Proof of the Master Theorem

Theorem 1. ***[Master Theorem]*** *Let $a \geq 1$ and $b > 1$ be constants, and let $f(n)$ be an asymptotically positive function. Let $T(n)$ be defined, for integers $n > 0$, as*

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

Then:

1. *If $f(n) = O(n^{(\log_b a)-\epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.*
2. *If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.*
3. *If $f(n) = \Omega(n^{(\log_b a)+\epsilon})$ for some constant $\epsilon > 0$ and $af(n/b) \leq cf(n)$ for some constant $c < 1$ and sufficiently large n, then $T(n) = \Theta(f(n))$.*

For now we will assume that n is a power of b, but this is not, in general, required. Our proof will make use of two lemmas.

Lemma 1. *Let $T(n)$ be defined as follows for integers $n > 0$:*

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ aT(n/b) + f(n) & n = b^i,\ (i \text{ integer},\ i > 0) \end{cases}.$$

Then

$$T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{(\log_b n)-1} a^j f(n/b^j).$$

Lemma 2. *Let*

$$g(n) = \sum_{j=0}^{(\log_b n)-1} a^j f(n/b^j).$$

Then $g(n)$ has the following asymptotic bounds:

1. *If $f(n) = O(n^{(\log_b a)-\epsilon})$ for some constant $\epsilon > 0$, then $g(n) = O(n^{\log_b a})$.*
2. *If $f(n) = \Theta(n^{\log_b a})$, then $g(n) = \Theta(n^{\log_b a} \lg n)$.*
3. *If $af(n/b) \leq cf(n)$ for some constant $c < 1$ and sufficiently large n, then $g(n) = \Theta(f(n))$.*

Proof. **[Master Theorem]**
By Lemma 1, we have:

Case 1: $f(n) = O(n^{(\log_b a)-\epsilon})$ for some constant $\epsilon > 0$. Then by Lemma 2, we have

$$g(n) = \qquad .$$

So we conclude that

$$T(n) = \qquad .$$

Case 2: $f(n) = \Theta(n^{\log_b a})$. Then by Lemma 2, we have

$$g(n) = \qquad .$$

So we conclude that

$$T(n) = \qquad .$$

Case 3: $f(n) = \Omega(n^{(\log_b a)+\epsilon})$ for some constant $\epsilon > 0$, and $af(n/b) \leq cf(n)$ for some constant $c < 1$ and sufficiently large n. Then by Lemma 2, we have

$$g(n) = \qquad .$$

So we conclude that

$$T(n) = \qquad .$$

□

Proof. **[Lemma 1]** Sketch the recursion tree for the recurrence $T(n)$:

How many nodes are there at level j?

What is the cost of each node at level j?

What is the depth of the tree?

What is the total cost of all internal nodes (i.e., excluding leaves)?

How many leaves are there?

What is the cost of each leaf?

What is the total cost of the entire tree (internal nodes plus leaves)?

□

Proof. **[Lemma 2]**

Case 1: $f(n) = O(n^{(\log_b a)-\epsilon})$ for some constant $\epsilon > 0$.

What is the asymptotic class of $f(n/b^j)$?

Plug this result into $g(n)$ and simplify:

Case 2: $f(n) = \Theta(n^{\log_b a})$.

What is the asymptotic class of $f(n/b^j)$?

Plug this result into $g(n)$ and simplify:

Case 3: $af(n/b) \le cf(n)$ for some constant $c < 1$ and sufficiently large n.

Why can we say immediately that $g(n) = \Omega(f(n))$?

Next we need to show that $g(n) = O(f(n))$.

Rewrite the regularity condition $af(n/b) \le cf(n)$ by dividing both sides by a:

Iterate j times:

Multiply both sides by a^j:

Plug this result into $g(n)$ and simplify:

Combining these two results, we get that $g(n) = \Theta(f(n))$. $\square$