CHAPTER 9

# ASYMPTOTIC BEHAVIOUR

In Chapter 7 we saw that, in order to evaluate a complexity or a cost, estimating its order of magnitude could be useful. This can happen when exact computation is not possible (for instance, the average complexity of Quicksort, see Section 14.1), or when an approximate value of the cost or complexity is enough. For instance, comparing it to the cost of other algorithms or, *a priori*, excluding a too costly algorithm or determining the maximum size of data which can be dealt with by a given algorithm, etc.

This evaluation of an order of magnitude consists of finding an approximation of the behaviour of a function in limit conditions ($n$ going to infinity, $x$ going to zero, etc.); this is why such behaviour is called *asymptotic*. Most often, such evaluations will be used to study the complexity of an algorithm when the size $n$ of the data goes to infinity. Note, however, that an algorithm which is optimal for large-size data is not always the best one for smaller-size data.

After the basic definitions, this chapter introduces methods to determine the asymptotic behaviour of functions, and to classify functions according to their asymptotic behaviour.

We recommend the following handbook:

Ronald Graham, Donald Knuth, Oren Patashnik, *Concrete Mathematics*, Addison-Wesley, London (1989).

## 9.1 Generalities

### 9.1.1 Definitions

**Definition 9.1** *Let $f, g$ be two mappings from $\mathbb{N}$ into $\mathbb{R}^+$. $f$ is said to be dominated by $g$ or to be a 'big-Oh' of $g$ (resp. $f$ is said to dominate $g$ or to be a 'big-Omega' of $g$), and we note $f = O(g)$ (resp. $f = \Omega(g)$) if and only if*

$$\exists c \in \mathbb{R}^+,\ \exists n_0 \in \mathbb{N},\ \forall n > n_0\ :\ f(n) \leq cg(n) \quad (\textit{resp. } f(n) \geq cg(n)).$$

*This definition also holds for functions with several arguments, for instance* $f(n,p) = O(g(n,p))$ *if and only if*

$$\exists c, n_0, p_0,\ \forall n > n_0\,, \forall p > p_0\ :\quad f(n,p) \le cg(n,p).$$

The $O$-notation was introduced by the German mathematician Bachmann at the end of the last century (1894), and was made popular by his fellow countryman Landau, after whom it is named.

REMARK 9.2 The equality in the notation $f = O(g)$ is somewhat inappropriate but handy: for instance $n^2 + 2n = O(n^2)$, but no equality holds, since also $n^2 + 2n = O(n^5)$, but not $O(n^2) = O(n^5)$! To be quite precise, we should write $f \in O(g)$, and we would then have $O(n^2) \subsetneq O(n^5)$; but we will follow the standard notation by writing $f = O(g)$.

**Definition 9.3** *$f$ and $g$ are said to have the same order of magnitude, and this is denoted by $f = \theta(g)$, if and only if $f = O(g)$ and $g = O(f)$.*

**Proposition 9.4** $f = \theta(g) \implies f = O(g)$, *but the converse is false.*

*Proof.* Immediate as $f = \theta(g)$ if and only if $f = O(g)$ and $g = O(f)$. The converse is false, e.g. $n^p = O(n^{p+1})$. □

**Proposition 9.5** $f = O(g) \iff g = \Omega(f)$.

*Proof.* Immediate since $\exists c \in \mathbb{R}^+,\ \exists n_0 \in \mathbb{N},\ \forall n > n_0$: $f(n) \le cg(n)$ if and only if $\exists c' = \dfrac{1}{c} \in \mathbb{R}^+,\ \exists n_0 \in \mathbb{N},\ \forall n > n_0$: $g(n) \ge c' f(n)$. □

**Definition 9.6** *$f$ is said to grow more slowly than $g$, and we write $f = o(g)$, or $f \prec g$, if and only if* $\forall \varepsilon \in \mathbb{R}^+,\ \exists n_0 \in \mathbb{N},\ \forall n > n_0$: $f(n) \le \varepsilon g(n)$.

