INF4705 — Analyse et conception d'algorithmes

CHAPITRE 2 : NOTATION ASYMPTOTIQUE

Nous avons vu que nous souhaitions exprimer le temps d'exécution d'un algorithme à une constante multiplicative près. C'est ce qu'accomplit la *notation asymptotique*, qu'on dit asymptotique parce qu'elle décrit le comportement des fonctions au-delà d'un certain seuil. Nous l'utiliserons aussi pour d'autres ressources, comme l'espace mémoire.

2.1 $\mathcal{O}$, Ω et Θ

$\star$ Soit une fonction $f : \mathbb{N} \to \mathbb{R}_+$,

$$\mathcal{O}(f) = \{t : \mathbb{N} \to \mathbb{R}_+ \mid (\exists c \in \mathbb{R}_+^\star)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)[t(n) \leq c \cdot f(n)]\}$$

- $g \in \mathcal{O}(f)$ se lit "g est *dans l'ordre de f*".
- Cette notation décrit une borne supérieure sur la consommation de ressource.

Notez que, par exemple, $3n + 4 \in \mathcal{O}(n), \in \mathcal{O}(n^3), \in \mathcal{O}(2^n)$.

$\star$ Soit la fonction $f : \mathbb{N} \to \mathbb{R}_+$,

$$\Omega(f) = \{t : \mathbb{N} \to \mathbb{R}_+ \mid (\exists c \in \mathbb{R}_+^\star)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)[t(n) \geq c \cdot f(n)]\}$$

- $g \in \Omega(f)$ se lit "g est *dans oméga de f*".
- Cette notation décrit une borne inférieure sur la consommation de ressource.

Il existe une dualité entre ces deux notations :

$$g \in \Omega(f) \Leftrightarrow f \in \mathcal{O}(g))$$

puisque $g(n) \geq c \cdot f(n) \equiv f(n) \leq \frac{1}{c} \cdot g(n)$.

Attention ! Quand on s'intéresse à l'analyse en pire cas, il y a une nuance subtile entre ce que disent les notations $\mathcal{O}$ et Ω. La première borne tous les exemplaires d'une taille donnée mais pas la seconde ...

$\star$ Soit la fonction $f : \mathbb{N} \to \mathbb{R}_+$,

$$\Theta(f) = \mathcal{O}(f) \cap \Omega(f)$$

ou encore,

$$\Theta(f) = \{t : \mathbb{N} \to \mathbb{R}_+ \mid (\exists c, d \in \mathbb{R}_+^\star)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)[d \cdot f(n) \leq t(n) \leq c \cdot f(n)]\}$$

- $g \in \Theta(f)$ se lit "g est *dans l'ordre exact de f*".
- Cette notation décrit à la fois une borne inférieure et une borne supérieure sur la consommation de ressource, à l'aide d'une même fonction.

Par le principe d'invariance, l'ordre (ou oméga, ou l'ordre exact) de f ne caractérise pas seulement la consommation de ressource d'une implantation particulière d'un algorithme mais bien celle de *n'importe laquelle* de ses implantations.

Exemple 1 *Soient $T_1(n) = 8n^2 + 3n + 2$ et $T_2(n) = 4000n^2 + 10n + 1$ des fonctions représentant le temps d'exécution en pire cas de deux implantations d'un même algorithme, en fonction de n, la taille de l'exemplaire.*

$$\begin{aligned} T_1(n) &= 8n^2 + 3n + 2 \\ &\leq 8n^2 + 3n^2 + 2n^2 \text{ dès que } n \geq 1 \\ &= 13n^2 \end{aligned}$$

Donc $T_1(n) \in \mathcal{O}(n^2)$, en prenant $c = 13$ et $n_0 = 1$. (On aurait tout aussi bien pu prendre $c = 20$ et $n_0 = 5$.)

$$\begin{aligned} T_2(n) &= 4000n^2 + 10n + 1 \\ &\leq 4000n^2 + 10n^2 + 1n^2 \text{ dès que } n \geq 1 \\ &= 4011n^2 \end{aligned}$$

Donc $T_2(n) \in \mathcal{O}(n^2)$, en prenant $c = 4011$ et $n_0 = 1$.

