



**HAL**  
open science

## Algorithmique et complexité

Olivier Ridoux

► **To cite this version:**

| Olivier Ridoux. Algorithmique et complexité. Licence. France. 2021. hal-03212394

**HAL Id: hal-03212394**

**<https://univ-rennes.hal.science/hal-03212394v1>**

Submitted on 29 Apr 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## Mode d'emploi :

- Lire cette page de prérequis, **calcul** et **logique**, comprendre, apprendre le cas échéant.
- Préparer les plis (voir traits gris rentrants et traits **rouges** saillants au verso de cette page).
- Découper selon le trait rouge entre les deux ● du verso de cette page, puis achever le pliage.

### 1 Puissances, racines et logarithmes

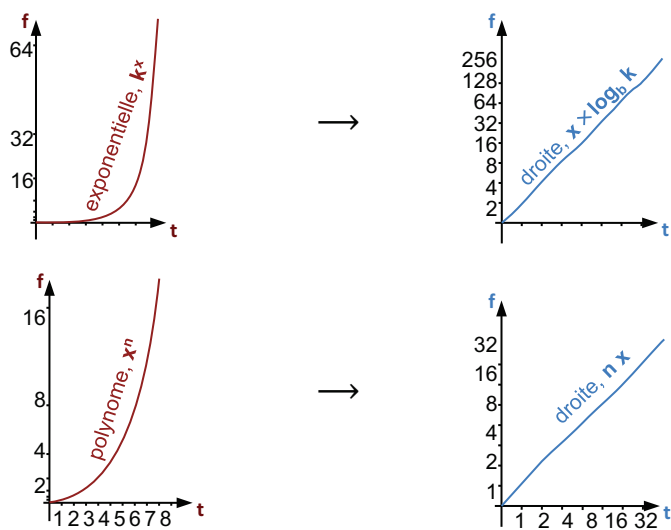
La **n<sup>ème</sup> puissance** d'un nombre  $x$ , notée  $x^n$ , est le produit de  $n$  copies de ce nombre. La **racine n<sup>ème</sup>** d'un nombre  $x$ , notée  $\sqrt[n]{x}$ , est un nombre  $y$  tel que  $y^n = x$ , soit **de quoi  $x$  est le produit de  $n$  copies** ;  $y$  est en général **irrationnel**. Par exemple,  $\sqrt[12]{2} = 1.05946\dots$  est le **rapport des fréquences de deux notes de musique séparées par 1 demi-ton** ; c'est de quoi 2 (le rapport de fréquence d'un octave) est le produit de 12 copies (les 12 demi-tons de l'octave). Initialement conçues pour  $n$  à valeur entière, ces opérations ont été progressivement généralisées par continuité à toutes sortes de nombres, si bien que puissances et racines sont maintenant deux instances du même concept. On montre que  $x^{a+b} = x^a \times x^b$  et  $x^{a \cdot b} = (x^a)^b = (x^b)^a$ , que  $x^{-n} = 1/x^n$  et  $x^{1/n} = \sqrt[n]{x}$ , et que  $x^{a/b}$  est un nombre  $y$  tel que  $x^a = y^b$ .

