Superexponential growth in text manipulation

Arjen Markus

July 2, 2023

Preface

I wrote the original text quite a few years ago and posted it on the Tclers' Wiki (`https://wiki.tcl-lang.org`). As I introduced it then:

> This lengthy title is meant to introduce a subject for the mathematically interested. It has to do with replacing strings by other strings in a short text and suddenly finding memory has exhausted.

Introduction

This note discusses a small problem in the area of text manipulation. To get some feeling for this subject, it is perhaps good to know the circumstances under which it arose. I was testing a small utility that would edit text files based on a few simple commands: you could replace strings that matched a certain pattern, by something else, you could insert the contents of another text file after a line containing a string pattern or replace that text by it. That was (and is) about all the utility did.

While I was testing it, I used the same small text file (some 20 lines of text) over and over again and though I had noticed that its size grew, I had not expected to run out of memory – at last the utility needed some 100 MB of memory all of a sudden.

This led me to consider the mathematics of the process in more detail. We can strip it down to an example like below:

- Take the string "AbbAbb"
- Insert after each "A" the original string and repeat this procedure
- The resulting strings are:
  Initial string: `AbbAbb`
  After one round: `AAbbAbbbbAAbbAbbbb`
  After two rounds:
  `AAAbbAbbbbAAbbAbbbbAAAbbAbbbbAAbbAbbbbbbAAAbbAbbbbAAbbAbbbbbbbbAAAbbAbbbbAAbbAbbbbAAAbbAbbbbAAbbAbbbbbbAAAbbAbbbbAAbbAbbbbbbbb`

As you can see, it is growing very fast!

Analysing the problem

There are a number of variations possible on the original problem:

1. Insert a fixed string that does not contain ”A”
2. Insert a fixed string that does contain one or more ”A”s
3. Replace ”A” by a such a fixed string
4. Replace ”A” by the string from the previous iteration, rather than insert that string after the ”A”.

Let us examine what happens in each of these cases:
Ad 1.
As the number of ”A”s does not change, the length of the resulting string is a linear function of the iteration step: at each step nm characters are added, where n is the number of ”A”s in the original string and m is the length of the inserted string:

$$\begin{aligned} N(k+1) &= N(k) = n \\ L(k+1) &= L(k) + nm \end{aligned}$$

Ad 2.
If, at step k, the string contains $N(k)$ ”A”s and the fixed string contains n ”A”s and has a total length m, then the resulting string from the next round has $N(k)$ extra copies of the fixed string. So:

$$\begin{aligned} N(k+1) &= N(k) + N(k)n = (n+1)N(k) \\ L(k+1) &= L(k) + N(k)m \end{aligned}$$

The solution is:

$$\begin{aligned} N(k) &= (n+1)^{k-1}N(0) \\ L(k) &= L(0) + N(0)m\frac{(n+1)^k - 1}{n} \end{aligned}$$

Ad 3.
When replacing the ”A”, the formula changes, as $N(k)$ ”A”s will be deleted:

$$\begin{aligned} N(k+1) &= N(k) + nN(k) - N(k) = nN(k) \\ L(k+1) &= L(k) + (m-1)N(k-1) \end{aligned}$$

And the solution is:

$$\begin{aligned} N(k) &= n^(k-1)N(0) \\ L(k) &= L(0) + m(n^k - 1)/(n-1)N(0) \end{aligned}$$

Ad 4.
In this case $N(k)$ "A"s will be replaced by a string of length $L(k)$ and containing $N(k)$ "A"s as well. So:

$$\begin{aligned} N(k+1) &= N(k)N(k) = N(k)^2 \\ L(k+1) &= L(k) + N(k)(L(k)-1) = N(k)L(k) \end{aligned}$$

A closed solution for N(k) is easy to find:

$$N(k) = N(0)^{2^k}$$

The solution for L(k) is found when you take the logarithm of the above equation:

$$\ln L(k+1) = \ln L(k) + \ln N(k) = \ln L(k) + 2^k \ln N(0)$$

Therefore:

$$L(k) = L(0) + 2^{2^{k+1}-1} N(0)$$

Ad 5.
The original problem is described by the following equations:

$$\begin{aligned} N(k+1) &= N(k) + N(k)N(k) = (1+N(k))N(k) \\ L(k+1) &= L(k) + N(k)L(k) = (1+N(k))L(k) \end{aligned}$$

As far as I know this problem has no closed solution.
Below is a table that compares these five problems (N(0) = n = 2, L(0) = m = 6):

Iteration	*Case 1*	*Case 2*	*Case 3*	*Case 4*	*Case 5*
k	Nk, Lk	Nk, Lk	Nk, Lk	Nk, Lk	Nk, Lk
0	2, 6	2, 6	2, 6	2, 6	2, 6
1	2, 18	6, 18	4, 16	4, 12	6, 18
2	2, 30	18, 54	8, 36	16, 48	42, 126
3	2, 42	54, 162	16, 76	256, 748	1806, 5418
4	2, 54	162, 486	32, 156	65536, 196608	3263442, 9790326

Comments from fellow Tclers:

Kevin Kenny: Arjen, you're starting off on an expedition into the wonderful world of computability theory. I haven't looked at your recurrence in detail, but I'd conjecture that it's bounded by some expression involving Ackermann's function `http://mathworld.wolfram.com/AckermannFunction.html`.

It is possible, using calculations similar to yours, to define recursive functions that grow faster than any tower of exponentials:

```
f(x) > 2^(2^(2^(...(2^x)...)))  for any value of n !
       \_______/
          n times
```

In fact, Ackermann's function was the first function discovered that is recursively enumerable but not primitive recursive. In computer geek terms, that means that the function can be computed with while-loops, but not with counted for-loops.

Arjen Markus I know. It caught me by surprise. My fifth case involves a power of 2. Unfortunately, my knowledge of the theory (beyond reading a book on Turing machines, another on NP-completeness and sundry publications of similar sort that happened to pass my way) is rather limited. It is more or less recreational mathematics for me, it was fun to investigate the problem on my way to work and to write it up during a week in China (hotels get boring, don't they). I have passed a message to MathWorld, but got no reply as yet.

Arjen Markus There is a limiting solution for case 5. I do not have the details right now, but you can show that it is delimited by two functions similar to case 4.

Andreas Kupries: The string replacement algorithm above seems to resemble the type of substitution done by Lindenmeyer system (L-systems). One application of L-systems is to model the growth of a plant. This has applications in computer graphics to generate naturalistically looking plants (`https://wiki.tcl-lang.org/page/Pixar`).

Arjen Markus: A long time ago, I read an article about this in the Scientific American. It was in the heyday of fractals :-)