$\mathcal{O}(n^2) = \{8n^2 + 3n + 2,\ 4000n^2 + 10n + 1,\ n^2,\ n, \ldots\}$

On peut dire que l'algorithme lui-même prend un temps dans l'ordre de n^2. □

Remarques :

- Si j'ai deux algorithmes pour un même problème, A $\in \Theta(n^2)$ et B $\in \Theta(n)$, lequel devrais-je choisir ? Ça dépend entre autres des constantes ! Si A prend n^2 secondes et B n heures (c.-à-d. $3600n$ secondes), B ne sera meilleur que pour $n > 3600$.

  Donc, attention aux constantes multiplicatives cachées !
- Parce qu'on s'intéresse au comportement à une constante multiplicative près, il est habituellement inutile de préciser la base d'un logarithme, puisque $\log_a n = \log_a b \cdot \log_b n$ pour $a, b, n \in \mathbb{R}_+,\ a, b \neq 1$, et que $\log_a b$ est une constante positive si $a, b > 1$.
  Donc dans ce cas $\Omega(\log_a n) = \Omega(\log_b n)$.
  Mais attention ! on ne peut négliger la base lorsque le logarithme apparaît en exposant.
  P.e. $\Omega(2^{\lg n}) \neq \Omega(2^{\log_{10} n})$.
- On se permettra un abus de notation en travaillant avec des fonctions parfois négatives ou non définies, mais seulement pour un nombre fini de valeurs de n. Il suffira de hausser le seuil afin d'éviter ces “accidents de parcours”.
  Exemple 2
  On peut parler de $\mathcal{O}(\frac{n}{\lg n})$ même si cette fonction n'est pas définie pour $n = 0$ ou $n = 1$.
  On peut aussi écrire $n^3 - 3n^2 - 1 \in \mathcal{O}(n^3)$ même si $n^3 - 3n^2 - 1 < 0$ pour $n \leq 3$. □
- Les notations asymptotiques définissent donc des ensembles, qu'on peut comparer à l'aide des relations habituelles d'inclusion et d'égalité. Les notations $\mathcal{O}()$ et $\Omega()$ forment en fait une hiérarchie d'inclusions. Il existe certaines fonctions f et g définissant des ensembles incomparables c.-à-d. tels que

  $$f \notin \mathcal{O}(g) \text{ et } g \notin \mathcal{O}(f).$$

  Pouvez-vous en trouver ?

2.2 Notation asymptotique à plusieurs paramètres

Par exemple, pour résoudre des problèmes sur les graphes, où la taille d'un exemplaire est fonction du nombre de sommets et aussi du nombre d'arêtes.

Soit une fonction $f : \mathbb{N} \times \mathbb{N} \to \mathbb{R}_+$,

$$\mathcal{O}(f) = \{t : \mathbb{N} \times \mathbb{N} \to \mathbb{R}_+ \mid (\exists c \in \mathbb{R}_+^\star)(\exists m_0, n_0 \in \mathbb{N})(\forall m \geq m_0, n \geq n_0)[t(m,n) \leq c \cdot f(m,n)]\}$$

(et de la même façon pour Ω et Θ)

2.3 Quelques équivalences utiles (pour $\mathcal{O}$ mais aussi pour Ω et Θ)

Règle du maximum : $\mathcal{O}(f+g) = \mathcal{O}(\max(f,g))$

Démonstration 1

Nous démontrerons que toute fonction appartenant à l'un des ensembles appartient aussi à l'autre.
Remarquons que $f(n)+g(n) = \min(f(n),g(n)) + \max(f(n),g(n))$.
Donc, $\max(f(n),g(n)) \leq f(n)+g(n) \leq 2\cdot\max(f(n),g(n))$.
Soit une fonction quelconque $t \in \mathcal{O}(f+g)$ *et donc* $t(n) \leq c\cdot(f(n)+g(n))$ *pour un* c *approprié et* n *suffisamment grand.*

$$\begin{aligned} t(n) &\leq c\cdot(f(n)+g(n)) \\ &\leq c\cdot 2\cdot\max(f(n),g(n)) \end{aligned}$$

Ainsi $t \in \mathcal{O}(\max(f,g))$.
Soit maintenant une fonction quelconque $t \in \mathcal{O}(\max(f,g))$

$\vdots$

$t \in \mathcal{O}(f+g)$. □

Cette règle se généralise à un nombre fini et constant de fonctions. Elle permet de simplifier l'analyse et la notation.

