Fundamentals of Algorithmics. Gilles Brassard & Paul Bratley. Prentice Hall. 1996

The characteristic polynomial is $(x-4)^2$ and hence all solutions are of the form

$$t_i = c_1 4^i + c_2\, i\, 4^i.$$

In terms of $T(n)$, this is

$$T(n) = c_1 n^2 + c_2 n^2 \lg n. \tag{4.27}$$

Substituting Equation 4.27 into the original recurrence yields

$$n^2 = T(n) - 4T(n/2) = c_2 n^2$$

and thus $c_2 = 1$. Therefore

$$T(n) \in \Theta(n^2 \log n \mid n \text{ is a power of } 2),$$

regardless of initial conditions (even if $T(1)$ is negative). □

Example 4.7.12. Consider the recurrence

$$T(n) = 2T(n/2) + n \lg n$$

when n is a power of 2, $n \ge 2$. As before, we obtain

$$\begin{aligned} t_i = T(2^i) &= 2T(2^{i-1}) + i\,2^i \\ &= 2t_{i-1} + i\,2^i \end{aligned}$$

We rewrite this in the form of Equation 4.10.

$$t_i - 2t_{i-1} = i\,2^i$$

The characteristic polynomial is $(x-2)(x-2)^2 = (x-2)^3$ and hence all solutions are of the form

$$t_i = c_1 2^i + c_2\, i\, 2^i + c_3\, i^2 2^i.$$

In terms of $T(n)$, this is

$$T(n) = c_1 n + c_2 n \lg n + c_3 n \lg^2 n. \tag{4.28}$$

Substituting Equation 4.28 into the original recurrence yields

$$n \lg n = T(n) - 2T(n/2) = (c_2 - c_3)\, n + 2c_3\, n \lg n,$$

which implies that $c_2 = c_3$ and $2c_3 = 1$, and thus $c_2 = c_3 = \frac{1}{2}$. Therefore

$$T(n) \in \Theta(n \log^2 n \mid n \text{ is a power of } 2),$$

regardless of initial conditions. □

Remark: In the preceding examples, the recurrence given for $T(n)$ only applies when n is a power of 2. It is therefore inevitable that the solution obtained should be in conditional asymptotic notation. In each case, however, it is sufficient to add the condition that $T(n)$ is eventually nondecreasing to be able to conclude that the asymptotic results obtained apply unconditionally for all values of n. This follows from the smoothness rule (Section 3.4) since the functions $n^{\lg 3}$, $n^2 \log n$ and $n \log^2 n$ are smooth.

Example 4.7.13. We are now ready to solve one of the most important recurrences for algorithmic purposes. This recurrence is particularly useful for the analysis of divide-and-conquer algorithms, as we shall see in Chapter 7. The constants $n_0 \ge 1$, $\ell \ge 1$, $b \ge 2$ and $k \ge 0$ are integers, whereas c is a strictly positive real number. Let $T : \mathbb{N} \to \mathbb{R}^+$ be an eventually nondecreasing function such that

$$T(n) = \ell\, T(n/b) + cn^k \qquad n > n_0 \tag{4.29}$$

when n/n_0 is an exact power of b, that is when $n \in \{bn_0, b^2 n_0, b^3 n_0, \ldots\}$. This time, the appropriate change of variable is $n = b^i n_0$.

$$\begin{aligned} t_i = T(b^i n_0) &= \ell\, T(b^{i-1} n_0) + c(b^i n_0)^k \\ &= \ell\, t_{i-1} + c\, n_0^k\, b^{ik} \end{aligned}$$

We rewrite this in the form of Equation 4.10.

