SICP Answers to Exercises

Yin Wang
Department of Computer Science and Technology
Tsinghua University
Beijing, China

1 Building Abstractions with Procedures

2 Building Abstractions with Data

3 Modularity, Objects, and State

Exercise 3.66

Examine the stream (*pairs integers integers*). Can you make any general comments about the order in which the pairs are placed into the stream? For example, about how many pairs precede the pair (1,100)? the pair (99,100)? the pair (100,100)? (If you can make precise mathematical statements here, all the better. But feel free to give more qualitative answers if you find yourself getting bogged down.)

Answer:

I am not bogged down :) The closed form solution can be deduced recursively as follows.

Let $P_{m,m}$ denote the pairs in the upper triangular block starting with the pair (S_m, T_m). Let L_m denote the pairs in the first row of the triangle, excluding (S_m, T_m). $P_{m,m}$ is composed of three parts: the pair (S_m, T_m), the line L_m, and the block $P_{m+1,m+1}$. This partition scheme of $P_{0,0}$ is depicted in the following table:

(S_0, T_0)	(S_0, T_1)	(S_0, T_2)	$\ldots$	$\ldots$
	(S_1, T_1)	(S_1, T_2)	$\ldots$	(S_m, T_n)
		(S_2, T_2)	$\ldots$	$\ldots$
			$\ldots$	$\ldots$

Now we are going to find how many pairs precede the pair (S_m, T_n). Observe that the pair (S_m, T_n) is inside the block $P_{m,m}$. Before (S_m, T_n), we must have enumerated $(S_m, T_{m+1}) \ldots (S_m, T_{n-1})$, a total of $n - m - 1$ pairs on the line L_m. According to the interleave procedure:

```
(define (pairs s t)
  (cons-stream
   (list (stream-car s) (stream-car t))
   (interleave
    (stream-map (lambda (x) (list (stream-car s) x))
                (stream-cdr t))
    (pairs (stream-cdr s) (stream-cdr t)))))
```

we must have also enumerated $n - m - 1$ pairs in the block $P_{m+1,m+1}$. Plus the pair (S_m, T_m), we have enumerated $2(n - m - 1) + 1$ pairs in the block $P_{m,m}$ before we reach (S_m, T_n). We denote the number $2(n - m - 1) + 1$ by N_0.

According to the interleave procedure, $N_0 + 2$ pairs must have been enumerated on the line L_{m-1}, just above the block $P_{m,m}$. So we have enuerated $2N_0 + 2$ pairs in the block $P_{m-1,m-1}$. We denote this number by N_1. Continue this argument, we have:

$$
\begin{aligned}
N_0 &= 2(n-m-1)+1 \\
N_1 &= 2N_0+2 \\
N_2 &= 2N_1+2 \\
\cdots & \quad \cdots \\
N_m &= 2N_{m-1}+2
\end{aligned}
$$

We can get N_m as follows:

$$
\begin{aligned}
N_m &= 2N_{m-1}+2 \\
&= 2(2N_{m-2})+2 \\
&= 2^2N_{m-2}+2^2+2 \\
&= 2^2(2N_{m-3}+2)+2^2+2 \\
&= 2^3N_{m-3}+2^3+2^2+2 \\
&= \cdots \\
&= 2^mN_0+2^2+2^{m-1}+\cdots+2 \\
&= 2^m(2(n-m-1)+1)+(2-2^{m+1})/(1-2) \\
&= (n-m)\cdot 2^{m+1}+2^m-2
\end{aligned}
$$

This is the number of pairs before pair (S_m, T_n). Note, the pairs is counted from (S_0, T_0). Thus the pairs precede (99,100) is the pairs precede (S_{98}, T_{99}). This can be calculated as $2^{99}+2^{98}-2$, which is very large!

To verify our formula is correct. We can define a precedure before that can count the pairs before a specified integer pair. And we define another procedure before-closed to calculate the closed form. We can get the same results from the two procedures. We see the closed form procedure is much faster.

```
(define (before m n stream)
  (let loop ((i 0) (s stream))
    (if (equal? (stream-car s) (list m n))
        i
        (loop (+ 1 i) (stream-cdr s)))))
```

closed form

```
(define (before-closed m n stream)
  (let ((m (- m 1)) (n (- n 1)))
    (+ (* (- n m) (expt 2 (+ m 1))) (expt 2 m) -2)))

(before 13 40 int-pairs)
(before-closed 13 40 int-pairs)
```