REMARK 9.7 The notations $O, o, \Omega, \theta$ also hold for mappings from $\mathbb{R}$ into $\mathbb{R}$; but in this case, when speaking of limit conditions we must specify the limits we are talking about, because the same function $f$ may be an $O(g)$ when $x \longrightarrow \infty$, and an $O(g')$ for a quite different $g'$ when $x \longrightarrow 0$: for instance, let $f(x) = x^3 + 4x^2 + x$; then

- when $x \longrightarrow \infty$, $f = O(x^3)$, but $f = O(x)$ does not hold (on the contrary $f = \Omega(x)$, and even $x = o(f)$, since $x/f(x) \longrightarrow 0$),
- symmetrically, when $x \longrightarrow 0$, $f = O(x)$, but $f = O(x^3)$ does not hold (on the contrary $f = \Omega(x^3)$, and even $x^3 = o(f)$).

### 9.1.2 Operations on the orders of magnitude

**Proposition 9.8** *For all $g$, $g_1$, $g_2$ :*

$$g(n) = O(g(n))$$
$$g(n) = \theta(g(n))$$
$$cO(g(n)) = O(g(n))$$
$$O(g(n)) + O(g(n)) = O(g(n))$$
$$O(g_1(n)) + O(g_2(n)) = O(\max(g_1(n), g_2(n)))$$
$$O(O(g(n))) = O(g(n))$$
$$O(g_1(n)) \cdot O(g_2(n)) = O(g_1(n))g_2(n) = g_1(n)O(g_2(n)) \ .$$

*These equalities are abbreviated notations, for instance the third one stands for: $f = O(g(n)), f' = cf \implies f' = O(g(n))$, etc.*

*Proof.* Let us check for instance that

$$O(g_1(n)) + O(g_2(n)) = O(\max(g_1(n), g_2(n)))\,;$$

if $f(n) \in O(g_1(n)) + O(g_2(n))$, $c_1$ and $c_2$ exist such that $f(n) \le c_1 g_1(n) + c_2 g_2(n)$ for $n \ge n_0$. Letting $c = \max(c_1, c_2)$, then

$$f(n) \le c(g_1(n) + g_2(n)) \le 2c \max(g_1(n), g_2(n))\,. \qquad \square$$

EXERCISE 9.1 Find the error in the following argument: '$n = O(n)$ and $2n = O(n)$, and so on, hence $\sum_{k=1}^{n} kn = \sum_{k=1}^{n} O(n) = nO(n) = O(n^2)$'. ♢

EXERCISE 9.2 If $f(n) = O(n)$, do the following hold?:

1. $(f(n))^2 = O(n^2)$,
2. $2^{f(n)} = O(2^n)$. ♢

**Proposition 9.9**

1. $f = \theta(g) \implies \lambda f = \theta(g)$ *for any constant* $\lambda > 0$, *and*
2. $f_i = \theta(g_i)$ *for* $i = 1, 2, \ldots, k$, *implies* $\sum_{i=1}^{k} f_i = \theta(\max\{g_1, g_2, \ldots, g_n\})$.

*Proof.* 1 is immediate; to verify 2 note that

- $\sum_{i=1}^{k} f_i = O(\max\{g_1, g_2, \ldots, g_n\})$, by induction on $k$ and applying the fifth equality of Proposition 9.8,
- for $i = 1, 2, \ldots, k$, $g_i = O(f_i)$ implies that $g_i = O\big(\sum_{i=1}^{k} f_i\big)$, and thus $\max\{g_1, g_2, \ldots, g_n\} = O\big(\sum_{i=1}^{k} f_i\big)$. $\square$

EXAMPLE 9.10 $\forall k, \quad \sum_{i=1}^{n} i^k = \theta(n^{k+1})$. The proof is by induction on $n$:

