n+2}$$

### 1.4 Another Example

How many decimal strings of length $n$ have an even number of 0s?

We do a similar case analysis. Let $a_n$ be the number of strings with an even number of 0s and $b_n$ be the number of strings with an odd number of 0s. Clearly $a_n + b_n = 10^n$

We reason as follows: if we have a string $x_1 \cdots x_{n-1}$ of length $n - 1$ with an even number of 0s, the string $x_1 \cdots x_{n-1}x_n$ will have an even number of 0s iff $x_n \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. The number of such strings is $9a_{n-1}$

$x_1 \cdots x_{n-1}$ of length $n - 1$ with an odd number of 0s, the string $x_1 \cdots x_{n-1}x_n$ will have an even number of 0s iff $x_n = 0$. The number of such strings is $b_{n-1} = 10^{n-1} - a_{n-1}$.

So $a_n = 9a_{n-1} + 10^{n-1} - a_{n-1} = 8a_{n-1} + 10^{n-1}$