Modeling with Recurrence Relations

Another example:

Let $a_n$ denote the number of bit strings of length $n$ that do not have two consecutive 0s ("valid strings"). Find a recurrence relation and give initial conditions for the sequence $(a_n)$.

Solution:

Idea: The number of valid strings equals the number of valid strings ending with a 0 plus the number of valid strings ending with a 1.

Modeling with Recurrence Relations

Now we need to know: How many valid strings of length $n$ are there, if the string ends with a 0?

Valid strings of length $n$ ending with a 0 must have a 1 as their $(n – 1)$st bit (otherwise they would end with 00 and would not be valid).

And what is the number of valid strings of length $(n – 1)$ that end with a 1?

We already know that there are $a_{n-1}$ strings of length $n$ that end with a 1.

Therefore, there are $a_{n-2}$ strings of length $(n – 1)$ that end with a 1.

So there are $a_{n-2}$ valid strings of length $n$ that end with a 0 (all valid strings of length $(n – 2)$ with 10 appended to them).

That gives us the following recurrence relation:

$$a_n = a_{n-1} + a_{n-2}$$

What are the initial conditions?

$a_1 = 2$ (0 and 1)

$a_2 = 3$ (01, 10, and 11)

$a_3 = a_2 + a_1 = 3 + 2 = 5$

$a_4 = a_3 + a_2 = 5 + 3 = 8$

$a_5 = a_4 + a_3 = 8 + 5 = 13$

This sequence satisfies the same recurrence relation as the Fibonacci sequence.

Since $a_1 = f_1$ and $a_2 = f_4$, we have $a_n = f_{n+2}$.

Solving Recurrence Relations

In general, we would prefer to have an explicit formula to compute the value of $a_n$, rather than conducting $n$ iterations.

For one class of recurrence relations, we can obtain such formulas in a systematic way.

Those are the recurrence relations that express the terms of a sequence as linear combinations of previous terms.
Solving Recurrence Relations

**Definition:** A linear homogeneous recurrence relation of degree k with constant coefficients is a recurrence relation of the form:

\[ a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k}, \]

where \( c_1, c_2, \ldots, c_k \) are real numbers, and \( c_k \neq 0 \).

A sequence satisfying such a recurrence relation is uniquely determined by the recurrence relation and the k initial conditions

\[ a_0 = C_0, \ a_1 = C_1, \ a_2 = C_2, \ldots, a_{k-1} = C_{k-1}. \]

Solving Recurrence Relations

**Examples:**

The recurrence relation \( P_n = (1.05)P_{n-1} \) is a linear homogeneous recurrence relation of degree one.

The recurrence relation \( f_n = f_{n-1} + f_{n-2} \) is a linear homogeneous recurrence relation of degree two.

The recurrence relation \( a_n = a_{n-5} \) is a linear homogeneous recurrence relation of degree five.

Solving Recurrence Relations

Basically, when solving such recurrence relations, we try to find solutions of the form \( a_n = r^n \), where \( r \) is a constant.

\( a_n = r^n \) is a solution of the recurrence relation \( a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k} \) if and only if

\[ r^n = c_1r^{n-1} + c_2r^{n-2} + \ldots + c_kr^0. \]

Divide this equation by \( r^{n-k} \) and subtract the right-hand side from the left:

\[ r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_{k-1}r - c_k = 0 \]

This is called the characteristic equation of the recurrence relation.

Solving Recurrence Relations

The solutions of this equation are called the characteristic roots of the recurrence relation.

Let us consider linear homogeneous recurrence relations of degree two.

**Theorem:** Let \( c_1 \) and \( c_2 \) be real numbers. Suppose that \( r^2 - c_1r - c_2 = 0 \) has two distinct roots \( r_1 \) and \( r_2 \).

Then the sequence \( \{a_n\} \) is a solution of the recurrence relation \( a_n = c_1a_{n-1} + c_2a_{n-2} \) if and only if

\[ a_n = \alpha_1r_1^n + \alpha_2r_2^n \]

for \( n = 0, 1, 2, \ldots \), where \( \alpha_1 \) and \( \alpha_2 \) are constants.


Solving Recurrence Relations

Example: What is the solution of the recurrence relation \( a_n = a_{n-1} + 2a_{n-2} \) with \( a_0 = 2 \) and \( a_1 = 7 \)?

**Solution:**

The characteristic equation of the recurrence relation is \( r^2 - r - 2 = 0 \).

Its roots are \( r_1 = 2 \) and \( r_2 = -1 \).

Hence, the sequence \( \{a_n\} \) is a solution to the recurrence relation if and only if:

\[ a_n = \alpha_12^n + \alpha_2(-1)^n \]

for some constants \( \alpha_1 \) and \( \alpha_2 \).
Solving Recurrence Relations

Another Example: Give an explicit formula for the Fibonacci numbers.

Solution: The Fibonacci numbers satisfy the recurrence relation $f_n = f_{n-1} + f_{n-2}$ with initial conditions $f_0 = 0$ and $f_1 = 1$.

The characteristic equation is $r^2 - r - 1 = 0$.