Le **logarithme en base  $b$**  d'un nombre  $x$ , noté  $\log_b x$ , est un nombre  $y$  tel que  $b^y = x$ , soit **de quel nombre de copies de  $b$   $x$  est le produit** ;  $y$  est en général **non-algébrique**, c-à-d. pas racine d'un polynôme à coefficients rationnels. On montre que  $\log_b x$  est strictement croissante, que  $\log_b 1 = 0$  et  $\log_b b = 1$ , que  $\log_b (x \cdot y) = (\log_b x) + (\log_b y)$ , que  $\log_b x = (\log_b x) \times (\log_b b^y)$ , et que  $x^{\log_b y} = y^{\log_b x}$ . Noter que en général  $\log_b x \ll x$ . Par exemple, **le logarithme en base 10 d'un nombre est à peu près le nombre des chiffres de sa représentation décimale** : 1,95... pour 90, 2,99... pour 990 et 5,999... pour 999000. L'erreur faite en arrondissant un logarithme à un entier voisin est bornée de la façon suivante :  $n \leq b^{\log_b n} < b \times n$  et  $n/b < b^{\lfloor \log_b n \rfloor} \leq n$ . Par exemple,  $4 \leq 2^{\lfloor \log_2 4 \rfloor} < 8$ ,  $7 \leq 2^{\lfloor \log_2 7 \rfloor} = 8 < 2 \times 7 = 14$ ,  $8/2 = 4 < 2^{\lfloor \log_2 8 \rfloor} \leq 8$ , et  $5/2 = 2,5 < 2^{\lfloor \log_2 5 \rfloor} = 4 \leq 5$ . Noter enfin que pour tout  $\delta \geq 0$ ,  $\lim_{x \rightarrow \infty} x^\delta / \log_b x = \infty$  ; une puissance finit toujours par dépasser un logarithme.

Le logarithme permet des **changements de variable** intéressants. Quand un phénomène a une dimension qui croit par multiplication d'un facteur constant à chaque pas de temps, par exemple, **loi de Moore, multiplication par 2 de la capacité des disques durs tous les 2 ans, doublement du nombre de grains de riz à chaque case de l'échiquier**, on dit qu'il a une **croissance exponentielle**. Ces phénomènes sont difficiles à représenter graphiquement car le rapport entre les plus petites valeurs et les plus grandes est énorme : par exemple, **plusieurs millions sur l'histoire des capacités des disques durs**, c'est-à-dire beaucoup plus que le nombre de points d'impression sur la hauteur d'une page A4.

Le changement de variable  $(t, f(t)) \rightarrow (t, \log_b f(t))$ , c-à-d. **afficher  $\log_b f(x)$  plutôt que  $f(x)$  à l'abscisse  $x$** , rend le graphe de  $f$  plus lisible et l'hypothèse de croissance exponentielle facile à vérifier si les points s'alignent sur une droite ; on appelle cela la **projection lin  $\times$  log**.

On peut aussi convenir du changement de variable  $(t, f(t)) \rightarrow (\log_b t, \log_b f(t))$ , c-à-d. **afficher  $\log_b f(x)$  à l'abscisse  $\log_b x$  plutôt que  $f(x)$  à l'abscisse  $x$**  pour former la **projection log  $\times$  log**. Son avantage est que le graphe des phénomènes de croissance polynomiale  $y$  tend vers une droite dont la pente est le degré du polynôme. C'est un moyen très simple de vérifier si un phénomène a une croissance polynomiale ou non.



### N'oubliez jamais !

Un **modèle** est toujours **imparfait**, parfois **utile**, et c'est tout ce qu'on peut lui demander (d'après George Box 1978, voir aussi les cartes à l'échelle 1/1 de Jorge Luis Borges 1946 et Lewis Carroll 1893 qui sont parfaites et inutiles, parfaitement inutiles).

L'informatique ne travaille que sur la **représentation** des choses ; il lui faut des capteurs et des actionneurs pour toucher les choses. Par contre, travailler sur des représentations de représentations de ... est courant. Par exemple, **440** est une représentation d'un entier qui peut être la représentation de la mesure d'une fréquence, tout comme **188**, et **La3** pourrait être une autre représentation de la même fréquence. Il faut toujours se demander ce qui a du sens par rapport à ce qui est modélisé.

### 1 Calcul des propositions

**Propositions atomiques** (ou **atomes**) : Ce sont des formules élémentaires à qui on peut attribuer une valeur de vérité, **Vrai** ou **Faux**. On les note souvent par des lettres minuscules : ex. **a, b, c**.

**Formules propositionnelles** : Ce sont des formules, élémentaires ou non, qui sont reliées par des connecteurs logiques, généralement **∧, ∨, ¬, ⇒**.

