CSE 638: Advanced Algorithms

Supplemental Material
(The Master Theorem)

Rezaul A. Chowdhury

Department of Computer Science
SUNY Stony Brook

Spring 2013

A Useful Recurrence

Consider the following recurrence:

$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise; \end{cases}$$

where, $a \geq 1$ and $b > 1$.

Arises frequently in the analyses of *divide-and-conquer* algorithms.

Consider the following recurrences from my CSE548 (Fall'12) lectures.

Karatsuba's Algorithm: $T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$

Strassen's Algorithm: $T(n) = 7T\left(\frac{n}{2}\right) + \Theta(n^2)$

Fast Fourier Transform: $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$

How the Recurrence Unfolds

$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$

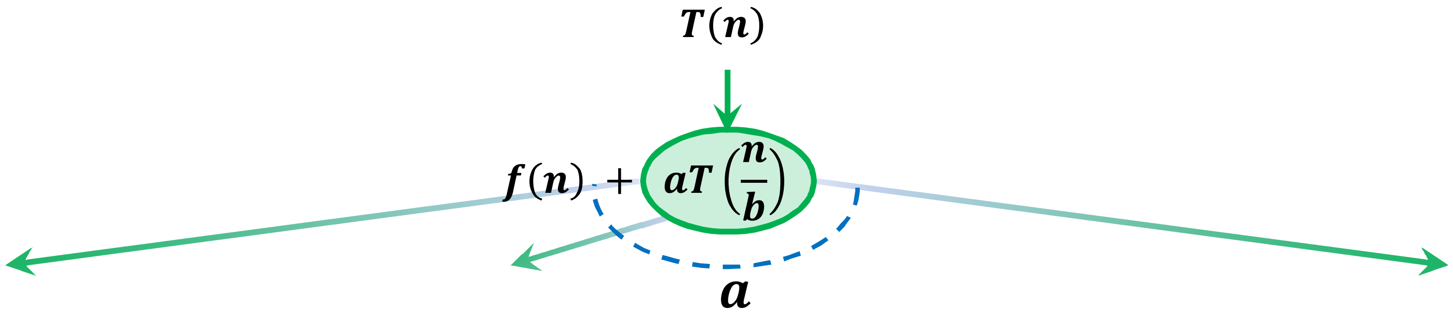
How the Recurrence Unfolds

$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$

$$\begin{tikzcd} & \boldsymbol{T(n)} \arrow[d] \\ \boldsymbol{f(n)} \ + & \boldsymbol{aT\left(\frac{n}{b}\right)} \end{tikzcd}$$