- Basis. For $k = 0, \quad \sum_{i=1}^{n} i^0 = n = \theta(n)$;
- Inductive step. Assuming $\forall j < k$, $\sum_{i=1}^{n} i^j = \theta(n^{j+1})$, note that

$$i^{k+1} = ((i-1)+1)^{k+1} = \sum_{p=0}^{k+1} \binom{k+1}{p} (i-1)^p,$$

and let us apply this formula for $i = 1, 2, \ldots, n$:

$$\begin{aligned} n^{k+1} &= (n-1)^{k+1} + (k+1)(n-1)^k + \cdots + 1 \\ (n-1)^{k+1} &= (n-2)^{k+1} + (k+1)(n-2)^k + \cdots + 1 \\ &\vdots \\ 2^{k+1} &= 1^{k+1} + (k+1)1^k + \cdots + 1 \\ 1^{k+1} &= 0 + 0 + \cdots + 1 \ . \end{aligned}$$

By summing these $n$ equalities, we obtain, after simplifications,

$$n^{k+1} = (k+1) \sum_{i=1}^{n-1} i^k + \binom{k+1}{2} \sum_{i=1}^{n-1} i^{k-1} + \cdots + n \ ;$$

hence,

$$\sum_{i=1}^{n-1} i^k = \frac{n^{k+1}}{k+1} - \frac{1}{k+1} \left[ \left( \binom{k+1}{2} \sum_{i=1}^{n-1} i^{k-1} \right) + \cdots + n \right] ,$$

and

$$\sum_{i=1}^{n} i^k = \frac{(n+1)^{k+1}}{k+1} - \frac{1}{k+1} \left[ \sum_{j=2}^{k+1} \binom{k+1}{j} \left( \sum_{i=1}^{n} i^{k+1-j} \right) \right] .$$

We then have by the induction, $\forall j = 2, \ldots, k+1$,

$$\frac{\binom{k+1}{j}}{k+1} \sum_{i=1}^{n} i^{k+1-j} = \theta(n^{k+2-j}),$$

and $\dfrac{n^{k+1}}{k+1} = \theta(n^{k+1})$. Hence, as $\max\{n^{k+1}, \ldots, n\} = n^{k+1}$, Proposition 9.9, 2, enables us to conclude

$$\sum_{i=1}^{n-1} i^k = \theta(n^{k+1}) \ ,$$

which proves the inductive step.

EXERCISE 9.3 Let $g_1, g_2$ be two mappings from $\mathbb{N}$ into $\mathbb{R}^+$; does the following equality hold?

$$O(g_1(n) + g_2(n)) = g_1(n) + O(g_2(n)). \qquad \diamondsuit$$

REMARK 9.11 The relation $f\,\mathcal{R}\,g$ if and only if $f = O(g)$ is a preorder relation, i.e. a reflexive and transitive relation. It is naturally associated with an order relation: let $E$ be the set of mappings from $\mathbb{N}$ into $\mathbb{R}^+$, the relation $f \equiv g$ if and only if $f = \theta(g)$ is an equivalence relation, called the equivalence associated with $\mathcal{R}$; on the factor set $E/\equiv$, the relation $[\mathcal{R}/\equiv]$ defined by $[f]\,[\mathcal{R}/\equiv]\,[g]$ if and only if $f\,\mathcal{R}\,g$, is an order relation, which is the order relation canonically associated with $\mathcal{R}$ (see Proposition 2.9).

## 9.2 Criteria of asymptotic behaviour of functions

### 9.2.1 A sufficient condition

Given, now, two positive functions $f$ and $g$, we will compare them to find out whether $f$ is $O$, $\Omega$ or $\theta$ of function $g$ or not. To this end, assuming $f$ and $g$ are not zero, we will form the quotients $f/g$ or $g/f$.

