Lecture II
RECURRENCES

§1. Master Theorem

If $a > 0, b > 1$ are constants, we consider the **master recurrence**

$$T(n) = aT(n/b) + f(n) \tag{1}$$

where $f(n)$ is some function, which we may call the **driving function**. Evidently, this is the recurrence to solve if we manage to solve a problem of size n by breaking it up into a subproblems each of size n/b, and merging these a subsolutions in time $f(n)$. The recurrence was systematically studied by Bentley, Haken and Saxe [1]. Solving it requires a combination of domain and range transformation.

First apply a domain transformation by defining

$$t(k) := T(b^k) \quad \text{(for all } k\text{)}.$$

Hence

$$t(k) = a\,t(k-1) + f(b^k).$$

Next, transform the range by using the summation factor $1/a^k$. This defines the function $s(k)$:

$$s(k) := t(k)/a^k.$$

Now $s(k)$ satisfies a recurrence in standard form:

$$\begin{aligned} s(k) &= \frac{t(k)}{a^k} \\ &= \frac{t(k-1)}{a^{k-1}} + \frac{f(b^k)}{a^k} \\ &= s(k-1) + \frac{f(b^k)}{a^k} \end{aligned}$$

Telescoping, we get

$$s(k) = s(k) - s(0) = \sum_{i=1}^{k} \frac{f(b^i)}{a^i},$$

where we have chosen the boundary condition $s(0) = 0$. *Now, we cannot proceed any further without knowing the nature of the function f.*

The master theorem considers three possibilities for f. The three cases are based on comparing f with a function

$$w(n) = n^{\log_b a}$$

which we shall call the **watershed function**. Note that b is the *b*ase of this function. The easiest possibility is the "watershed" case (CASE (0)). This is when $f(n) = \Theta(w(n))$. The other two possibilities are where f grows polynomially slower or "polynomially faster" than the watershed function. The latter is in quotes because the actual condition is technically stronger than polynomially faster.

CASE (0) This is when $f(n)$ satisfies

$$f(n) = \Theta(w(n)). \tag{2}$$

Then $f(b^i) = \Theta(a^i)$ and $s(k) = \sum_{i=1}^{k} f(b^i)/a^i = \Theta(k)$.

CASE (-1) This is when $f(n)$ grows polynomially slower than the watershed function:

$$f(n) = \mathcal{O}(n^{-\epsilon} w(n)), \tag{3}$$

for some $\epsilon > 0$. Then $f(b^i) = \mathcal{O}(b^{i(\log_b a - \epsilon)})$. Let $f(b^i) = \mathcal{O}_1(a^i b^{-i\epsilon})$ (using the subscripting notation for $\mathcal{O}$). So $s(k) = \sum_{i=1}^k f(b^i)/a^i = \sum \mathcal{O}_1(b^{-i\epsilon}) = \mathcal{O}_2(1)$, since $b > 1$ implies $b^{-\epsilon} < 1$. Hence $s(k) = \Theta(1)$.

CASE $(+1)$ This is when $f(n)$ satisfies the **regularity condition**

$$af(n/b) \le cf(n) \tag{4}$$

for some $c < 1$. Expanding this,

$$\begin{aligned} f(n) &\ge \frac{a}{c} f\left(\frac{n}{b}\right) \\ &\ge \left(\frac{a}{c}\right)^i f\left(\frac{n}{b^i}\right), \qquad (i = 1, 2, \ldots) \\ &\ge \left(\frac{a}{c}\right)^{\lceil \log_b n \rceil} f(x), \qquad (x \le 1) \\ &= \Omega(n^{\epsilon + \log_b a}), \end{aligned}$$

where $\epsilon = -\log_b c > 0$. Thus the regularity condition implies that $f(n)$ grows polynomially faster than the watershed function,

$$f(n) = \Omega(n^{\epsilon} w(n)). \tag{5}$$

It follows from (4) that $f(b^{k-i}) \le (c/a)^i f(b^k)$. So

$$\begin{aligned} s(k) &= \sum_{i=1}^{k} f(b^i)/a^i \\ &= \sum_{i=0}^{k-1} f(b^{k-i})/a^{k-i} \\ &\le \sum_{i=0}^{k-1} (c/a)^i f(b^k)/a^{k-i} \\ &= f(b^k)/a^k \left(\sum_{i=0}^{k-1} c^{k-i} \right) \\ &= \mathcal{O}\left(\frac{f(b^k)}{a^k} \right), \end{aligned}$$

since $c < 1$. But clearly, $s(k) \ge f(b^k)/a^k$. Hence we have $s(k) = \Theta(f(b^k)/a^k)$.