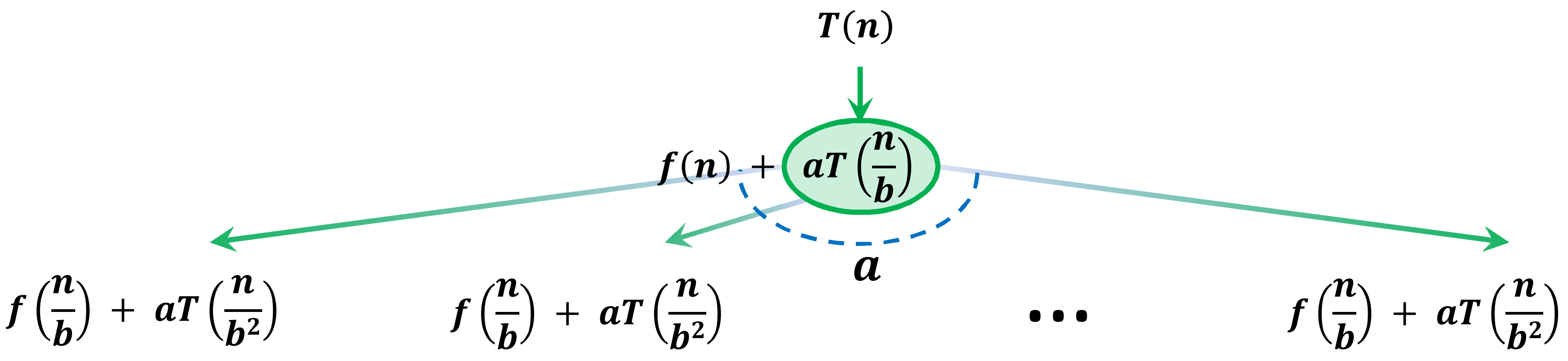
How the Recurrence Unfolds

$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$



How the Recurrence Unfolds

$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$



$\boldsymbol{T(n)}$

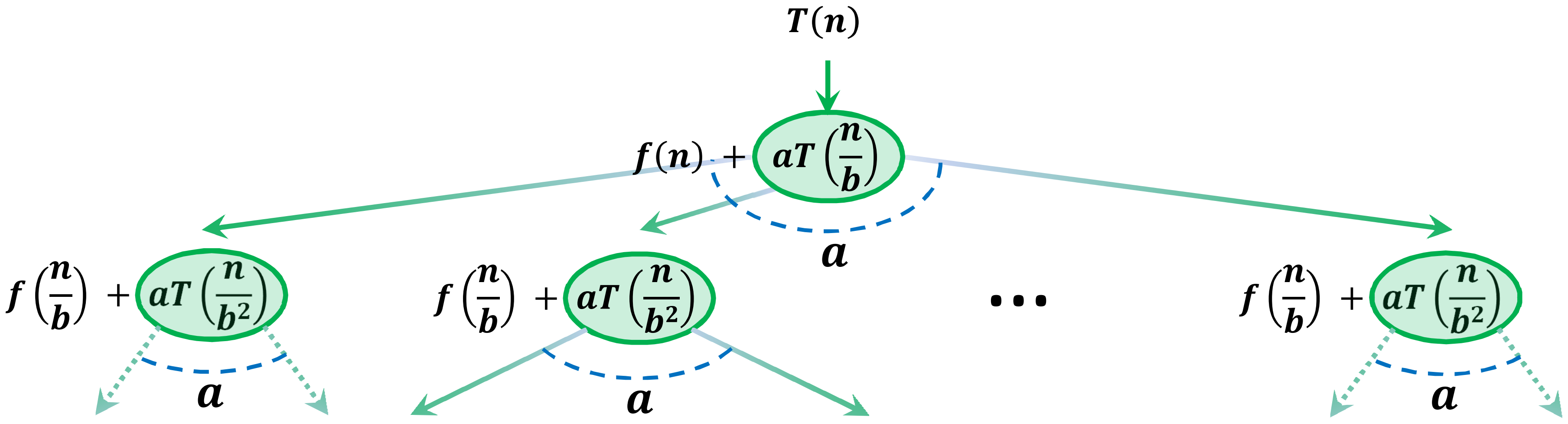
$\boldsymbol{f(n) + aT\left(\frac{n}{b}\right)}$

$\boldsymbol{a}$

$\boldsymbol{f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)}$ $\boldsymbol{f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)}$ … $\boldsymbol{f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)}$

How the Recurrence Unfolds

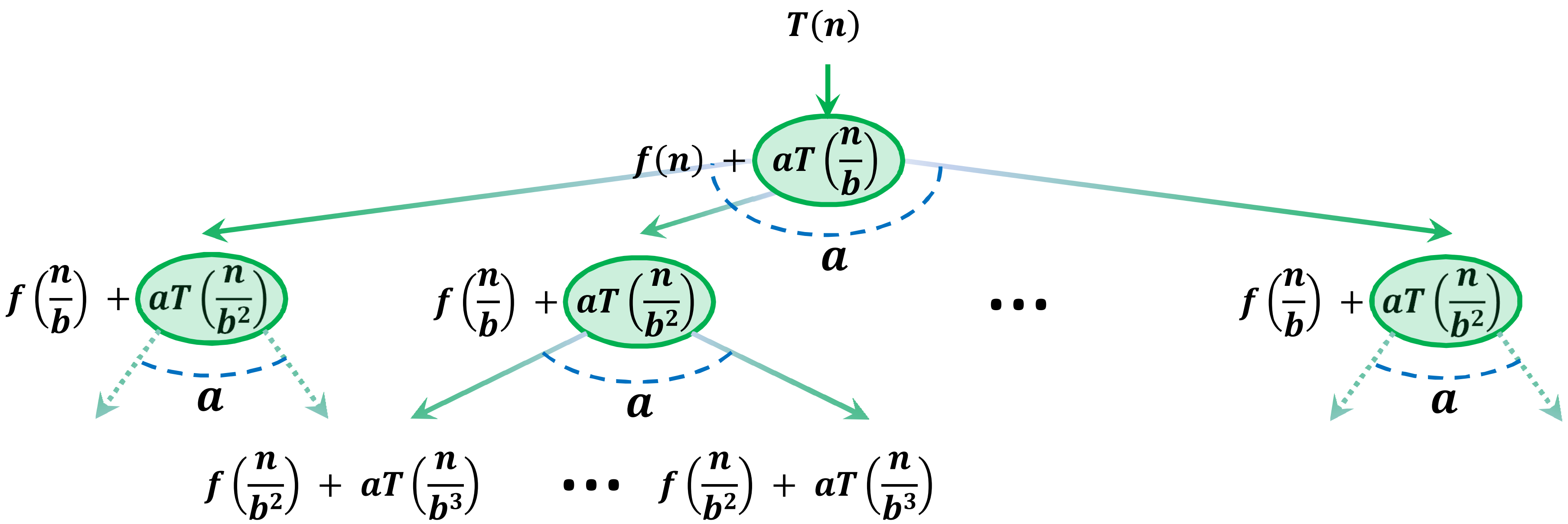
$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$



$T(n)$
$f(n) + aT\left(\frac{n}{b}\right)$
a
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
a
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
a
…
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
a

