Solutions to Recurrences

Introduction

A recurrence is a definition of values of a function in terms of previous values of the function. To be complete, a definition of a function using a recurrence must have a non-recursive branch. However, if the object is to express the value of the function asymptotically, the non-recursive branch may not be necessary, as in the examples we give here.

In most of our examples, the solution is expressed in Θ notation, but sometimes we need to write O or Ω.

Anti-Derivative Method

The derivative f' of a real-valued function f is defined as fillows: $f'(x) = \lim_{h \to 0} \frac{f(x) - f(x-h)}{h}$

In asymptotic analysis, we only need h to be "close" to zero. How close? A general rule is that h must be asymptotically smaller than x.

1. $F(n) = F(n-1) + n$ We can write $\frac{F(n) - F(n-1)}{1} = n$

   1 is close to zero, so we have $F'(n) = \Theta(n)$, from which we obtain hence $F(n) = \Theta(n^2)$.

2. $F(n) = F(n - \sqrt{n}) + n$ We can write $\frac{F(n) - F(n - \sqrt{n})}{\sqrt{n}} = \frac{n}{\sqrt{n}}$

   $\sqrt{n}$ is close enough to zero that the left-hand-side is asymptotically the derivative of F.

   We have $F'(n) = \Theta(\sqrt{n})$ Taking the anti-derivative, we obtain $F(n) = \Theta(n^{3/2})$

The Master Theorem

If we have the recurrence $F(n) = A\,F(n/B) + n^C$ where A, B, and C are constants, where $A > 0$, $B > 1$, and $C \geq 0$

$$F(n) = \begin{cases} \Theta(n^C \text{ if } B^C > A \\ \Theta(n^C \log n) \text{ if } B^C = A \\ \Theta\big(n^{\log_B A}\big) \text{ if } B^C < A \end{cases}$$

3. $F(n) = F(n/2) + 1$

   $A = 1$, $B = 2$, and $C = 0$, and $B^C = A$. Thus $F(n) = \Theta(n^C \log n) = \Theta(n^0 \log n) = \Theta(\log n)$.

4. $F(n) = 2F(n/2) + n$

   This is one of the most commonly occuring recurrences. $A = 2$, $B = 2$, and $C = 1$. Thus $B^C = A$. We obtain $F(n) = \Theta(n \log n)$.

5. $F(n) = 2F(n/2) + 1$

   $A = 2, B = 2$, and $C = 0$. $B^C < A$. $\log_2 2 = 1$, and $n^1 = n$. Thus $F(n) = \Theta(n)$.

6. $F(n) = 2F(n/2) + n^2$

   $A = 2, B = 2$, and $C = 2$. $B^C > A$. Thus $F(n) = \Theta(n^C) = \Theta(n^2)$.

The Generalized Master Theorem

We change the notation to Greek letters, changing A to α, $1/B$ to β, and C to γ, for example. The recurrence $F(n) = AF(n/B) + n^C$ is now written $F(n) = \alpha F(\beta n) + n^\gamma$.

In the generalized master theorem, we allow multiple terms on the right hand side, each with its own α_i and β_i. The general form of the recurrence is

$$F(n) = \alpha_1 F(\beta_1 n) + \alpha_2 F(\beta_2 n) + \cdots + \alpha_k F(\beta_k n) + n^\gamma$$

To solve the recurrence, we first compute $\Gamma = \sum_{i=1}^{k} \alpha_i \beta_i^\gamma$ If $\Gamma = 1$, then $F(n) = \Theta(n^\gamma \log n)$. If $\Gamma < 1$, then $F(n) = \Theta(n^\gamma)$. The hard case is $\Gamma > 1$. We need to find a constant δ such that $\sum_{i=1}^{k} \alpha_i \beta_i^\delta = 1$. Then $F(n) = \Theta(n^\delta)$.

7. The following recurrence gives the time complexity of the BFPRT algorithm, also known as the "median of medians" algorithm for selecting the k^{th} smallest item in an array. $F(n) \leq 2\,F(n/5) + F(n/2) + n$

   $k = 2$, $\alpha_1 = 2$, $\beta_1 = \frac{1}{5}$, $\alpha_2 = 1$, $\beta_2 = \frac{1}{2}$, and $\gamma = 1$.

   $\Gamma = 2\,\frac{1}{5} + \frac{1}{2} = \frac{9}{10} < 1$. Thus $F(n) = O(n^\gamma) = O(n)$. However, for other reasons, the complexity is actually $\Theta(n)$.