$$t_i - \ell\, t_{i-1} = (cn_0^k)\,(b^k)^i$$

The right-hand side is of the required form $a^i p(i)$ where $p(i) = cn_0^k$ is a constant polynomial (of degree 0) and $a = b^k$. Thus, the characteristic polynomial is $(x-\ell)(x-b^k)$ whose roots are ℓ and b^k. From this, it is tempting (but false in general!) to conclude that all solutions are of the form

$$t_i = c_1 \ell^i + c_2 (b^k)^i. \tag{4.30}$$

To write this in terms of $T(n)$, note that $i = \log_b(n/n_0)$ when n is of the proper form, and thus $d^i = (n/n_0)^{\log_b d}$ for arbitrary positive values of d. Therefore,

$$\begin{aligned} T(n) &= (c_1/n_0^{\log_b \ell})\, n^{\log_b \ell} + (c_2/n_0^k)\, n^k \\ &= c_3\, n^{\log_b \ell} + c_4\, n^k \end{aligned} \tag{4.31}$$

for appropriate new constants c_3 and c_4. To learn about these constants, we substitute Equation 4.31 into the original recurrence.

$$\begin{aligned} cn^k &= T(n) - \ell\, T(n/b) \\ &= c_3\, n^{\log_b \ell} + c_4\, n^k - \ell\,(c_3 (n/b)^{\log_b \ell} + c_4 (n/b)^k) \\ &= \left(1 - \frac{\ell}{b^k}\right) c_4\, n^k \end{aligned}$$

Therefore $c_4 = c/(1 - \ell/b^k)$. To express $T(n)$ in asymptotic notation, we need to keep only the dominant term in Equation 4.31. There are three cases to consider, depending whether ℓ is smaller than, bigger than or equal to b^k.

- If $\ell < b^k$ then $c_4 > 0$ and $k > \log_b \ell$. Therefore the term $c_4 n^k$ dominates Equation 4.31. We conclude that $T(n) \in \Theta(n^k \mid (n/n_0)$ is a power of $b)$. But n^k is a smooth function and $T(n)$ is eventually nondecreasing by assumption. Therefore $T(n) \in \Theta(n^k)$.
- If $\ell > b^k$ then $c_4 < 0$ and $\log_b \ell > k$. The fact that c_4 is negative implies that c_3 is positive, for otherwise Equation 4.31 would imply that $T(n)$ is negative, contrary to the specification that $T : \mathbb{N} \to \mathbb{R}^+$. Therefore the term $c_3 n^{\log_b \ell}$ dominates Equation 4.31. Furthermore $n^{\log_b \ell}$ is a smooth function and $T(n)$ is eventually nondecreasing. Therefore $T(n) \in \Theta(n^{\log_b \ell})$.
- If $\ell = b^k$, however, we are in trouble because the formula for c_4 involves a division by zero! What went wrong is that in this case the characteristic polynomial has a single root of multiplicity 2 rather than two distinct roots. Therefore Equation 4.30 does *not* provide the general solution to the recurrence. Rather, the general solution in this case is

$$t_i = c_5 (b^k)^i + c_6\, i\, (b^k)^i.$$

In terms of $T(n)$, this is

$$T(n) = c_7 n^k + c_8 n^k \log_b(n/n_0) \qquad (4.32)$$

for appropriate constants c_7 and c_8. Substituting this into the original recurrence, our usual manipulation yields a surprisingly simple $c_8 = c$. Therefore, $c n^k \log_b n$ is the dominant term in Equation 4.32 because c was assumed to be strictly positive at the beginning of this example. Since $n^k \log n$ is smooth and $T(n)$ is eventually nondecreasing, we conclude that $T(n) \in \Theta(n^k \log n)$.

Putting it all together,

$$T(n) \in \begin{cases} \Theta(n^k) & \text{if } \ell < b^k \\ \Theta(n^k \log n) & \text{if } \ell = b^k \\ \Theta(n^{\log_b \ell}) & \text{if } \ell > b^k \end{cases} \qquad (4.33)$$