How the Recurrence Unfolds

$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$



$T(n)$

$f(n) + aT\left(\frac{n}{b}\right)$

a

$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$ a

$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$ a

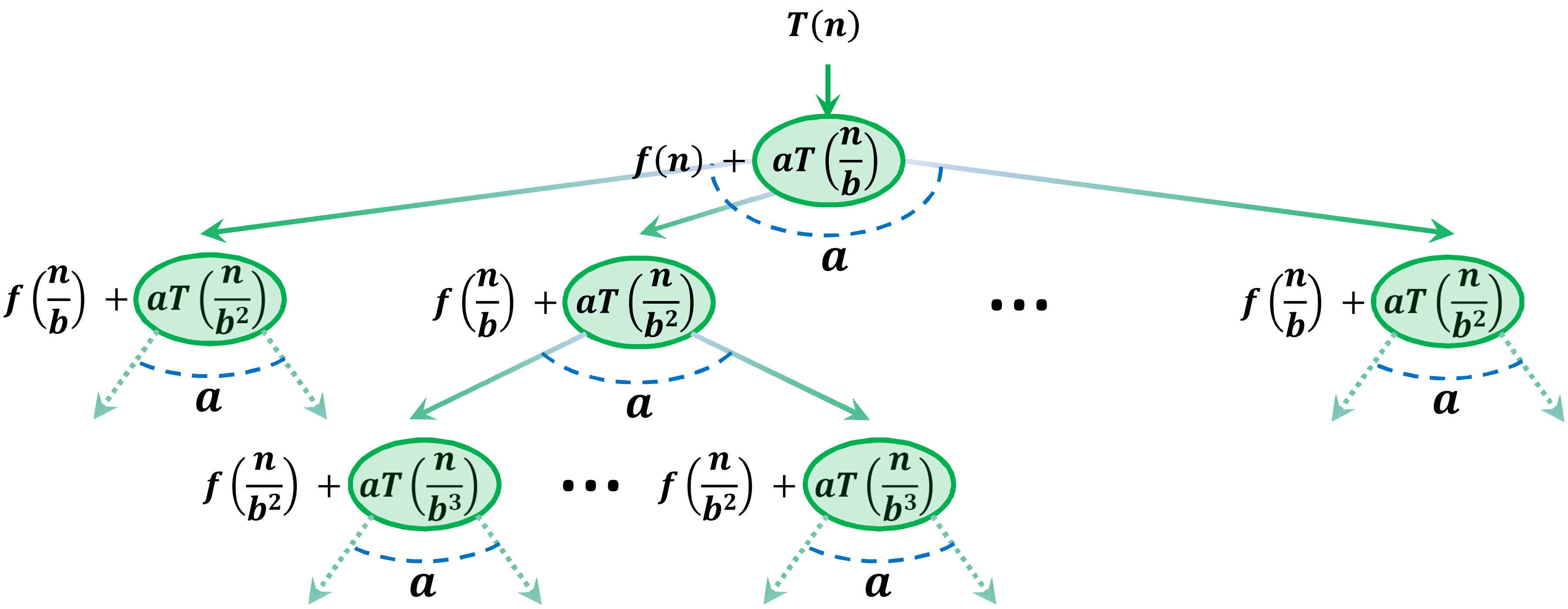
…

$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$ a

$f\left(\frac{n}{b^2}\right) + aT\left(\frac{n}{b^3}\right)$ … $f\left(\frac{n}{b^2}\right) + aT\left(\frac{n}{b^3}\right)$

How the Recurrence Unfolds

$$T(n) = \begin{cases} \Theta(1), & \quad if\ n \le 1, \\ aT\left(\frac{n}{b}\right) + f(n), & \quad otherwise. \end{cases}$$



$\boldsymbol{T(n)}$

$\boldsymbol{f(n)} + \boldsymbol{aT\left(\frac{n}{b}\right)}$

$\boldsymbol{a}$

$\boldsymbol{f\left(\frac{n}{b}\right)} + \boldsymbol{aT\left(\frac{n}{b^2}\right)}$ $\boldsymbol{f\left(\frac{n}{b}\right)} + \boldsymbol{aT\left(\frac{n}{b^2}\right)}$ $\cdots$ $\boldsymbol{f\left(\frac{n}{b}\right)} + \boldsymbol{aT\left(\frac{n}{b^2}\right)}$