- **conjonction**,  $\phi_1 \wedge \phi_2$  : relie deux formules pour en former une troisième qui est vraie ssi les deux premières le sont. Le connecteur **∧** est commutatif, associatif, et a pour élément neutre la valeur de vérité **Vrai**. Cela permet la notation de conjonction étendue comme  $\bigwedge_{i \in [1, n]} \phi_i$ . Si on convenait que **Faux < Vrai**,  $\phi_1 \wedge \phi_2$  serait le minimum de  $\phi_1$  et  $\phi_2$ . Si on convenait que **Faux = 0** et **Vrai = 1**,  $\phi_1 \wedge \phi_2$  serait  $\phi_1 \times \phi_2$ .
- **disjonction**,  $\phi_1 \vee \phi_2$  : relie deux formules pour en former une troisième qui est vraie ssi au moins une des deux premières l'est. Le connecteur **∨** est commutatif, associatif, et a pour élément neutre la valeur de vérité **Faux**. Cela permet la notation de disjonction étendue comme  $\bigvee_{i \in [1, n]} \phi_i$ . Si on convenait que **Faux < Vrai**,  $\phi_1 \vee \phi_2$  serait le maximum de  $\phi_1$  et  $\phi_2$ . Si on convenait que **Faux = 0** et **Vrai = 1**,  $\phi_1 \vee \phi_2$  serait  $1 - (1 - \phi_1) \times (1 - \phi_2)$ .
- **implication**,  $\phi_1 \Rightarrow \phi_2$  : relie deux formules pour en former une troisième qui est fautive ssi  $\phi_1$  est vraie alors que  $\phi_2$  est fautive. Le connecteur **⇒** n'est ni commutatif ni associatif. Si on convenait que **Faux < Vrai**,  $\phi_1 \Rightarrow \phi_2$  serait **Vrai** ssi  $\phi_1 \leq \phi_2$ .
- **négation**,  $\neg \phi$  : constitue une formule qui est fautive ssi  $\phi$  est vraie. Si on convenait que **Faux = 0** et **Vrai = 1**,  $\neg \phi$  serait  $1 - \phi$ .

a	b	c	(a∨b)∧c
Vrai	Vrai	Vrai	Vrai
Vrai	Vrai	Faux	Faux
Vrai	Faux	Vrai	Vrai
Vrai	Faux	Faux	Faux
Faux	Vrai	Vrai	Vrai
Faux	Vrai	Faux	Faux
Faux	Faux	Vrai	Faux
Faux	Faux	Faux	Faux

a	a∧¬a
Vrai	Faux
Faux	Faux

a	a∨¬a
Vrai	Vrai
Faux	Vrai

**Satisfaisabilité** : les formules du calcul des propositions peuvent être vues comme des fonctions des variables qu'elles contiennent :  $\{\text{Vrai, Faux}\}^n \rightarrow \{\text{Vrai, Faux}\}$  pour une formule qui compte  $n$  variables. On peut représenter chacune de ces fonctions par une **table de vérité**. Il s'agit d'un tableau comportant  $2^n$  lignes, et  $n+1$  colonnes. Dans leurs  $n$  premières colonnes, les lignes représentent toutes les entrées possibles (appelées **valuations** des variables), et dans leur  $n+1$ -ème colonne, elles représentent ce que vaut la fonction pour cette entrée. Ex. les formules **(a∨b)∧c**, **a∨¬a**, et **a∧¬a** peuvent être représentées par les tables de vérité ci-contre.

On peut remarquer que toutes les lignes de la table de **a∨¬a** ont la valeur **Vrai** en position de résultat ; on dit que cette formule est une **tautologie**, elle vaut **Vrai** pour toutes ses entrées. Toutes les lignes de la table de **a∧¬a** ont la valeur **Faux** ; on dit que cette formule est **contradictoire** ou **absurde**. Finalement, les lignes de la table de **(a∨b)∧c** ont soit **Vrai** soit **Faux** ; on dit que cette formule est **satisfaisable** (bien sûr, les tautologies sont aussi satisfaisables). Il est souvent intéressant de savoir pour quelles entrées une formule vaut **Vrai**.