Problem 4.44 gives a generalization of this example. □

Remark: It often happens in the analysis of algorithms that we derive a recurrence in the form of an *in*equality. For instance, we may get

$$T(n) \le \ell T(n/b) + cn^k \qquad n > n_0$$

when n/n_0 is an exact power of b, instead of Equation 4.29. What can we say about the asymptotic behaviour of such a recurrence? First note that we do not have enough information to determine the *exact* order of $T(n)$ because we are given

only an upper bound on its value. (For all we know, it could be that $T(n) = 1$ for all n.) The best we can do in this case is to analyse the recurrence in terms of the O notation. For this we introduce an auxiliary recurrence patterned after the original but defined in terms of an equation (not an inequation). In this case

$$\hat{T}(n) = \begin{cases} T(n_0) & \text{if } n = n_0 \\ \ell \hat{T}(n/b) + cn^k & \text{if } n/n_0 \text{ is a power of } b,\ n > n_0. \end{cases}$$

This new recurrence falls under the scope of Example 4.7.13, except that we have no evidence that $\hat{T}(n)$ is eventually nondecreasing. Therefore Equation 4.33 holds for $\hat{T}(n)$, provided we use conditional asymptotic notation to restrict n/n_0 to being a power of b. Now, it is easy to prove by mathematical induction that $T(n) \le \hat{T}(n)$ for all $n \ge n_0$ such that n/n_0 is a power of b. But clearly if

$$f(n) \in \Theta(t(n) \mid P(n))$$

and $g(n) \le f(n)$ for all n such that $P(n)$ holds, then $g(n) \in O(t(n) \mid P(n))$. Therefore, our conclusion about the conditional asymptotic behaviour of $\hat{T}(n)$ holds for $T(n)$ as well, provided we replace Θ by O. Finally, whenever we know that $T(n)$ is eventually nondecreasing, we can invoke the smoothness of the functions involved to conclude that Equation 4.33 holds unconditionally for $T(n)$, provided again we replace Θ by O. The solution of our recurrence is thus

$$T(n) \in \begin{cases} O(n^k) & \text{if } \ell < b^k \\ O(n^k \log n) & \text{if } \ell = b^k \\ O(n^{\log_b \ell}) & \text{if } \ell > b^k \end{cases}$$

We shall study further recurrences involving inequalities in Section 4.7.6.

So far, the changes of variable we have used have all been of the same logarithmic nature. Rather different changes of variable are sometimes useful. We illustrate this with one example that comes from the analysis of the divide-and-conquer algorithm for multiplying large integers (see Section 7.1).

Example 4.7.14. Consider an eventually nondecreasing function $T(n)$ such that

$$T(n) \le T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + T(1 + \lceil n/2 \rceil) + cn \qquad (4.34)$$

for all sufficiently large n, where c is some positive real constant. As explained in the remark following the previous example, we have to be content here to analyse the recurrence in terms of the O notation rather than the Θ notation.

Let $n_0 \ge 1$ be large enough that $T(m) \ge T(n)$ for all $m \ge n \ge n_0/2$ and Equation 4.34 holds for all $n > n_0$. Consider any $n > n_0$. First observe that

$$\lfloor n/2 \rfloor \le \lceil n/2 \rceil < 1 + \lceil n/2 \rceil,$$

which implies that

$$T(\lfloor n/2 \rfloor) \le T(\lceil n/2 \rceil) \le T(1 + \lceil n/2 \rceil).$$

Therefore, Equation 4.34 gives rise to

$$T(n) \le 3T(1 + \lceil n/2 \rceil) + cn.$$

Now make a change of variable by introducing a new function $\hat{T}$ such that $\hat{T}(n) = T(n+2)$ for all n. Consider again any $n > n_0$.

$$\begin{aligned} \hat{T}(n) = T(n+2) &\le 3T(1 + \lceil (n+2)/2 \rceil) + c(n+2) \\ &\le 3T(2 + \lceil n/2 \rceil) + 2cn \qquad \text{(because } n+2 \le 2n) \\ &= 3\hat{T}(\lceil n/2 \rceil) + 2cn \end{aligned}$$