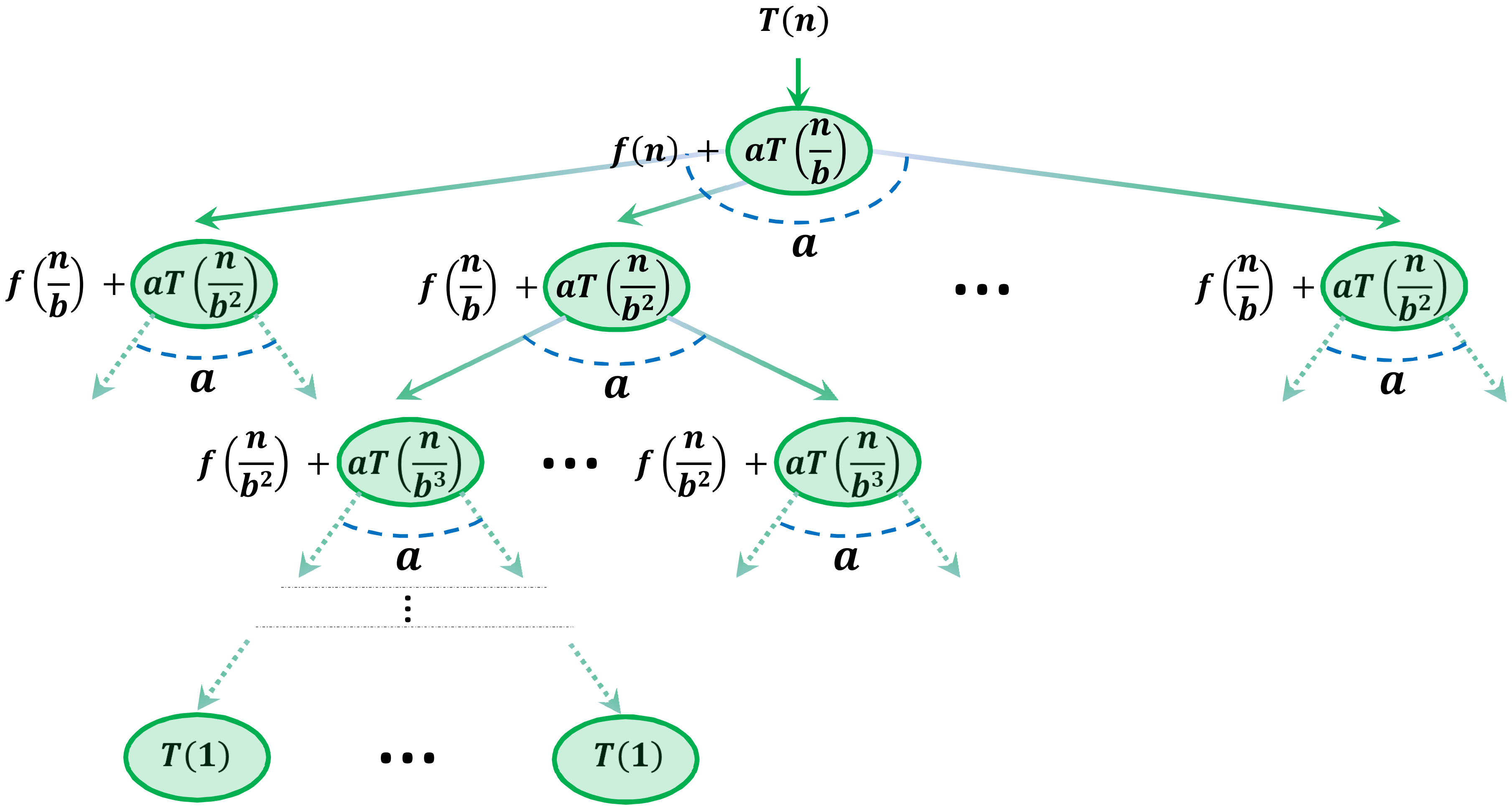
$\boldsymbol{a}$ $\boldsymbol{a}$ $\boldsymbol{a}$

$\boldsymbol{f\left(\frac{n}{b^2}\right)} + \boldsymbol{aT\left(\frac{n}{b^3}\right)}$ $\cdots$ $\boldsymbol{f\left(\frac{n}{b^2}\right)} + \boldsymbol{aT\left(\frac{n}{b^3}\right)}$