### 2 Calcul des prédicats

Dans sa définition la plus naïve le **calcul des prédicats** est la formalisation de la logique employée tous les jours dans les activités un peu scientifiques. Elle permet d'exprimer des **jugements** sous forme de formules, et de décider formellement si ils sont **vrais** ou **faux**. Les formules de la logique des prédicats sont formées des connecteurs du calcul des propositions (ex. **∧, ∨, ¬, et ⇒**) et des quantificateurs **∀** et **∃**.

**Formules quantifiées** : Ce sont des formules, élémentaires ou non, dont la valeur de vérité s'évalue par rapport à un ensemble d'objets, plutôt que par rapport à un objet. Syntaxiquement, une quantification lie une variable, comme le font  $\int \dots dx$  ou  $\partial \dots / \partial x$ . Sémantiquement, les quantifications sont définies comme suit :

- **quantification universelle**,  $\forall x . \phi(x)$  : constitue une formule qui est vraie ssi  $\phi$  est vraie de tous les éléments du domaine. Si le domaine est fini, la quantification universelle est juste une **conjonction** des applications de  $\phi$  à tous les éléments du domaine.
- **quantification existentielle**,  $\exists x . \phi(x)$  : constitue une formule qui est vraie ssi  $\phi$  est vraie d'au moins un élément du domaine. Si le domaine est fini, la quantification existentielle est juste une **disjonction** des applications de  $\phi$  à tous les éléments du domaine.

**Parenthèses et priorité des opérateurs** : Il est courant d'assigner des priorités d'opérateurs aux différents connecteurs et quantificateurs afin de spécifier comment se lisent les formules complexes en l'absence de parenthèses. C'est utile mais absolument pas fondamental car ces conventions dépendent trop des outils, et même quand ce n'est pas le cas, il n'est jamais très prudent de se reposer trop lourdement sur les priorités des opérateurs quand le coût de quelques parenthèses est si faible devant le coût d'une grosse erreur. Ce genre d'expertise est à réserver aux experts. Il vaut mieux ne pas être avare de parenthèses, ex. utiliser les parenthèses rondes **(...)** pour structurer les connecteurs, et les parenthèses carrées (ou crochets, **[...]**) pour les quantificateurs.

### 3 Idioms logiques

En pratique, on n'utilise pas n'importe quelle formule de la logique des prédicats. On a tendance à utiliser des imbrications de quantificateurs et de connecteurs qui sont idiomatiques et qu'il convient de reconnaître et interpréter correctement au premier coup d'œil.

**Quantifications dans un domaine** : Très souvent, on écrit des formules comme  $\forall x \in E . \phi(x)$  ou  $\forall x \text{ tq } \psi(x) . \phi(x)$ . Les  $x \in E$  et  $\text{tq } \psi(x)$  sont une façon de dire dans quel domaine doit être évaluée la quantification. Ces formules **doivent** être lues  $\forall x . [x \in E \Rightarrow \phi(x)]$  et  $\forall x . [\psi(x) \Rightarrow \phi(x)]$ . Une conséquence directe est qu'une quantification universelle sur un domaine vide est trivialement vraie. Cela paraît une situation étrange, mais c'est banal en algorithmique, spécialement quand on considère les cas d'initialisation : ex. **Tous les utilisateurs sont ...** lorsqu'il n'y a pas d'utilisateurs. De la même façon, on écrit des formules comme  $\exists x \in E . \phi(x)$  ou  $\exists x \text{ tq } \psi(x) . \phi(x)$ . Ces formules **doivent** être lues  $\exists x . [x \in E \wedge \phi(x)]$  et  $\exists x . [\psi(x) \wedge \phi(x)]$ . On voit alors qu'une quantification existentielle sur un domaine vide est trivialement fautive.

