Recurrences for the man or boy sequence

Eric Schmidt

June 24, 2021

In this note, we prove some recurrences that can be used to compute the "man or boy" sequence (OEIS A132343). In particular, we prove a recurrence added without proof to the OEIS entry by Markus Jarderot. Our analysis of the sequence is similar to that indicated by Donald Knuth in the ALGOL Bulletin, 19.2.3.4. In that note, Knuth presented recurrences essentially the same as those in the below theorem, but did not say why they held. He also presented a table similar to the one we provide (though we have independently computed the values). We will use our own C++ program (presented in the OEIS entry), rather than Knuth's original, for understanding the algorithm.

While various generalizations of the sequence are possible, the only one we consider is to vary the values for the input parameters 1, -1, -1, 1, 0 in the initial call to A. It is convenient to view these parameters as indeterminates $x_1, \ldots, x_5$. Then, the result of the computation, for an index k, will be a linear combination $v(k) = \sum_i a_i(k)x_i$. The $a_i(k)$ have a natural interpretation: $a_i(k)$ is the number of times that xi is called during the computation of $v(k)$. We will see that Jarderot's recurrence holds even for v, and hence for the integer sequence arising from substituting any particular values for the indeterminates.

For negative k, the result is the same as when $k = 0$. Thus, we can safely restrict attention to the case where $k \geq 0$. It is evident that $a_5(0) = 1$ and $a_5(k) = 0$ for $k > 0$, so it is just the other four sequences that we need to worry about. By direct computation, we can determine $a_i(k)$ for small values of k:

k	$a_1(k)$	$a_2(k)$	$a_3(k)$	$a_4(k)$	$a_5(k)$
0	0	0	0	1	1
1	0	0	1	1	0
2	0	1	1	0	0
3	1	1	0	0	0
4	2	1	0	0	0
5	3	2	1	0	0
6	5	3	3	2	0
7	8	6	9	6	0
8	14	15	22	13	0
9	29	37	48	26	0
10	66	85	102	54	0

In particular, the result of Knuth's original program is $66 - 85 - 102 + 54 = -67$.

There are a couple of issues that need to be settled before we prove any recurrences. First, it is not immediately apparent that the computation terminates. (This question is a bit more subtle than it may seem. If the expression x4() + x5() is replaced by one that calls x1, an infinite recursion results.) Second, the computation at first appears ambiguous: in the expression x4() + x5(), the evaluation of x4() and x5() can have side effects, introducing the concern that the computed result may depend on the order of evaluation. However, this turns out not to be the case.

Theorem. *The computation terminates and does not depend on the order of evaluation. For $k \geq 6$,*

$$\begin{aligned}
a_1(k) &= a_1(k-1) + a_2(k-1), \\
a_2(k) &= a_2(k-1) + a_3(k-1), \\
a_3(k) &= a_3(k-1) + a_4(k-1) + a_1(k-1) - 1, \\
a_4(k) &= a_4(k-1) + a_1(k-1) - 1.
\end{aligned}$$

Proof. We obtain $v(k)$ by calling A(k, x1, x2, x3, x4, x5), where xi is a lambda that just returns x_i. Inside A, we first create a lambda B, which has access to the parameters of A. The value of k, which decreases whenever B is called, will be called the *index*. When k is at most 0, A immediately returns x4() + x5(), which evaluates to $x_4 + x_5$. Otherwise, we call B. This decreases the index to $k-1$, and then $v(k)$ is obtained as the result of calling A($k-1$, B, x1, x2, x3, x4). By induction, we may assume that $v(j)$ is well-defined for $j < k$. The execution of A($k-1$, B, x1, x2, x3, x4) mirrors that for $v(k-1)$, but the $a_1(k-1)$ calls to x1 in the computation of $v(k-1)$ are now instead calls to B. We will call these the *top-level B calls.* The computation will terminate as long as all of these calls do.

The calls to B decrease the index and then call A(j, B, x1, x2, x3, x4), where j is the index. Again, these calls terminate, provided all resulting further calls to B terminate. However, once the index reaches 0, the calls to A just return x4() + x5(), which evaluates to $x_3 + x_4$, so there are no further calls to B introduced. Hence, the entire computation terminates.

To see that the evaluation order does not matter, we may assume so for all $j < k$. In particular, the number of top-level B calls does not depend on the evaluation order. Additionally, the top-level B calls do not affect the rest of the computation, and affect each other only by the decreasing of the index. Since we are assuming by induction that the evaluation order within these calls does not matter, the order for the whole computation does not matter.

Now we turn to the recurrences. Other than the top-level B calls, the computation of A($k-1$, B, x1, x2, x3, x4) will result in $a_{i+1}(k-1)$ calls to xi, for $1 \leq i \leq 4$. This accounts for the second terms in the recurrences for a_1, a_2, and a_3. We omit the term $a_5(k-1)$ from the formula for $a_4(k)$, since $a_5(k-1) = 0$. To analyse the top-level B calls, let us assume for now that $a_1(k-1) \geq k-2$. In particular, there is at least one top-level B call. The index will be $k-1$ when the first such call is made. Now, the computation of $v(k-1)$ creates a B lambda, with index $k-1$, and calls it, and so this computation is identical to that for the first top-level B call. This accounts for the first terms in the recurrences for a_i. Moreover, our assumption that $a_1(k-1) \geq k-2$ implies that during the first top-level B call, the index decreases by at least $k-2$, so will be at most 1 afterwards. There are $a_1(k-1)-1$ remaining top-level B calls. Each of them results in a call A(j, B, x1, x2, x3, x4) with $j \leq 0$, which results in $x_3 + x_4$. This accounts for $a_1(k-1)-1$ appearing in the recurrences for a_3 and a_4. The recurrences have now been established for all k for which our assumption holds.

To prove the recurrences for all $k \geq 6$, we first look at the above table to see that they do hold for $k = 6$. For $k > 6$, we use induction to simultaneously prove $a_1(k-1) \geq k-2$ (our prior assumption) and $a_2(k-1) \geq 1$. The table shows that these inequalities hold for $k = 7$. Moreover, if they hold for some k, then we know the recurrences hold for that k, so we apply the recurrences to see that $a_1(k) = a_1(k-1)+a_2(k-1) \geq (k-2)+1 = k-1$ and $a_2(k) = a_2(k-1)+a_3(k-1) \geq a_2(k-1) \geq 1$, showing that the inequalities hold for $k+1$. The proof is complete. □

Jarderot's recurrence is now straightforward:

Corollary. *For $k \geq 10$, we have*

$$v(k) = 5v(k-1) - 10v(k-2) + 11v(k-3) - 6v(k-4) + v(k-5).$$

Proof. This may be verified mechanically, by using the theorem to write both sides in terms of $v(k-5)$. Indeed, for the left side, we find that

$$\begin{aligned}
a_1(k) &= 16a_1(k-5) + 11a_2(k-5) + 11a_3(k-5) + 10a_4(k-5) - 15,\\
a_2(k) &= 21a_1(k-5) + 16a_2(k-5) + 11a_3(k-5) + 11a_4(k-5) - 21,\\
a_3(k) &= 22a_1(k-5) + 21a_2(k-5) + 16a_3(k-5) + 11a_4(k-5) - 22,\\
a_4(k) &= 11a_1(k-5) + 11a_2(k-5) + 10a_3(k-5) + 6a_4(k-5) - 11,
\end{aligned}$$

and the same formulas give the components of the right side. $\square$