Théorème maître

Antoine Dequay

21 septembre 2022

Notes

- Prof : Lilian Besson.
- Leçon : 903, 926, 931.
- Références :
  - Beauquier.

Théorème 1 Soit $T : \mathbb{N} \longrightarrow \mathbb{R}_+$ croissante à partir d'un certain rang. On suppose qu'il existe un rang $n_0 \geq 1$, des entiers $b \geq 2$ et $k \geq 0$ et des réels $a, c, d > 0$ tels que la relation de récurrence suivante soit vérifiée :

$$\begin{cases} T(n_0) = d \\ T(n) = aT\left(\dfrac{n}{b}\right) + cn^k \text{ pour tout entier de la forme } n = b^p n_0 \text{ avec } p \in \mathbb{N} \end{cases}$$

Alors on a la disjonction de cas suivante :

— Si $a < b^k$ alors $T(n) = \Theta(n^k)$

— Si $a = b^k$ alors $T(n) = \Theta(n^k log_b(n))$

— Si $a > b^k$ alors $T(n) = \Theta(n^{log_b(a)})$

Preuve. Soit $p \in \mathbb{N}$, $n = b^p n_0$. Par récurrence, on a :

$$T(n) = da^p + \sum_{i=0}^{p-1} ca^i \left(\frac{n}{b^i}\right)^k = \underbrace{da^p}_{=\delta(n)} + \underbrace{cn^k \sum_{i=0}^{p-1} \left(\frac{a}{b^k}\right)^i}_{=\gamma(n)}$$

On a $\delta(n) = da^p = da^{log_b(n) - log_b(n_0)} = \dfrac{d}{a^{log_b(n_0)}} b^{log_b(a) log_b(n)} = \dfrac{d}{a^{log_b(n_0)}} n^{log_b(a)} = \Theta(n^{log_b(a)})$.

De plus, $\gamma(n)$ est une somme partielle d'une série géométrique. On distingue trois cas :

— Si $a < b^k$, la série géométrique de raison $\dfrac{a}{b^k}$ converge, donc $\gamma(n)$ est bornée. On a donc

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

car $log_b(a) < k$.

— Si $a = b^k$, alors $\gamma(n) = p = log_b(n) - log_b(n_0) = \Theta(log_b(n))$. Donc

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

car $log_b(a) = k$.

— Si $a > b^k$, alors $\gamma(n) \sim \alpha \dfrac{a^p}{b^{kp}}$ avec $\alpha = \dfrac{1}{\frac{a}{b^k} - 1}$. Donc $cn^k \gamma(n) \sim c\alpha \left(\dfrac{n}{b^p}\right)^k a^p \sim c\alpha n_0^k a^p$. Or $a^p = \Theta(n^{log_b(a)})$. On a donc :

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

On a donc le résultat pour n de la forme $n = b^p n_0$ avec $p \in \mathbb{N}$.

Soit $n \geq n_0$ et $p \in \mathbb{N}$ tel que $b^p n_0 \leq n \leq b^{p+1} n_0$. Puisque T est croissante à partir d'un certain rang, pour n assez grand on a $T(b^p n_0) \leq T(n) \leq T(b^{p+1} n_0)$. Or pour $f \in \left\{m \longmapsto m^k,\ m \longmapsto m^k log_b(m),\ m \longmapsto m^{log_b(a)}\right\}$, on a $f(bm) = \Theta(f(m))$. Par encadrement, on a le résultat pour tout n à partir certain rang.

□

Application 2 Tri fusion : $a = 2$, $b = 2$, $k = 1$, $n_0 = 1$, $d = 1$, $c = 1$, d'où une complexité en $O(nlog_2(n))$ $(a = b^k)$.

Remarque
- On a une version similaire avec les O, mais aussi avec les même hypothèses et un $\leq$ à la place de l'égalité dans la relation de récurrence.
- Dans le cas du tri fusion, on n'a pas directement le résultat en posant $T(n)$ comme "le temps d'exécution dans le pire cas pour une entrée de taille n". Il faut regarder $T_m(n) := \max_{k \leq n} T(k)$, lui appliquer le théorème maître (version avec l'inégalité et donc le O), et enfin remarquer que $T(n) \leq T_m(n)$ pour conclure.