CMPS 101
Algorithms and Abstract Data Types

Recurrence Relations

Iteration Method
Recall the following example from the induction handout.

$$T(n) = \begin{cases} 0 & n = 1 \\ T(\lfloor n/2 \rfloor) + 1 & n \geq 2 \end{cases}$$

We begin by illustrating a solution technique called *Iteration*, which consists of repeatedly substituting the recurrence into itself until a pattern emerges.

$$\begin{aligned} T(n) &= 1 + T(\lfloor n/2 \rfloor) \\ &= 1 + 1 + T(\lfloor \lfloor n/2 \rfloor / 2 \rfloor) = 2 + T(\lfloor n/2^2 \rfloor) \\ &= 2 + 1 + T(\lfloor \lfloor n/2^2 \rfloor / 2 \rfloor) = 3 + T(\lfloor n/2^3 \rfloor) \\ &\vdots \\ &= k + T(\lfloor n/2^k \rfloor) \end{aligned}$$

This process must terminate when the recursion depth k is at its maximum, i.e. when $\lfloor n/2^k \rfloor = 1$. To solve this equation for k in terms of n, we use the inequality definition of the floor function.

$$\begin{aligned} & 1 \leq n/2^k < 2 \\ \therefore \quad & 2^k \leq n < 2^{k+1} \\ \therefore \quad & k \leq \lg(n) < k+1 \\ \therefore \quad & k = \lfloor \lg(n) \rfloor \end{aligned}$$

Thus for the recursion depth $k = \lfloor \lg(n) \rfloor$ we have $T(\lfloor n/2^k \rfloor) = T(1) = 0$, and hence the solution to the above recurrence is $T(n) = \lfloor \lg(n) \rfloor$. It follows that $T(n) = \Theta(\log(n))$.

Exercise
Check directly that $T(n) = \lfloor \lg(n) \rfloor$ is the solution to the above recurrence relation, i.e. check that $T(1) = 0$, and for any $n \geq 2$, that $T(n) = 1 + T(\lfloor n/2 \rfloor)$.

Exercise
Use this same technique to show that the recurrence

$$S(n) = \begin{cases} 0 & n = 1 \\ S(\lceil n/2 \rceil) + 1 & n \geq 2 \end{cases}$$

has solution $S(n) = \lceil \lg(n) \rceil$, and hence also $S(n) = \Theta(\log(n))$.

Comparing the solutions to the preceding examples, we see that replacing floor $\lfloor\ \rfloor$ by ceiling $\lceil\ \rceil$ has no affect on the *asymptotic solution* $\Theta(\log(n))$, while the *exact solutions* are different: $\lfloor \lg(n) \rfloor$ vs. $\lceil \lg(n) \rceil$. We can change other details in a recurrence without changing the asymptotic solution. The following recurrence satisfies $T(n) = \Theta(\log(n))$ for any values of the constants c, d, and n_0.

$$T(n) = \begin{cases} c & 1 \le n < n_0 \\ T(\lfloor n/2 \rfloor) + d & n \ge n_0 \end{cases}$$

Exercise Use the iteration method to determine the exact solution to this recurrence when $c = 3$, $d = 5$, and $n_0 = 10$. (Answer: $T(n) = 5\lfloor \lg(n/10) \rfloor + 8$ for $n \ge 10$, whence $T(n) = \Theta(\log(n))$.)

Often it is difficult or impossible to determine an exact solution via the iteration method, while it *is* possible to obtain an asymptotic solution. Consider

$$T(n) = \begin{cases} 1 & n = 1 \\ T(\lfloor n/2 \rfloor) + n^2 & n \ge 2 \end{cases}$$

Upon iterating this recurrence we find $T(n) = \sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor^2 + 1$, where $k = \lfloor \lg n \rfloor$. We can use this expression to show that $T(n) = \Theta(n^2)$.