**Cascades de quantifications** : On imbrique souvent les quantifications. Certaines imbrications doivent faire réfléchir, et d'autres moins.

- $\exists x . \exists y . \phi(x, y)$  est équivalent à  $\exists y . \exists x . \phi(x, y)$  qu'on note souvent  $\exists x, y . \phi(x, y)$  ; l'ordre des quantificateurs ne compte pas.
- $\forall x . \forall y . \phi(x, y)$  est équivalent à  $\forall y . \forall x . \phi(x, y)$  qu'on note souvent  $\forall x, y . \phi(x, y)$  ; l'ordre des quantificateurs ne compte pas non plus.
- $\forall x . \exists y . \phi(x, y)$  exprime que pour chaque  $x$  il y a un  $y$ , **qui peut dépendre de  $x$** , qui a la propriété désirée. Ex. dans  $\forall x . \exists y . [x \times y = 0]$ , le même  $y$  convient pour tous les  $x$ , mais dans  $\forall x . \exists y . [x \times y = 1]$ , à tout  $x$  correspond un  $y$  différent. Un  **$y$  existentiel** est donc implicitement une fonction de tous les  $x$  **universels** qui le précèdent.
- $\exists x . \forall y . \phi(x, y)$  exprime qu'un  $x$  unique a la propriété désirée pour tous les  $y$ . Ex.  $\exists x . \forall y . [x \times y = 0]$  est vrai, mais pas  $\exists x . \forall y . [x \times y = 1]$ . On voit donc que lorsqu'il y a une **alternance de quantificateurs** l'ordre compte ! Ces alternances sont l'essence de nombreuses définitions importantes en mathématiques (ex. continuité, convergence) et en informatique (ex. ordres de grandeurs au verso de cette page).

### 2 Sommations

La somme et le produit étant des opérations **associatives**, **commutatives** et ayant chacune un **élément neutre**, on peut se permettre des notations de **sommes** et **produits itérés** :  $\sum_{i \in [b_1, b_2]} t_i$  et  $\prod_{i \in [b_1, b_2]} f_i$  où les  $t_i$  et  $f_i$  sont des **termes** et **facteurs** dépendants de  $i$ . Si l'intervalle des indices est vide (ex. **[0 0]**) la valeur conventionnelle de l'expression est l'**élément neutre** de l'opération, ex. **0** pour la somme et **1** pour le produit, mais aussi **Faux** pour la disjonction (**∨**), qui est une sorte de somme, et **Vrai** pour la conjonction (**∧**), qui est une sorte de produit.

Certaines sommations doivent être connues ; il est même bon de savoir les retrouver :

- $\sum_{i \in [1, n]} i$  : somme des  $n$  premiers entiers (c-à-d. de 1 à  $n$ ). C'est égal à  $n \times (n+1) / 2$ . Ex.  $\sum_{i \in [1, 17]} i = 153$ .
- $\sum_{i \in [0, n]} x^i$  : somme des  $n$  premières puissances de  $x$  (c-à-d. de  $x^0$  à  $x^{n-1}$ ). C'est égal à  $(x^n - 1) / (x - 1)$ . Ex.  $\sum_{i \in [0, n]} 2^i = 2^n - 1$ .

Il est bon aussi de se rappeler quelques transformations élémentaires :

- $\sum_{i \in [0, n]} (k \times t_i) = k \times \sum_{i \in [0, n]} t_i$  et  $\sum_{i \in [0, n]} k = k \times n$ , si  $k$  ne dépend pas de  $i$ .
- $\sum_{i \in [0, n]} (s_i + t_i) = \sum_{i \in [0, n]} s_i + \sum_{i \in [0, n]} t_i$ . Attention, rien de tel pour  $\sum_{i \in [0, n]} (s_i \times t_i)$  !