$\boldsymbol{a}$ $\boldsymbol{a}$

How the Recurrence Unfolds

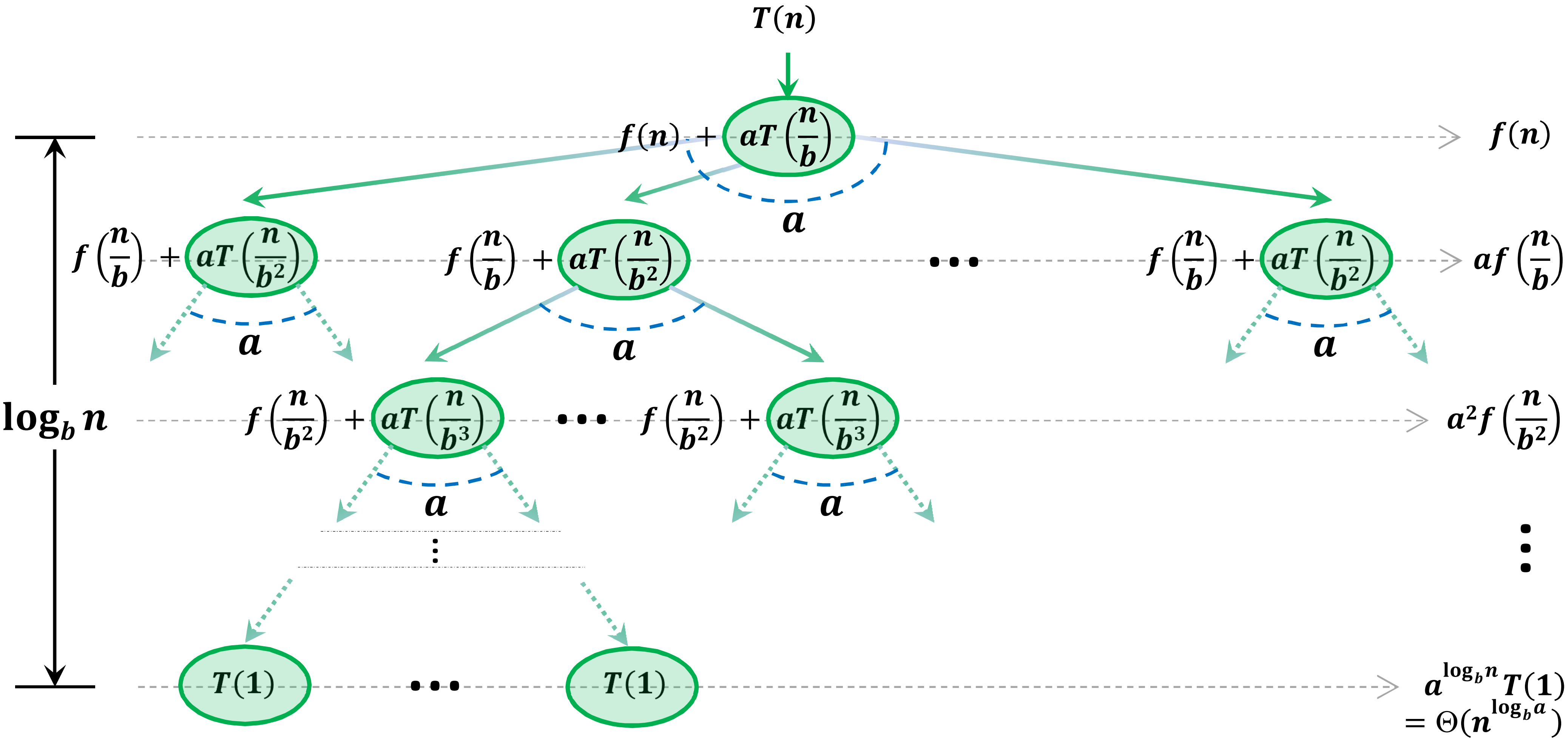
$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$



$T(n)$
$f(n) + aT\left(\frac{n}{b}\right)$
a
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
a
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
a
…
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
a
$f\left(\frac{n}{b^2}\right) + aT\left(\frac{n}{b^3}\right)$
a
…
$f\left(\frac{n}{b^2}\right) + aT\left(\frac{n}{b^3}\right)$
a
⋮
$T(1)$
…
$T(1)$

How the Recurrence Unfolds

$$T(n) = \begin{cases} \Theta(1), & \textit{if } n \le 1, \\ aT\left(\frac{n}{b}\right) + f(n), & \textit{otherwise.} \end{cases}$$



$T(n)$

$f(n) + aT\left(\frac{n}{b}\right)$ → $f(n)$

a

$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$ $f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$ ••• $f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$ → $af\left(\frac{n}{b}\right)$

a a a

$\log_b n$

$f\left(\frac{n}{b^2}\right) + aT\left(\frac{n}{b^3}\right)$ ••• $f\left(\frac{n}{b^2}\right) + aT\left(\frac{n}{b^3}\right)$ → $a^2 f\left(\frac{n}{b^2}\right)$