In particular,

$$\hat{T}(n) \le 3\hat{T}(n/2) + dn \qquad n > n_0$$

when n/n_0 is a power of 2, where $d = 2c$. This is a special case of the recurrence analysed in the remark following Example 4.7.13, with $\ell = 3$, $b = 2$ and $k = 1$. Since $\ell > b^k$, we obtain $\hat{T}(n) \in O(n^{\lg 3})$. Finally, we use one last time the fact that $T(n)$ is eventually nondecreasing: $T(n) \le T(n+2) = \hat{T}(n)$ for any sufficiently large n. Therefore any asymptotic upper bound on $\hat{T}(n)$ applies equally to $T(n)$, which concludes the proof that $T(n) \in O(n^{\lg 3})$. □

4.7.5 Range transformations

When we make a change of variable, we transform the domain of the recurrence. Instead, it may be useful to transform the range to obtain a recurrence in a form that we know how to solve. Both transformations can sometimes be used together. We give just one example of this approach.

Example 4.7.15. Consider the following recurrence, which defines $T(n)$ when n is a power of 2.

$$T(n) = \begin{cases} 1/3 & \text{if } n = 1 \\ n\,T^2(n/2) & \text{otherwise} \end{cases}$$

The first step is a change of variable: let t_i denote $T(2^i)$.

$$\begin{aligned} t_i = T(2^i) &= 2^i\, T^2(2^{i-1}) \\ &= 2^i\, t_{i-1}^2 \end{aligned}$$

At first glance, none of the techniques we have seen applies to this recurrence since it is not linear; furthermore the coefficient 2^i is not a constant. To transform the range, we create yet another recurrence by using u_i to denote $\lg t_i$.

$$\begin{aligned} u_i = \lg t_i &= i + 2 \lg t_{i-1} \\ &= i + 2u_{i-1} \end{aligned}$$

This time, once rewritten as

$$u_i - 2u_{i-1} = i,$$

the recurrence fits Equation 4.10. The characteristic polynomial is

$$(x-2)(x-1)^2$$

and thus all solutions are of the form

$$u_i = c_1 2^i + c_2 1^i + c_3\, i\, 1^i.$$

Substituting this solution into the recurrence for u_i yields

$$\begin{aligned} i &= u_i - 2u_{i-1} \\ &= c_1 2^i + c_2 + c_3\, i - 2(c_1 2^{i-1} + c_2 + c_3(i-1)) \\ &= (2c_3 - c_2) - c_3\, i \end{aligned}$$

and thus $c_3 = -1$ and $c_2 = 2c_3 = -2$. Therefore, the general solution for u_i, if the initial condition is not taken into account, is $u_i = c_1 2^i - i - 2$. This gives us the general solution for t_i and $T(n)$.

$$\begin{aligned} t_i &= 2^{u_i} = 2^{c_1 2^i - i - 2} \\ T(n) &= t_{\lg n} = 2^{c_1 n - \lg n - 2} = \frac{2^{c_1 n}}{4n} \end{aligned}$$

We use the initial condition $T(1) = 1/3$ to determine c_1: $T(1) = 2^{c_1}/4 = 1/3$ implies that $c_1 = \lg(4/3) = 2 - \lg 3$. The final solution is therefore

$$T(n) = \frac{2^{2n}}{4n\,3^n}.$$

□

4.7.6 Asymptotic recurrences

When recurrences arise in the analysis of algorithms, they are often not as "clean" as

$$S(n) = \begin{cases} a & \text{if } n = 1 \\ 4S(n \div 2) + bn & \text{if } n > 1 \end{cases} \tag{4.35}$$

for specific positive real constants a and b. Instead, we usually have to deal with something less precise, such as

$$T(n) = 4T(n \div 2) + f(n) \tag{4.36}$$

when n is sufficiently large, where all we know about $f(n)$ is that it is in the exact order of n, and we know nothing specific about the initial condition that defines $T(n)$ except that it is positive for all n. Such an equation is called an *asymptotic recurrence*. Fortunately, the asymptotic solution of a recurrence such as Equation 4.36 is virtually always identical to that of the simpler Equation 4.35. The general technique to solve an asymptotic recurrence is to "sandwich" the function it defines between two recurrences of the simpler type. When both simpler recurrences have