Exemple 3

$\mathcal{O}(8n^2+3n+2) = \mathcal{O}(\max(8n^2,3n,2)) = \mathcal{O}(n^2)$.
$\mathcal{O}(n) = \mathcal{O}(n+n^2-n^2) = \mathcal{O}(\max(n,n^2,-n^2)) = \mathcal{O}(n^2)$. Qu'est-ce qui ne va pas ? $-n^2 < 0$ inf. souvent.

$$\begin{aligned} \mathcal{O}(12n^3\lg n - 5n^2 + \lg^2 n + 36) &= \mathcal{O}(11n^3\lg n + n^3\lg n - 5n^2 + \lg^2 n + 36) \\ &= \mathcal{O}(\max(11n^3\lg n, n^3\lg n - 5n^2, \lg^2 n, 36)) \\ &= \mathcal{O}(11n^3\lg n) = \mathcal{O}(n^3\lg n) \end{aligned}$$

□

Opérations sur la notation asymptotique : quelques équivalences pour simplifier les calculs.

$$\mathcal{O}(f) + \mathcal{O}(g) = \mathcal{O}(f+g) = \mathcal{O}(\max(f,g))$$
$$\mathcal{O}(f) \times \mathcal{O}(g) = \mathcal{O}(f\cdot g)$$

Note : le résultat de ces opérations arithmétiques sur des ensembles de fonctions doit être interprété comme l'ensemble des fonctions obtenues en appliquant l'opération sur les paires de fonctions issues de $\mathcal{O}(f)$ et $\mathcal{O}(g)$.

Règle de la limite : très utile pour déterminer notre hiérarchie.

1. $\lim_{n\to\infty} \frac{f(n)}{g(n)} \in \mathbb{R}_+^\star$ ssi $[f(n) \in \Theta(g(n))]$.
2. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ ssi $[f(n) \in \mathcal{O}(g(n))$ mais $f(n) \notin \Theta(g(n))]$.
3. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ ssi $[f(n) \in \Omega(g(n))$ mais $f(n) \notin \Theta(g(n))]$.

Exemple 4

Soient $f(n) = \lg n$ *et* $g(n) = \sqrt{n}$ *:*

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{\lg n}{\sqrt{n}} = \textit{(par l'Hôpital)} \lim_{n\to\infty}\frac{(\lg n)'}{(\sqrt{n})'} = \lim_{n\to\infty}\frac{1/n}{1/(2\sqrt{n})} = \lim_{n\to\infty}\frac{2}{\sqrt{n}} = 0$$

Donc $\lg n \in \mathcal{O}(\sqrt{n})$ *mais* $\sqrt{n} \notin \mathcal{O}(\lg n)$

Exemple 5

Afin de nous faire la main, prouvons que $\forall k \in \mathbb{N}$ *donné,* $\Sigma_{i=1}^{n} i^k \in \Theta(n^{k+1})$.

Démonstration 2

$$\Sigma_{i=1}^{n} i^k \leq \Sigma_{i=1}^{n} n^k = n^{k+1} \text{ et donc } \Sigma_{i=1}^{n} i^k \in \mathcal{O}(n^{k+1}) \text{ avec } c = 1.$$

$$\Sigma_{i=1}^{n} i^k \geq \Sigma_{i=\lceil \frac{n}{2} \rceil}^{n} i^k \geq \Sigma_{i=\lceil \frac{n}{2} \rceil}^{n} (\frac{n}{2})^k \geq \frac{n}{2} \cdot (\frac{n}{2})^k = \frac{n^{k+1}}{2^{k+1}} \text{ et donc } \Sigma_{i=1}^{n} i^k \in \Omega(n^{k+1}) \text{ avec } c = 1/2^{k+1}.$$

$\square$