8. $F(n) = F(n/3) + F(n/6) + F(n/2) + n$

   $k = 3$, $\alpha_1 = 1$, $\beta_1 = \frac{1}{3}$, $\alpha_2 = 1$, $\beta_2 = \frac{1}{6}$, $\alpha_3 = 1$, $\beta_3 = \frac{1}{2}$, $\gamma = 1$.

   $\Gamma = \frac{1}{3} + \frac{1}{6} + \frac{1}{2} = 1$. Thus $F(n) = \Theta(n^\gamma \log n) = \Theta(n \log n)$.

9. $F(n) = F(3n/5) + F(4n/5) + n^2$

   $k = 2$, $\alpha_1 = 1$, $\beta_1 = \frac{3}{5}$, $\alpha_2 = 1$, $\beta_2 = \frac{4}{5}$, $\gamma = 2$.

   $\Gamma = \left(\frac{3}{5}\right)^2 + \left(\frac{4}{5}\right)^2 = 1$. Thus $F(n) = \Theta(n^\gamma \log n) = \Theta(n^2 \log n)$.

10. $F(n) = 2\,F(2n/3) + F(n/3) + n$

    $k = 2$, $\alpha_1 = 2$, $\beta_1 = \frac{2}{3}$, $\alpha_2 = 1$, $\beta_2 = \frac{1}{3}$. $\gamma = 1$.

    $\Gamma = 2\left(\frac{2}{3}\right) + \frac{1}{3} = \frac{5}{3} > 1$. Therefore, we must find δ such that $2\left(\frac{2}{3}\right)^\delta + \left(\frac{1}{3}\right)^\delta = 1$. The correct value is $\delta = 2$. Thus $F(n) = \Theta(n^2)$.

Substitution

We can sometimes use substitution to transform a recurrence into one which we can solve using one of the above methods.

11. $F(n) = F(\sqrt{n}) + 1$

    Define a new function G by letting $G(m) = F(2^m)$ for any m. Now, let $m = \log_2 n$, hence $G(m) = F(n)$ and $F(n) = G(\log_2 n)$, which implies that $F(\sqrt{n}) = G(\log_2(\sqrt{n})) = G(\frac{1}{2}\log_2 n = G(m/2)$. Substituting in the original recurrence, we obtain $G(m) = G(m/2) + 1$. From Example 3 above, we have $G(m) = \Theta(\log m)$, hence $F(n) = G(m) = \Theta(\log m) = \Theta(\log\log n)$.

12. $F(n) = 2\,F(\sqrt{n}) + \log n$

    We use the same substitution as in the previous problem, namely $m = \log_2 n$ and $G(m) = F(n)$ We obtain $G(m) = 2\,G(m/2) + m$. By Example 4, we have $G(m) = \Theta(m\log m) = \Theta(\log n \log\log n)$.

13. $F(n) = 2\,F(n-1) + 1$

    You can problably immediately guess that the solution is exponential. We can obtain the solution by substitution: We define $G(m) = F(\log_2 m)$. Let $m = 2^n$ equivalently, $n = \log_2 m$. Thus $F(n) = G(2^n) = G(m)$ and $F(n-1) = G(2^{n-1}) = G(2^n/2) = G(m/2)$. Substituting in the original recurrence we have: $G(m) = 2G(m/2) + 1$ From Example 5 we have $G(m) = \Theta(m)$. Thus $F(n) = G(m) = \Theta(m) = \Theta(2^n)$.

More Generalizations of the Master Theorem

There are other, even more sophisticated, generalizations of the master theorem. You can find these on the internet, for example, in Wikipedia.

Other

14. $F(n) = F(\log n) + 1$

    The function $\log^* x$, the so-called iterated logarithm, is defined recursively, as follows:

    $$\log^* x = \begin{cases} 0 \text{ if } x \leq 1 \\ 1 + \log^*(\log x) \text{ otherwise} \end{cases}$$

    The solution to our recurrence is then $F(n) = \Theta(\log^* n)$. Think of $\log^* x$ this way. Enter x onto your calculator. If $x \leq 1$, then $\log^* x$ is zero. Otherwise, push the $\boxed{\log}$ button on your calculator until you see a number which is less than or equal to 1. The number of times you pushed that button is $\log^* x$. What is $\log^*$ of the number of atoms in the known universe?

I have given you a few easy-to-understand methods, which are sufficient to solve many practical recurrences. But there are recurrences whose solution requires more advanced methods, and some which have no closed form solution.