SICP 1.19

byronluk

June 2019

1 Question

There is a clever algorithm for computing the Fibonacci numbers in a logarithmic number of steps. Recall the transformation of the state variables a and b in the fib-iter process of 1.2.2: $a \leftarrow a + b$ and $b \leftarrow a$. Call this transformation T, and observe that applying T over and over again n times, starting with 1 and 0, produces the pair $Fib(n + 1)$ and $Fib(n)$. In other words, the Fibonacci numbers are produced by applying T^n, the n^{th} power of the transformation T, starting with the pair (1, 0). Now consider T to be the special case of $p = 0$ and $q = 1$ in a family of transformations T_{pq}, where T_{pq} transforms the pair (a,b) according to $a \leftarrow bq + aq + ap$ and $b \leftarrow bp + aq$. Show that if we apply such a transformation T_{pq} twice, the effect is the same as using a single transformation $T_{p'q'}$ of the same form, and compute p' and q' in terms of p and q. This gives us an explicit way to square these transformations, and thus we can compute T^n using successive squaring, as in the fast-expt procedure.

2 Solution

$$\begin{aligned} a_1 &= bq + aq + ap \\ b_1 &= bp + aq \end{aligned} \tag{1}$$

$$\begin{aligned} a_2 &= b_1 q + a_1 q + a_1 p \\ b_2 &= b_1 p + a_1 q \end{aligned} \tag{2}$$

Solving for a_2

$$\begin{aligned} a_2 &= (bp + aq)q + (bq + aq + ap)q + (bq + aq + ap)p \\ &= bpq + aq^2 + bq^2 + aq^2 + apq + bpq + apq + ap^2 \\ &= bq^2 + 2bpq + aq^2 + 2apq + ap^2 + aq^2 \\ &= b(q^2 + 2pq) + a(q^2 + 2pq) + a(p^2 + q^2) \end{aligned} \tag{3}$$

Solving for b_2

$$\begin{aligned} b_2 &= (bp+aq)p + (bq+aq+ap)q \\ &= bp^2 + apq + bq^2 + aq^2 + apq \\ &= b(p^2+q^2) + a(q^2+2pq) \end{aligned} \tag{4}$$

$$\begin{aligned} bq+aq+ap &= b(q^2+2pq) + a(q^2+2pq) + a(p^2+q^2) \\ bp+aq &= b(p^2+q^2) + a(q^2+2pq) \end{aligned} \tag{5}$$

$$\begin{aligned} q' &= q^2+2pq \\ p' &= p^2+q^2 \end{aligned} \tag{6}$$