**Proposition 9.12** *Let $f$ and $g$ be two positive functions, then:*

(i) $\displaystyle\lim_{n\to\infty}\frac{f(n)}{g(n)} = a \neq 0 \quad\Longrightarrow\quad f = \theta(g)$,

(ii) $\displaystyle\lim_{n\to\infty}\frac{f(n)}{g(n)} = 0 \quad\Longrightarrow\quad f = O(g)$ *and* $f \neq \theta(g)$,

(iii) $\displaystyle\lim_{n\to\infty}\frac{f(n)}{g(n)} = \infty \quad\Longrightarrow\quad g = O(f)$ *and* $g \neq \theta(f)$.

*Proof.* Let us prove (i) for instance. $\displaystyle\lim_{n\to\infty}\frac{f(n)}{g(n)} = a \neq 0$ implies

$$\forall\varepsilon\,,\ \exists n_0\,,\quad \forall n > n_0\ :\ \left|\frac{f(n)}{g(n)} - a\right| < \varepsilon\,.$$

Since $f$ and $g$ are positive, this in turn implies

$$(a-\varepsilon)g(n) < f(n) < (a+\varepsilon)g(n),$$

and thus $f(n) < (a+\varepsilon)g(n)$, $g(n) < \dfrac{1}{a-\varepsilon}f(n)$, hence $f = O(g)$ and $g = O(f)$. □

EXERCISE 9.4 Prove cases (ii) and (iii). ♢

**Definition 9.13** *If* $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$, *$f$ is said to be asymptotic to $g$ and we write $f \sim g$ ; $\sim$ is an equivalence relation.*

*Proof.* It can easily be checked that $\sim$ is an equivalence relation. Let us check for instance that $f \sim g$ and $g \sim h \implies f \sim h$: as $h$ and $g$ are not zero, we have

$$\lim_{n\to\infty} \frac{f(n)}{h(n)} = \lim_{n\to\infty} \frac{f(n)}{g(n)} \times \frac{g(n)}{h(n)} = 1.$$

□

**Proposition 9.14** *If* $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, *then $f$ grows more slowly than $g$, i.e. $f = o(g)$.*

*Proof.* Straightforward. □

Remark 9.15 1. The preceding proposition gives a criterion which is sufficient but not necessary to classify the functions among $O$ and $\Omega$: for instance, let $g(n) = 2n$ and

$$f(n) = \begin{cases} n & \text{if } n \text{ is odd,} \\ 2n & \text{if } n \text{ is even,} \end{cases}$$

then $\frac{f(n)}{g(n)}$ has no limit, even though $f = \theta(g)$ since

$$\forall n, \qquad \frac{g(n)}{2} \leq f(n) \leq g(n).$$

2. If $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ is one of the indeterminate forms $\frac{0}{0}$ or $\frac{\infty}{\infty}$, we can apply l'Hospital's rule in the form: if $\lim_{n\to\infty} \frac{f'(n)}{g'(n)} = a$ exists, then $\lim_{n\to\infty} \frac{f(n)}{g(n)} = a$.

Example 9.16

$$\forall a > 0, \forall b > 0\ , \quad \lim_{n\to\infty} \frac{\log(n)^a}{n^b} = 0$$

and

$$\forall a > 1, \forall b > 0\ , \quad \lim_{n\to\infty} \frac{n^b}{a^n} = 0\ ,$$

which is abbreviated to 'exponential wins over powers, and powers win over logarithms'. More precisely, for all $\varepsilon$ and $c$ such that $0 < \varepsilon < 1 < c$, the following holds:

$$1 \prec \log n \prec n^{\varepsilon} \prec n^{c} \prec n^{\log n} \prec c^{n} \prec n^{n} \prec c^{c^{n}}.$$