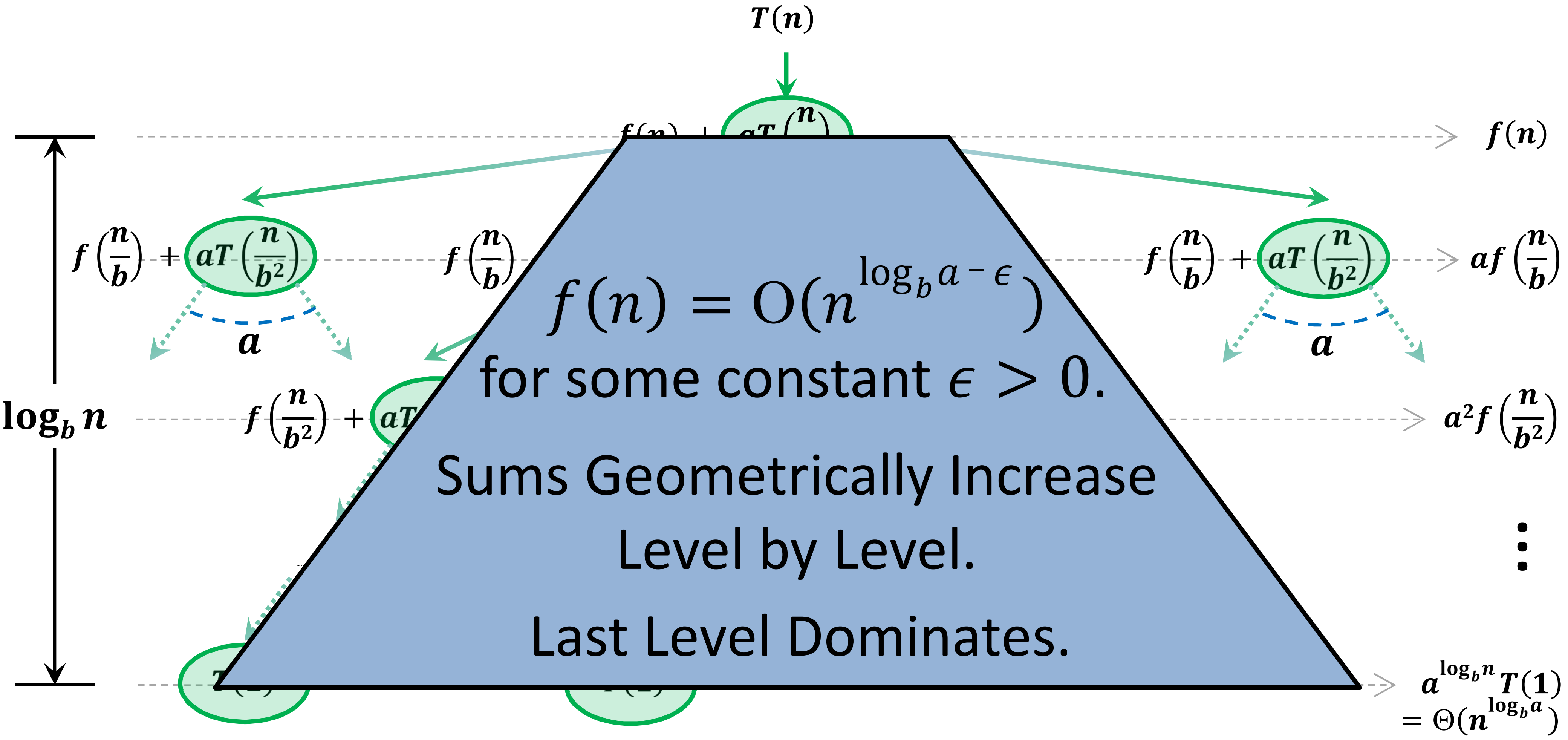
a a

⋮

$T(1)$ ••• $T(1)$ → $a^{\log_b n} T(1) = \Theta(n^{\log_b a})$

How the Recurrence Unfolds: Case 1

$$T(n) = \begin{cases} \Theta(1), & if\ n \le 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$



$T(n)$
$f(n) + aT\left(\frac{n}{b}\right)$
$f(n)$
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
$f\left(\frac{n}{b}\right)$
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
$af\left(\frac{n}{b}\right)$
a
a
$\log_b n$
$f\left(\frac{n}{b^2}\right) + aT$
$a^2 f\left(\frac{n}{b^2}\right)$
$T(1)$
$T(1)$
$a^{\log_b n} T(1)$
$= \Theta(n^{\log_b a})$

$f(n) = \mathrm{O}(n^{\log_b a - \epsilon})$

for some constant $\epsilon > 0$.