Its roots are $r_1 = \frac{1 + \sqrt{5}}{2}$, $r_2 = \frac{1 - \sqrt{5}}{2}$.

Therefore, the Fibonacci numbers are given by

$$f_n = \alpha_1 \left(\frac{1 + \sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1 - \sqrt{5}}{2}\right)^n$$

for some constants $\alpha_1$ and $\alpha_2$.

We can determine values for these constants so that the sequence meets the conditions $f_0 = 0$ and $f_1 = 1$:

$$f_0 = \alpha_1 + \alpha_2 = 0$$

$$f_1 = \alpha_1 \left(\frac{1 + \sqrt{5}}{2}\right) + \alpha_2 \left(\frac{1 - \sqrt{5}}{2}\right) = 1$$

Solving these equations yields $\alpha_1 = 1$ and $\alpha_2 = 1$.

Consequently, the overall solution is given by

$$f_n = \left(\frac{1 + \sqrt{5}}{2}\right)^n + \left(\frac{1 - \sqrt{5}}{2}\right)^n$$

Solving Recurrence Relations

The unique solution to this system of two equations and two variables is

$$\alpha_1 = \frac{1}{\sqrt{5}}, \quad \alpha_2 = -\frac{1}{\sqrt{5}}$$

So finally we obtained an explicit formula for the Fibonacci numbers:

$$f_n = \left(\frac{1 + \sqrt{5}}{2}\right)^n - \left(\frac{1 - \sqrt{5}}{2}\right)^n$$

Solving Recurrence Relations

Example: What is the solution of the recurrence relation $a_n = 6a_{n-1} - 9a_{n-2}$ with $a_0 = 1$ and $a_1 = 6$?

Solution: The only root of $r^2 - 6r + 9 = 0$ is $r_0 = 3$.

Hence, the solution to the recurrence relation is $a_n = \alpha_1 3^n + \alpha_2 n3^n$ for some constants $\alpha_1$ and $\alpha_2$.

To match the initial conditions, we need

$$a_0 = 1 = \alpha_1$$

$$a_1 = 6 = \alpha_1 3 + \alpha_2 3$$

Solving these equations yields $\alpha_1 = 1$ and $\alpha_2 = 1$.

Consequently, the overall solution is given by

$$a_n = 3^n + n3^n$$

Solving Recurrence Relations

But what happens if the characteristic equation has only one root?

How can we then match our equation with the initial conditions $a_0$ and $a_1$?

**Theorem:** Let $c_1$ and $c_2$ be real numbers with $c_2 \neq 0$.

Suppose that $r^2 - c_1r - c_2 = 0$ has only one root $r_0$.

A sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ if and only if

$$a_n = \alpha_1 r_0^n + \alpha_2 nr_0^n$$

for $n = 0, 1, 2, \ldots$, where $\alpha_1$ and $\alpha_2$ are constants.

Divide-and-Conquer Relations

Some algorithms take a problem and successively divide it into one or more smaller problems until there is a trivial solution to them.

For example, the binary search algorithm recursively divides the input into two halves and eliminates the irrelevant half until only one relevant element remained.

This technique is called “divide and conquer”.

We can use recurrence relations to analyze the complexity of such algorithms.
Divide-and-Conquer Relations

Suppose that an algorithm divides a problem (input) of size \( n \) into \( a \) subproblems, where each subproblem is of size \( n/b \). Assume that \( g(n) \) operations are performed for such a division of a problem.

Then, if \( f(n) \) represents the number of operations required to solve the problem, it follows that \( f \) satisfies the recurrence relation

\[
f(n) = af(n/b) + g(n).
\]

This is called a divide-and-conquer recurrence relation.

**Example:** The binary search algorithm reduces the search for an element in a search sequence of size \( n \) to the binary search for this element in a search sequence of size \( n/2 \) (if \( n \) is even).

Two comparisons are needed to perform this reduction.

Hence, if \( f(n) \) is the number of comparisons required to search for an element in a search sequence of size \( n \), then

\[
f(n) = f(n/2) + 2 \text{ if } n \text{ is even}.
\]

Usually, we do not try to solve such divide-and-conquer relations, but we use them to derive a big-O estimate for the complexity of an algorithm.

**Theorem:** Let \( f \) be an increasing function that satisfies the recurrence relation

\[
f(n) = af(n/b) + cn^d
\]

whenever \( n = bk \), where \( k \) is a positive integer, \( a, c, \) and \( d \) are real numbers with \( a \geq 1 \), and \( b \) is an integer greater than 1. Then \( f(n) \) is

- \( O(n^d) \), if \( a < b^d \),
- \( O(n^d \log n) \) if \( a = b^d \),
- \( O(n^{\log_b a}) \) if \( a > b^d \).

**Example:**

For binary search, we have \( f(n) = f(n/2) + 2 \), so \( a = 1 \), \( b = 2 \), and \( d = 0 \) (\( d = 0 \) because here, \( g(n) \) does not depend on \( n \)).

Consequently, \( a = b^d \), and therefore, \( f(n) \) is \( O(n^2 \log n) = O(\log n) \).

The binary search algorithm has logarithmic time complexity.