EXAMPLE 9.17 Let the partition recurrence $u_n = 2u_{n/2}$ with $u_1 = 1$. We have seen in Chapter 7 that $u_n$ is not uniquely determined on $\mathbb{N}$ (see Example 7.12). We can, however, determine the asymptotic order of magnitude of $u_n$: here we will obtain $u_n = \theta(n)$. If we assume that $u_n$ is ultimately increasing, namely, that there is an integer $n_0$ such that

$$\forall n \geq n_0, \forall p \geq n_0, \quad n \geq p \Longrightarrow u_n \geq u_p\,,$$

then $u_n$ is of the same order of magnitude as $n$. Indeed, in this case, for a large enough $k$, $2^k \geq n_0$. Thus let $n$ be such that $\exists k, \quad n_0 \leq 2^k \leq n \leq 2^{k+1}$; we deduce $2^k \leq u_n \leq 2^{k+1}$. We thus have

- on the one hand, $u_n \leq 2^{k+1}$ and $2^k \leq n$, hence $u_n \leq 2n$,
- on the other hand, $2^k \leq u_n$ and $n \leq 2^{k+1}$, hence $n \leq 2u_n$,

and thus, finally, $n/2 \leq u_n \leq 2n$, i.e. $u_n = \theta(n)$.

More generally, we can state the following proposition.

**Proposition 9.18** *Let $u_n = bu_{n/a} + cn^k$, with $a \geq 2, b, c > 0$, $k \in \mathbb{N}$. Assume, moreover, that $u_n$ is monotone increasing for $n \geq n_0$, then*

$$u_n = \begin{cases} \theta(n^k) & \text{if } b < a^k, \\ \theta(n^k \log n) & \text{if } b = a^k, \\ \theta(n^{\log_a b}) & \text{if } b > a^k. \end{cases}$$

The proof is not given here.

EXERCISE 9.5 Let the sequence $u_n$ be defined by: $u_o = c > 1$, and

$$u_n = u_{n-1} + \frac{1}{u_{n-1}}, \quad \text{for } n \geq 1\ .$$

1. Show that
$$\frac{1}{u_{n-1}^2} \leq \frac{1}{u_{n-1}}\,.$$

2. Show that
$$2 \leq u_n^2 - u_{n-1}^2 \leq 2 + u_n - u_{n-1}\,. \tag{9.1}$$

3. Determine the asymptotic behaviour of $u_n$. ♢

### 9.2.2 Hierarchies

A *hierarchy* is a set of functions against which all other functions are measured in order to study their asymptotic behaviour. For instance, $\{n^p \ / \ p \in \mathbb{N}\}$ is such a hierarchy. The formal definition follows.

**Definition 9.19** *A hierarchy $E$ is a set of functions:*

$$E = \{g\colon \mathbb{N} \longrightarrow \mathbb{R}\}$$

*such that:*

(i) $\forall g \in E$, $\lim_{n\to\infty} g(n) = 0$, *or* $\lim_{n\to\infty} g(n) = \infty$, *or $g$ is the constant function 1,*
(ii) $\forall g_1 \in E, \forall g_2 \in E$, *if $g_1 \neq g_2$ then either $g_1 = o(g_2)$, or $g_2 = o(g_1)$,*
(iii) $\forall g \in E$, $\forall n \in \mathbb{N}$, $g(n) > 0$.

The second condition of this definition asserts that two functions in the hierarchy never have the same order of magnitude.

EXAMPLE 9.20 $E = \{g \ / \ g(n) = n^a(\log n)^b,\ a, b \in \mathbb{R}\}$ is a hierarchy. Let $g_i = n^{a_i}(\log n)^{b_i},\ i = 1, 2$; we have $g_1 = o(g_2)$, or $g_1 \prec g_2$ if and only if $a_1b_1 < a_2b_2$ in the lexicographic ordering.