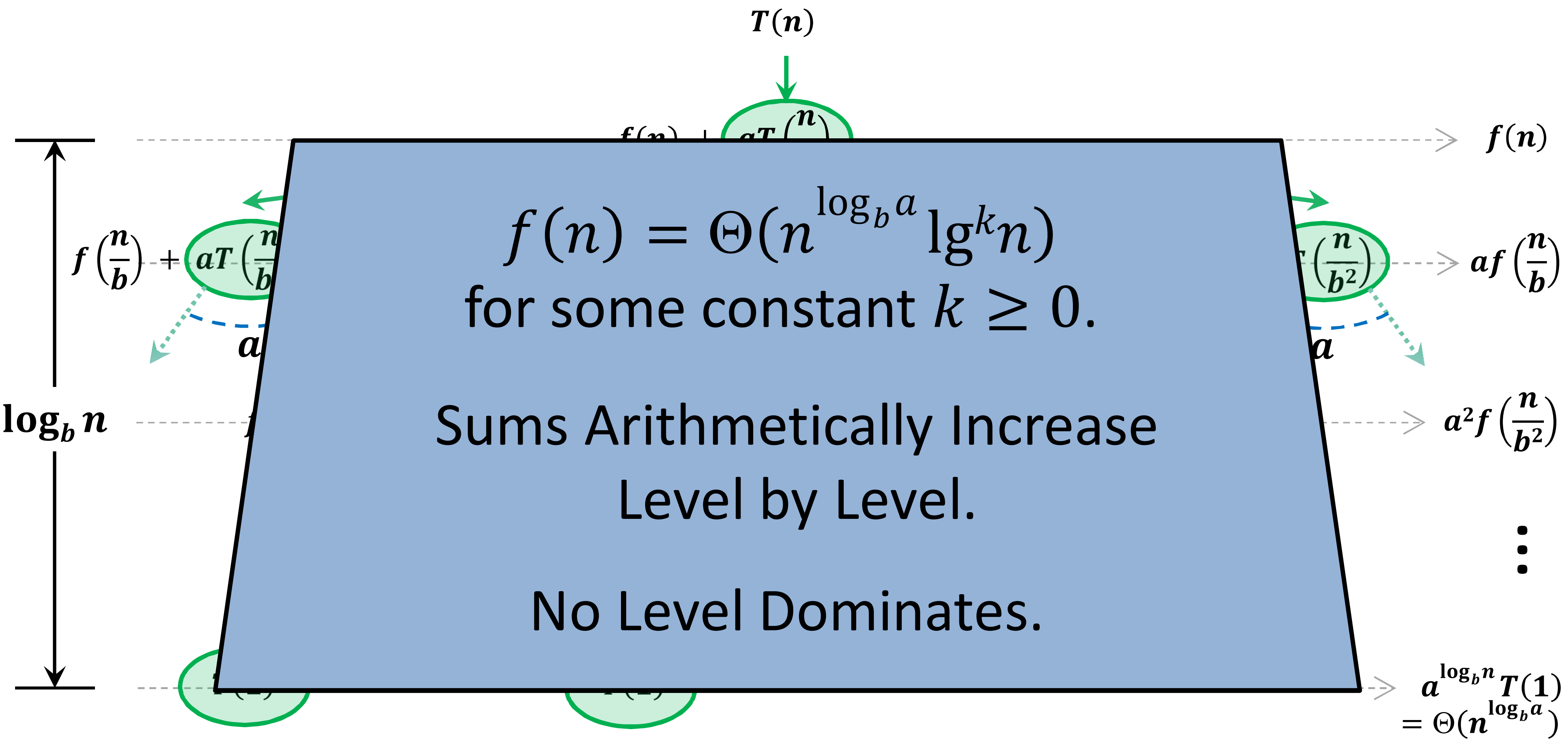
Sums Geometrically Increase Level by Level.

Last Level Dominates.

$$\boxed{T(n) = \Theta(n^{\log_b a})}$$

How the Recurrence Unfolds: Case 2

$$T(n) = \begin{cases} \Theta(1), & if\ n \le 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$



$\boldsymbol{T(n)}$
$\boldsymbol{f(n) + aT\left(\frac{n}{b}\right)}$
$\boldsymbol{f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)}$
$\boldsymbol{a}$
$\boldsymbol{a}$
$\boldsymbol{\log_b n}$
$\boldsymbol{T(1)}$
$\boldsymbol{f(n)}$
$\boldsymbol{af\left(\frac{n}{b}\right)}$
$\boldsymbol{a^2 f\left(\frac{n}{b^2}\right)}$
$\vdots$
$\boldsymbol{a^{\log_b n} T(1)}$
$= \Theta(\boldsymbol{n^{\log_b a}})$

$$f(n) = \Theta(n^{\log_b a} \lg^k n)$$

for some constant $k \ge 0$.

