CS350 ABET Objective 2

Prove and apply the Master Theorem

Textbook Appendix B, ~5%.

This is the proof.
Applications are in Objective 10-1.

Div and conq recurrence

A divide-and-conquer algorithm solves a problem by dividing its given instance into several smaller instances.

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

Assume:

$$a \geqslant 1$$
$$b \geqslant 2$$
$$n = b^k$$

By substitution:

$$T(b^k) = a^k \left(T(1) + \sum_{j=1}^{k} f(b^j)/a^j\right)$$

Since $a^k = a^{\log_b n} = n^{\log_b a}$

$$T(n) = n^{\log_b a} \left(T(1) + \sum_{j=1}^{\log_b n} f(b^j)/a^j \right)$$

Needs two new concepts.

eventually non-decreasing:

$$f(n_1) \leqslant f(n_2) \qquad \text{for all } n_2 > n_1 \geqslant n_0$$

smooth: eventually non-decreasing and

$$f(2n) \in \Theta(f(n))$$

e.g.: n, $n \log n$, n^2, n^3, $\ldots$ are smooth,
whereas a^n is not.

Theorem: If f is a smooth function,

$$f(bn) \in \Theta(f(n))$$

Proof: do independently the O and Ω cases.
Must prove exists c and n_0 such that

$$f(bn) \leqslant c f(n), \quad for \;\; n \geqslant n_0$$

Prove that for all $k = 1,2,\ldots$

$$f(2^k n) \leqslant c_2^k f(n), \quad \text{for} \;\; n \geqslant n_0 \text{ and some } c_2 > 0$$

by induction on k .

For any $b \geqslant 2$ there exists a k such that $2^{k-1} \leqslant b \leqslant 2^k$. Thus,

$$f(bn) \leqslant f(2^k n) \leqslant c_2^k f(n), \quad for \;\; n \geqslant n_0$$

Theorem: Let T eventually non-decreasing and f smooth. If $T(n) \in \Theta(f(n))$, for n power of $b \geqslant 2$, then $T(n) \in \Theta(f(n))$.

Proof: do independently the O and Ω cases. By the assumptions:

$$T(b^k) \leqslant c f(b^k) \quad \text{for}\, b^k \geqslant n_0$$

and by previous Theorem $f(bn) \leqslant c_b f(n)$.

Let $n_0 \leqslant b^k \leqslant n \leqslant b^{k+1}$

$$\begin{aligned} T(n) &\leqslant T(b^{k+1}) \leqslant c f(b^{k+1}) \\ &= c f(b\, b^k) \leqslant c\, c_b f(b^k) \leqslant c\, c_b f(n) \end{aligned}$$

Master Theorem: Let $a \geqslant 1, b \geqslant 2, c > 0,$ and

$$T(n) = aT(n/b) + f(n) \quad \text{for } n = b^k, k = 1,2,\ldots$$
$$T(1) = c$$

if $f(n) \in \Theta(n^d)$ and $d \geqslant 0$, then:

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d, \\ \Theta(n^d \log n) & \text{if } a = b^d, \\ \Theta(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$

Same for O and Ω .

Proof: Simplify $f(n) = n^d$. Have seen earlier:

$$T(n) = n^{\log_b a} \left(T(1) + \sum_{j=1}^{\log_b n} (b^d/a)^j\right)$$

Work on $\sum$:

If $b^d = a$, $\sum = \log_b n$ then $T(n) \in \Theta(n^d \log_b n)$

If $b^d \neq a$, $\sum = x \dfrac{x^{\log_b n} - 1}{x - 1}$ where $x = b^d/a$

if $a < b^d$, i.e., $b^d / a > 1$, $\sum \in \Theta\left(\left(b^d / a\right)^{\log_b n}\right)$

hence $T(n) \in n^{\log_b a} \Theta\left(\left(b^d / a\right)^{\log_b n}\right) = \ldots = \Theta\left(n^d\right)$

if $a > b^d$, i.e., $b^d / a < 1$, $\sum \in \Theta(1)$

hence $T(n) \in \Theta\left(n^{\log_b a}\right)$

References

Textbook Appendix B

Web:

http://en.wikipedia.org/wiki/Master_theorem