EXERCISE 9.6 Study the hierarchy $E = \{g \ / \ g(n) = e^{an^b} \times n^c \times (\log n)^d$, with $a, b, c, d \in \mathbb{R}$, $b > 0\}$. ♢

### 9.2.3 Asymptotic approximations

**Definition 9.21** *Let $E$ be a hierarchy and $f$ a function:*

- *If there exist $0 \neq a_1 \in \mathbb{R}$ and $g_1 \in E$ such that $f \sim a_1g_1$, then we have $f - a_1g_1 = o(g_1)$; we say that $g_1$ is the principal part of $f$ with respect to hierarchy $E$ and we write $f = a_1g_1 + o(g_1)$.*
- *More generally, the asymptotic power series expansion of order $k$ of $f$ with respect to $E$ is an expression of the form $f = a_1g_1 + \cdots + a_kg_k + o(g_k)$, with $g_{i+1} = o(g_i)$ for $i - 1, \ldots, k - 1$.*

**Lemma 9.22** *The principal part, if it exists, is unique.*

*Proof.* See Exercise 9.7. □

EXERCISE 9.7 Prove that, when it exists, the asymptotic power series expansion of order $k$ of $f$ is unique. ♢

EXAMPLE 9.23 The asymptotic power series expansion may not exist for various reasons:

- The hierarchy is not refined enough: for instance $e^n$ has no principal part with respect to $E = \{n^p \ / \ p \in \mathbb{N}\}$.
- The function has an irregular behaviour: for instance the function $f$ defined by

$$f(n) = \begin{cases} 2n & \text{if } n \text{ odd} \\ n & \text{if } n \text{ even} \end{cases}$$

has no limit with respect to $E$.

The following asymptotic power series expansions, with respect to the hierarchy

$$\{x^k \ / \ k \in \mathbb{N}\}\,,$$

and for $x \to 0$, are quite usual and useful:

for $|x| \le r < 1$,

$$\frac{1}{1-x} = 1 + x + \cdots + x^k + o(x^k),$$

for $\alpha \in \mathbb{R}$, $|x| \le r < 1$,

$$(1+x)^\alpha = 1 + \alpha x + \cdots + \alpha(\alpha-1)\cdots(\alpha-k+1)\frac{x^k}{k!} + o(x^k),$$

for $|x| \le r \in \mathbb{R}^+$,

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^k}{k!} + o(x^k),$$

for $|x| \le r < 1$,

$$\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \cdots + (-1)^{k+1}\frac{x^k}{k} + o(x^k).$$

The rules given in Proposition 9.24 complete those given in Proposition 9.8 and can be deduced from the asymptotic power series expansion given above.

**Proposition 9.24**

*if* $f(n) = o(1)$, $\qquad \log\big(1 + O(f(n))\big) = O\big(f(n)\big)$,

*if* $f(n) = O(1)$, $\qquad e^{O(f(n))} = 1 + O\big(f(n)\big)$,

*if* $f(n) = o(1)$ *and* $f(n)g(n) = O(1)$,

$$\big(1 + O(f(n))\big)^{O(g(n))} = 1 + O\big(f(n)g(n)\big).$$

*Proof.* Let $g = O(f)$, then $\exists c \in \mathbb{R}^+$, $\exists n_0 \in \mathbb{N}$, $\varepsilon \in \mathbb{R}^+$,

$$\forall n > n_0\,, \qquad |g(n)| \le c|f(n)| \le \varepsilon < 1\,.$$

The series

$$\log\big(1 + O(g(n))\big) = g(n)\bigg(1 - \frac{1}{2}g(n) + \frac{1}{3}g(n)^2 - \cdots\bigg)$$

thus converges $\forall n > n_0$, and the series $1 - \frac{1}{2}g(n) + \frac{1}{3}g(n)^2 - \cdots$ has the upper bound $1 + \frac{1}{2}\varepsilon + \frac{1}{3}\varepsilon^2 + \cdots$, whence the first rule. The second is proved similarly. The two first rules together imply the third rule. □

Exercise 9.8 Verify that

$$(1+n)^{1/n} = 1 + \frac{\log n}{n} + \frac{1}{2}\frac{(\log n)^2}{n^2} + \frac{1}{n^2} + \frac{1}{6}\frac{(\log n)^3}{n^3} + o\bigg(\frac{(\log n)^3}{n^3}\bigg). \quad \diamondsuit$$

### 9.2.4 Asymptotic approximations by partial sums

The following lemma, is very simple, but nevertheless very useful in many cases.