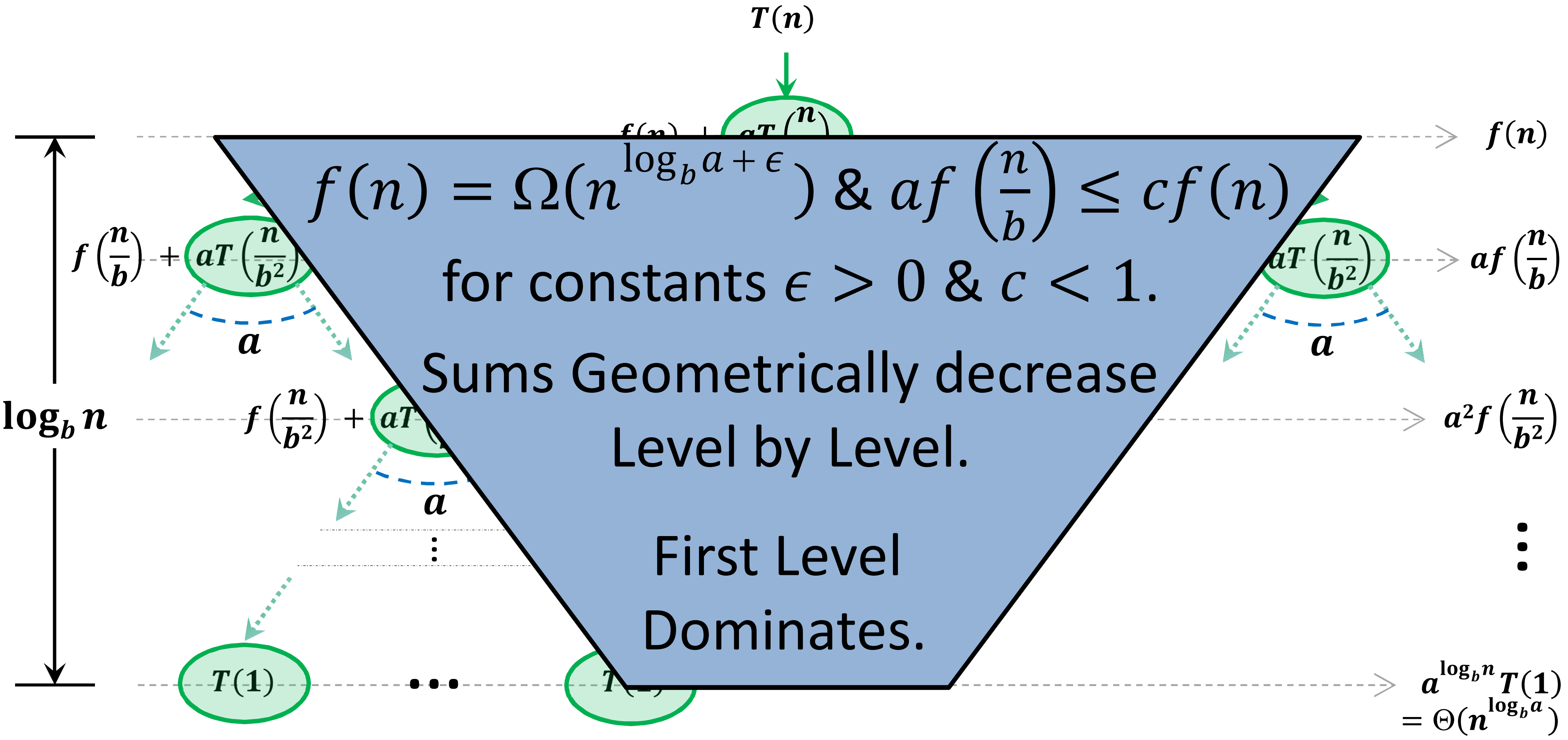
Sums Arithmetically Increase Level by Level.

No Level Dominates.

$$\boxed{\boldsymbol{T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)}}$$

How the Recurrence Unfolds: Case 3

$$T(n) = \begin{cases} \Theta(1), & if\ n \le 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise. \end{cases}$$



$T(n)$
$f(n) + aT\left(\frac{n}{b}\right)$
$f\left(\frac{n}{b}\right) + aT\left(\frac{n}{b^2}\right)$
a
a
$\log_b n$
$f\left(\frac{n}{b^2}\right) + aT$
a
⋮
$T(1)$
…
$T(1)$
$f(n)$
$af\left(\frac{n}{b}\right)$
$a^2 f\left(\frac{n}{b^2}\right)$
⋮
$a^{\log_b n} T(1)$
$= \Theta(n^{\log_b a})$

$$f(n) = \Omega(n^{\log_b a + \epsilon}) \ \&\ af\left(\frac{n}{b}\right) \le cf(n)$$

for constants $\epsilon > 0$ & $c < 1$.

Sums Geometrically decrease Level by Level.

First Level Dominates.

$$\boxed{T(n) = \Theta\big(f(n)\big)}$$

The Master Theorem

$$T(n) = \begin{cases} \Theta(1), & if\ n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & otherwise\ (a \geq 1, b > 1). \end{cases}$$

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

$$\boxed{T(n) = \Theta(n^{\log_b a})}$$

Case 2: $f(n) = \Theta(n^{\log_b a} \lg^k n)$ for some constant $k \geq 0$.

$$\boxed{T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)}$$

Case 3: $f(n) = \Omega(n^{\log_b a + \epsilon})$ and $af\left(\frac{n}{b}\right) \leq cf(n)$

for constants $\epsilon > 0$ and $c < 1$.

$$\boxed{T(n) = \Theta\big(f(n)\big)}$$

Example Applications of Master Theorem

Example 1: $T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$

Master Theorem Case 1: $T(n) = \Theta\left(n^{\log_2 3}\right)$

Example 2: $T(n) = 7T\left(\frac{n}{2}\right) + \Theta(n^2)$

Master Theorem Case 1: $T(n) = \Theta\left(n^{\log_2 7}\right)$

Example 3: $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$

Master Theorem Case 2: $T(n) = \Theta(n \log n)$

Assuming that we have an infinite number of processors, and each recursive call in example 2 above can be executed in parallel:

Example 4: $T(n) = T\left(\frac{n}{2}\right) + \Theta(n^2)$

Master Theorem Case 3: $T(n) = \Theta(n^2)$