$$\begin{aligned}
T(n) &= \sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor^2 + 1 \\
&\le n^2 \left(\sum_{i=0}^{k-1} (1/4)^i \right) + 1 && \text{(since } \lfloor x \rfloor \le x\text{)} \\
&= n^2 \left(\frac{1-(1/4)^k}{1-(1/4)} \right) + 1 && \text{(from a well known summation formula)} \\
&= \frac{4}{3} n^2 \left(1 - \frac{1}{4^{\lfloor \lg n \rfloor}} \right) + 1 \\
&\le \frac{4}{3} n^2 \left(1 - \frac{1}{4^{\lg n}} \right) + 1 && \text{(since } \lfloor \lg n \rfloor \le \lg n\text{)} \\
&= \frac{4}{3} n^2 \left(1 - \frac{1}{n^2} \right) + 1 && \text{(since } 4^{\lg n} = n^{\lg 4} = n^2\text{)} \\
&= \frac{4}{3} n^2 + \frac{7}{3} \\
&= O(n^2)
\end{aligned}$$

Therefore $T(n) = O(n^2)$. We leave it as an exercise to show in a similar manner that $T(n) = \Omega(n^2)$, whence $T(n) = \Theta(n^2)$.

The Master Method

This is a method for finding (asymptotic) solutions to recurrences of the form

$$T(n) = aT(n/b) + f(n)$$

where $a \geq 1$, $b > 1$, and the function $f(n)$ is asymptotically positive. Here $T(n/b)$ denotes either $T(\lfloor n/b \rfloor)$ or $T(\lceil n/b \rceil)$, and it is understood that $T(n) = \Theta(1)$ for some finite set of initial terms. Such a recurrence describes the run time of a 'divide and conquer' algorithm which divides a problem of size *n* into *a* subproblems, each of size n/b. In this context $f(n)$ represents the cost of doing the dividing and re-combining.

<u>Master Theorem</u>

Let $a \geq 1$, $b > 1$, $f(n)$ be asymptotically positive, and let $T(n)$ be defined by $T(n) = aT(n/b) + f(n)$. Then we have three cases:

(1) If $f(n) = O\left(n^{\log_b(a)-\varepsilon}\right)$ for some $\varepsilon > 0$, then $T(n) = \Theta\left(n^{\log_b(a)}\right)$.

(2) If $f(n) = \Theta\left(n^{\log_b(a)}\right)$, then $T(n) = \Theta\left(n^{\log_b(a)} \cdot \log(n)\right)$.

(3) If $f(n) = \Omega\left(n^{\log_b(a)+\varepsilon}\right)$ for some $\varepsilon > 0$, and if $af(n/b) \leq cf(n)$ for some $0 < c < 1$ and for all sufficiently large *n*, then $T(n) = \Theta(f(n))$.

Remarks In each case we compare $f(n)$ to the polynomial $n^{\log_b(a)}$, and the solution is determined by which function is of an asymptotically higher order. In case (1) $n^{\log_b(a)}$ is polynomially larger than $f(n)$ and the solution is in the class $\Theta\left(n^{\log_b(a)}\right)$. In case (3) $f(n)$ is polynomially larger (and an additional *regularity condition* is met) so the solution is $\Theta(f(n))$. In case (2) the functions are asymptotically equivalent and the solution is in the class $\Theta\left(n^{\log_b(a)} \cdot \log(n)\right)$, which is the same as $\Theta(f(n) \cdot \log(n))$. To say that $n^{\log_b(a)}$ is *polynomially larger* than $f(n)$ as in (1), means that $f(n)$ is bounded above by a function which is smaller than $n^{\log_b(a)}$ by a polynomial factor, namely n^{ε} for some $\varepsilon > 0$. Note that the conclusion reached by the master theorem does not change if we replace $f(n)$ by a function asymptotically equivalent to it. For this reason the recurrence may simply be given as $T(n) = aT(n/b) + \Theta(f(n))$. Notice also that there is no mention of initial terms. It is part of the content of the master theorem that the initial values of the recurrence do not effect it's asymptotic solution.

Examples

Let $T(n) = 8T(n/2) + n^3$ so that $a = 8$, $b = 2$, and $\log_b(a) = 3$. Hence $f(n) = n^3 = \Theta\left(n^{\log_b(a)}\right)$, so we are in case (2). Therefore $T(n) = \Theta(n^3 \log(n))$.