**Lemma 9.25**

1. *Let $f$ be a monotone decreasing mapping, and $u_n = \sum_{i=p}^{n} f(i)$, then,*

$$\int_p^{q+1} f(x)dx \le \sum_{i=p}^{q} f(i) \le \int_{p-1}^{q} f(x)dx.$$

2. *Similarly let $f$ be a monotone increasing mapping, then,*

$$\int_{p-1}^{q} f(x)dx \le \sum_{i=p}^{q} f(i) \le \int_p^{q+1} f(x)dx.$$

Figure 9.1 is an aid to the understanding of case 2.
We have already seen an application of the squeeze obtained in Lemma 9.25, 1. If $f(x) = \dfrac{1}{x}$, then

$$\int_2^{n+1} \frac{dx}{x} \le \sum_{i=2}^{n} \frac{1}{i} \le \int_1^{n} \frac{dx}{x},$$

which enables us to bound $H_n = 1 + \dfrac{1}{2} + \cdots + \dfrac{1}{n}$:

$$\log(n+1) + 1 - \log 2 \le H_n \le \log(n) + 1.$$

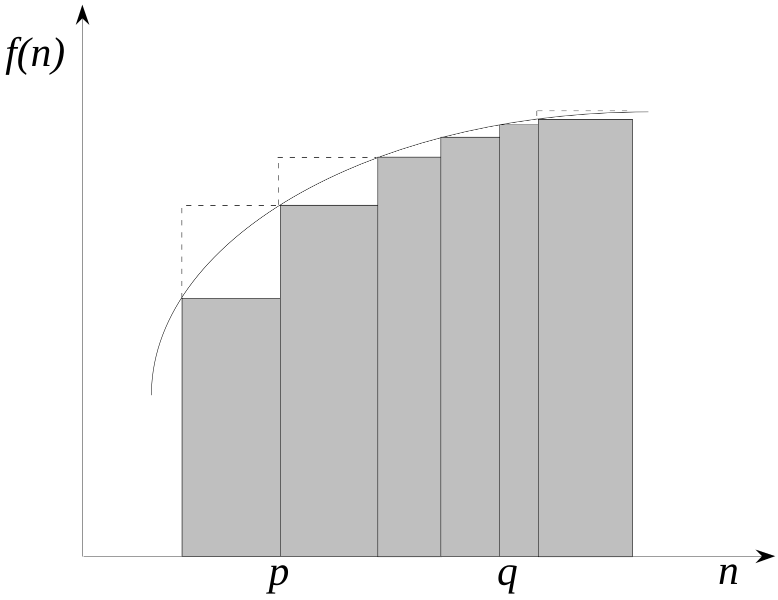


Figure 9.1 A monotone increasing mapping $f$.

We can prove, at the cost of more complex computations, that

$$H_n = \log n + \gamma + \frac{1}{2n} + o(n),$$

where $\gamma = 0.5772\ldots$ is Euler's constant.

We now apply Lemma 9.25, 2, to find the order of magnitude of $n!$: we reduce this problem to evaluating the order of magnitude of $\log n!$. If $f(x) = \log x$, we have $\log n! = \sum_{i=1}^{n} \log i = \sum_{i=2}^{n} \log i$ and, by the above remark,

$$\int_1^n \log x dx \leq \sum_{i=2}^{n} \log i = \sum_{i=1}^{n} \log i \leq \int_1^{n+1} \log x dx.$$