Summarizing,

$$s(k) = \begin{cases} \Theta(1), & \text{CASE } (-1) \\ \Theta(k), & \text{CASE } (0) \\ \Theta(f(b^k)/a^k), & \text{CASE } (+1). \end{cases}$$

Back substituting,

$$t(k) = a^k s(k) = \begin{cases} \Theta(a^k), & \text{CASE } (-1) \\ \Theta(a^k k), & \text{CASE } (0) \\ \Theta(f(b^k)), & \text{CASE } (+1). \end{cases}$$

Since $T(n) = t(\log_b n)$, we conclude:

THEOREM 1 (MASTER THEOREM) *The master recurrence (1) has solution:*

$$T(n) = \begin{cases} \Theta(w(n)), & \text{if } f(n) = \mathcal{O}(n^{-\epsilon} w(n)), \text{ for some } \epsilon > 0, \\ \Theta(w(n) \log n), & \text{if } f(n) = \Theta(w(n)), \\ \Theta(f(n)), & \text{if } af(n/b) \le cf(n) \text{ for some } c < 1. \end{cases}$$

Remarks on the regularity condition. Regularity condition is a kind of "smoothness" constraint on the function $f(n)$. Why does the regularity condition arise in case (+), but there is no analogue in case (-)? The reason is that in case (-), it is the watershed function $w(n)$ that dominates the overall complexity, and the regularity of the function $f(n)$ does not matter. In applications of the Master Theorem, before we test for the regulurity condition (4), it is often best to verify equation (5) first (this is usually easily done, even mentally). Although this is redundant, this is a useful "rejection test" because if equation (5) fails, then we know the regularity condition fails. But if equation (5) holds, then we know that either the Master Theorem is inapplicable, or case (+) of the Master Theorem holds. See Exercise for more on this point.

Limitations of the Master Theorem. The Master Theorem is powerful but unfortunately, there are gaps between its 3 cases. For instance, $f(n) = w(n)\log n$ grows faster than the watershed function, but not polynomially faster. Thus the Master Theorem is inapplicable for this $f(n)$. Yet it is just as easy to solve this case using the transformation techniques (see Exercise).

Note that the values a, b in the theorem are constants. Thus, attempting to apply this theorem to the recurrence

$$T(n) = 2^n T(n/2) + n^n$$

(with $a = 2^n$ and $b = 2$) leads to the false conclusion that $T(n) = \Theta(n^n \log n)$. See exercise. For a more general solution to the master recurrence, see [2].

Graphic Interpretation of the Master Recurrence. We imagine a "recursion tree" with branching factor of a at each node, and every leaf of the tree is at level $\log_b a$. Of course, this "tree" is not realizable unless a and $\log_b a$ are integers! We further associate a "size" of n/b^i and "cost" of $f(n/b^i)$ to each node at level i (root is at level $i = 0$). Then $T(n)$ is just the sum of the costs at all the nodes. The Master Theorem says this: In case (0), the total cost associated with nodes at any level is $\Theta(w(n))$ and there are $\log_b n$ levels giving an overall cost of $\Theta(w(n)\log n)$. In case (+1), the cost associated with the root is $\Theta(T(n))$. In case (-1), the total cost associated with the leaves is $\Theta(T(n))$.

EXERCISES

Exercise 1.1: State the solution, up to Θ-order of the following recurrences:

$$\begin{aligned} T(n) &= 10T(n/10) + \log^{10} n. \\ T(n) &= 100T(n/10) + n^{10}. \\ T(n) &= 10T(n/100) + (\log n)^{\log\log n}. \\ T(n) &= 16T(n/4) + 4^{\lg n}. \end{aligned}$$

◇

Exercise 1.2: Solve the following using the Master's theorem whenever possible. If the Master's theorem is inapplicable, say so (or, you can solve it by other means).

$$\begin{aligned} T(n) &= 3T(n/25) + \log^3 n. \\ T(n) &= 25T(n/3) + (n/\log n)^3. \\ T(n) &= T(\sqrt{n}) + n. \end{aligned}$$

HINT: in the third problem, the Master theorem is applicable after a simple transformation. ◇

Exercise 1.3: Solve the master recurrence when $f(n) = w(n)\log^k n$, for any k. NOTE: the Master Theorem is not applicable here, but the transformation method is applicable. ♢

Exercise 1.4: Re-prove the master theorem, but now apply the range transformation to the master recurrence before applying the domain transformation. ♢

Exercise 1.5: Show that the master theorem applies to the following variation of the master recurrence:

$$T(n) = a \cdot T(\frac{n+c}{b}) + f(n)$$

where $a > 0, b > 1$ and c is arbitrary. ♢

Exercise 1.6:
(a) Solve $T(n) = 2^n T(n/2) + n^n$ by direct expansion.
(b) Try to generalize the Master theorem to handle some cases of $T(n) = a_n T(n/b_n) + f(n)$ where a_n, b_n are both functions of n. ♢

End Exercises

References

[1] J. L. Bentley, D. Haken, and J. B. Saxe. A general method for solving divide-and-conquer recurrences. *ACM SIGACT News*, 12(3):36–44, 1980.

[2] X. Wang and Q. Fu. A frame for general divide-and-conquer recurrences. *Info. Processing Letters*, 59:45–51, 1996.