Now let $T(n) = 5T(n/4) + n$. Here $a = 5$, $b = 4$, $\log_b(a) = 1.609... > 1$. Let $\varepsilon = \log_4(5) - 1$ so that $\varepsilon > 0$, and $f(n) = n = O\left(n^{\log_4(5)-\varepsilon}\right)$. Therefore we are in case (1) and $T(n) = \Theta\left(n^{\log_4(5)}\right)$.

Next consider $T(n) = 5T(n/4) + n^2$. Again $a = 5$, $b = 4$, so $\log_b(a) = 1.609... < 2$. Let $\varepsilon = 2 - \log_4(5)$, so that $\varepsilon > 0$, and $f(n) = n^2 = \Omega\left(n^{\log_4(5)+\varepsilon}\right)$. We appear to be in case (3), but we must still check the regularity condition: $5f(n/4) \leq cf(n)$ for some $0 < c < 1$ and all sufficiently large *n*. This inequality says

$5(n/4)^2 \le cn^2$, i.e. $(5/16)n^2 \le cn^2$, which is true as long as c is chosen to satisfy $5/16 \le c < 1$. By case (3) of the Master Theorem $T(n) = \Theta(n^2)$.

Note that in applying the Master Theorem, we can always replace $f(n)$ by some simpler function that is asymptotically equivalent to it. This is part of the content of the theorem since the hypothesis in each case refers only to the asymptotic growth rate of $f(n)$. So for instance, if we were to replace n^3 in the first example above with $f(n) = 10n^3 + 15n^2 - n^{1.5} + n\log(n) + 1$, the analysis would be in no way different, and the recurrence would have the very same asymptotic solution. (Of course the exact solution to the recurrence would be very different.)

Observe also that in the three preceding examples, $f(n)$ was a polynomial. This is a particularly easy setting to apply the Master Theorem, since to establish which case to use, one merely compares $\deg(f)$ to the number $\log_b(a)$. If they are the same, case (2) applies. If $\log_b(a)$ is larger, case (1) applies by setting $\varepsilon = \log_b(a) - \deg(f)$. If $\deg(f)$ is larger, case(3) applies by setting $\varepsilon = \deg(f) - \log_b(a)$, and the regularity condition is easily checked. (**Exercise**: prove that if $f(n)$ is a polynomial, and if $\deg(f) > \log_b(a)$, then the regularity condition necessarily holds.)

Checking the hypotheses can be a little more complicated if $f(n)$ is not a polynomial. For example consider $T(n) = T(\lfloor n/2 \rfloor) + 2T(\lceil n/2 \rceil) + \log(n!)$. We first re-write this as $T(n) = 3T(n/2) + n\log(n)$. Upon letting $\varepsilon = \dfrac{1}{2}(\log_2(3) - 1)$, we have $\varepsilon > 0$, and $1 + \varepsilon = \log_2(3) - \varepsilon$. One checks easily that $n\log(n) = o(n^{1+\varepsilon})$, whence $n\log(n) = O(n^{1+\varepsilon}) = O(n^{\log_2(3)-\varepsilon})$. Case (1) now gives $T(n) = \Theta(n^{\log_2(3)})$.

In spite of the name "Master Theorem" the three cases do not cover all possibilities. There is a gap between cases (1) and (2) when $n^{\log_b(a)}$ is larger than $f(n)$, but not polynomially larger. Take for example the recurrence $T(n) = 2T(n/2) + n/\log(n)$. Observe that $n/\log(n) = \omega(n^{1-\varepsilon})$ for any $\varepsilon > 0$, whence $n/\log(n) \ne O\left(n^{1-\varepsilon}\right)$, and therefore we are not in case (1). But also $n/\log(n) = o(n)$, so that $n/\log(n) \ne \Theta(n)$, and neither are we in case (2). Thus the Master Theorem cannot be applied to this recurrence. A similar gap exists between cases (2) and (3). It is also possible that the regularity condition in case (3) fails, even though $f(n) = \Omega\left(n^{\log_b(a)+\varepsilon}\right)$ for some $\varepsilon > 0$.