Integrating by parts $\int_1^n \log x dx = [x \log x]_1^n - \int_1^n dx = n \log n - (n-1)$, whence the bounds

$$n \log n - n + 1 \leq \log n! \leq (n+1)\log(n+1) - n,$$

which will enable us to find an approximation of $n!$ . Forming the exponentials we have: $e^{1-n} n^n \leq n! \leq e^{-n}(n+1)^{n+1}$. The order of magnitude of $n!$ is thus $O(n^{n+1})$. A more precise asymptotic power series expansion of $n!$ can be obtained, yielding an exact equivalent of $n!$, but the proof is more complex and gives *Stirling's formula*

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \frac{1}{12n} + o\left(\frac{1}{n}\right)\right),$$

whence $n! = \theta(n^{n+1/2})$.

### 9.2.5 'Bootstrapping'

We now introduce a technique which is both simple and powerful for finding asymptotic expansions of functions or solving recurrences asymptotically, without explicit computation but by using successive approximations. We start with a rough estimate, which we improve by plugging the estimate in the recurrence; this process gives better and better estimates, and is stopped when a reasonably good estimate is obtained (sometimes the exact solution can even be found, see Exercise 9.10). This technique is called *bootstrapping*.

We illustrate this technique with an example taken from Graham–Knuth–Patashnik. Consider the exponential generating series defined by

$$u(z) = \sum_{n\geq 0} z^n = e^{v(z)}, \quad \text{with} \quad v(z) = \sum_{k\geq 1} \frac{z^k}{k^2}, \tag{9.2}$$

and try to find the asymptotic behaviour of the coefficients $u_n$ of the series $u(z)$. Differentiating equation (9.2), we find

$$u'(z) = \sum_{n\geq 0} nz^{n-1} = v'(z)e^{v(z)} = \sum_{k\geq 1} \frac{z^{k-1}}{k} u(z),$$

hence the recurrence equation defining $u_n$, $u_0 = 1$, and for $n \geq 1$,

$$nu_n = \sum_{0\leq k<n} \frac{u_k}{n-k}. \tag{9.3}$$

We check by induction on $n$ that $0 < u_n \leq 1$, since $u_0 = 1$, and $nu_n \leq \sum_{0\leq k<n} 1 = n$. Hence,

$$u_n = O(1).$$

This fact is used to 'start up the pump' of the bootstrapping, by plugging this information in equation (9.3); this gives

$$nu_n = \sum_{0\leq k<n} \frac{O(1)}{n-k} = H_n O(1) = O(\log n),$$

hence

$$u_n = O\Big(\frac{\log n}{n}\Big), \qquad \text{for } n > 1,$$

and

$$u_n = O\Big(\frac{1+\log n}{n}\Big), \qquad \text{for } n \geq 1.$$

Bootstrapping again in equation (9.3) gives, for $n > 1$,

$$nu_n = \frac{1}{n} + \sum_{0<k<n} \frac{1}{n-k} O\Big(\frac{1+\log k}{k}\Big),$$

since, for $1 \leq k < n$, $O(1+\log k) = O(\log n)$,

$$\begin{aligned} nu_n &= \frac{1}{n} + \sum_{0<k<n} \frac{O(\log n)}{k(n-k)} = \frac{1}{n} + \sum_{0<k<n} \Big(\frac{1}{k} + \frac{1}{n-k}\Big)\frac{O(\log n)}{n} \\ &= \frac{1}{n} + \frac{2}{n} H_{n-1} O(\log n) = \frac{1}{n} O((\log n)^2) \end{aligned}$$

and, finally,

$$u_n = O\Big(\Big(\frac{\log n}{n}\Big)^2\Big), \qquad \text{for } n > 1. \tag{9.4}$$

EXERCISE 9.9
1. What would come out of one more bootstrapping step using equation (9.4)?
2. Find the principal part of $u_n$. ♢

EXERCISE 9.10 Solve the recurrence $u_n = n + u_{n-1}$, with $u_0 = 0$, by bootstrapping. ♢