ALGORITHMIC COMPLEXITY OF TERM REWRITING SYSTEMS

C. CHOPPY, S. KAPLAN, M. SORIA
Laboratoire de Recherche en Informatique
U.A. C.N.R.S. 410
Université Paris-Sud, Bât 490
F-91405 ORSAY Cédex, FRANCE
net : mcvax!inria!lri!cc, mcvax!inria!lri!kaplan, mcvax!inria!lri!soria

Introduction

Algebraic specifications are now widely used for data structuring and they turn out to be quite useful for various aspects of program development, such as prototyping, assisted program construction, proving properties, etc [BCV 85, FG 84, FGJM 84, GHW 85, Kap 86]. Some of these applications require to add a notion of computation to algebraic specifications, for instance by providing a (convergent) rewrite rule system that expresses the properties of the operators. In this context, it may be of first interest to define a notion of algorithmic complexity for an algebraic specification, or, more precisely, a notion of complexity for each operator defined in the specification. Computing operator complexity within a given specification helps understanding how evaluation costs are distributed ; it may point out "costly" operators, and motivate the search for an equivalent, but "cheaper", specification.

In [CLR 80], the cost of a term is defined as the number of rewriting steps for reducing it to its normal form, and the cost of an operator is defined as the general cost of a term obtained by applying this operator to terms in normal form. In this paper, we further formalize this notion of operator complexity and investigate its computation through analysis methods developed for instance in [Stey 84] and [Fla 87]. We show how these methods apply to the computation of the enumerative series over the terms of an algebraic specification. We define the notion of *regular* rewriting systems, and consider cost series of operators that are described by such systems. We show how these analysis methods apply to compute such costs and provide an asymptotic evaluation of the average cost of an operator. Our results allow costs to be computed without any explicit manipulation of series. We provide an eventual user with ready-to-use formulae, where the different parameters only depend on the "geometry" of the system, e.g. the number of constructors in the left handside of rules, number of occurrences of a derived operator in the right handside, etc.

Quantitative evaluation of rewriting systems had not yet been studied under such an approach (except in [CLR 80]), to our knowledge. From a different point of view, complexity of algebraic implementations has been studied in [BBWT 81, E&M 81, etc.] w.r.t. computability issues.

1. Introductory example

Let us assume that one wants to evaluate the average cost of a given computation on some data set. The data belong to a set of objects, a size can be computed for each object ; let D_n denote the set of objects of size n, and N_n = card (D_n). Assuming all the objects have the same probability, the average cost is [GSF 86] :

$$\overline{C}_n = \frac{1}{N_n} \sum_{d \in D_n} \mathrm{cost}(d) = \frac{C_n}{N_n} .$$

Generating series are defined by associating the series $a(z) = \sum a_n z^n$ to a sequence (a_n) : to the sequence (N_n) is associated the *enumerative series* $N(z) = \sum N_n z^n$, and to the sequence (C_n) is associated the cost series $C(z) = \sum C_n z^n$. Computation of the coefficients of the generating series can be performed either using "exact" methods (e.g. using the Lagrange inversion theorem) or methods that provide an approximation, based on real or complex analysis techniques. The asymptotic value of the coefficients a_n of a complex series $\sum a_n z^n$ may be evaluated using results of complex functions theory (essentially based on Cauchy formula) : the singularity closest to the origin determines the order of growth of the coefficients (more precisely, their exponential factor is determined by convergence radius of the series and their polynomial factor is function of the nature of the singularity) [Fla 87].

Consider the example of a specification of binary trees with two constructors : the constant 'a' and the '_ . _' operator (think of "cons" in Lisp).

Considering the enumerative series : $N_{tree}(z) = \sum_{n \geq 0} N_n z^n$, since each nth power of z appears as many times as there are trees of size n, we have : $N_{tree}(z) = \sum_{t \in T_{tree}} z^{|t|}$ where T_{tree} is the set of terms built with the constructors 'a' and '_ . _' .

Let us first perform the computation of $N_{tree}(z)$ by case analysis on terms :

$$N_{tree}(z) = \sum_{t = a} z^{|t|} + \sum_{t = t_1.t_2} z^{|t_1.t_2|} .$$

Since $|t_1 . t_2| = 1 + |t_1| + |t_2|$: $N_{tree}(z) = z + z \sum_{t_1 \in T_{tree}} \sum_{t_2 \in T_{tree}} z^{|t_1|} z^{|t_2|} = z\,(1 + N_{tree}^2(z))$.

In the general case, computation of $N_{tree}(z)$ can also be performed using systematic methods for computing enumerative series and cost series for algorithms on combinatorial structures [Fla 87] (in particular, one may apply these methods when trees are used as data structures to represent terms) ; the main steps of these methods are the following :

• take the construction primitives of the combinatorial object and deduce the structural equations ; in the case of tree this leads to :

$$\text{tree} = a + \begin{array}{c} \cdot \\ / \quad \backslash \\ \text{tree} \quad \text{tree} \end{array} \qquad \text{or} \qquad \text{tree} = a + . \times \text{tree} \times \text{tree}$$

where + is the disjoint union and × the cartesian product

• transpose the structural equation(s) to generating series, using the fact that, when considering the associated series, any disjoint union is expressed by a sum and any cartesian product is expressed by a product ; in our example, this leads to :

$$N_{tree}(z) = z\,(1 + N_{tree}^2(z))$$

(the same result was obtained above by case analysis)

- solve the generating series equations. In the above example, simple resolution leads to : $N_{tree}(z) = \frac{1-\sqrt{1-4z^2}}{2z}$ (this solution is adequate since it is analytic at the origin), and further computation by series development leads $N_{2p} = 0$ and to $N_{2p+1} = \frac{1}{p+1}\binom{2p}{p}$ (Catalan numbers, cf. [Knu 73]).

Using the Stirling formula : $p! = \sqrt{2\pi p}\left[\frac{p}{e}\right]^p (1 + O(\frac{1}{p}))$, this yields :

$$N_{2p+1} = \frac{1}{\sqrt{\pi}} p^{-3/2} 2^{2p} \left(1 + O\left(\frac{1}{p}\right)\right)$$

Another approach is to use complex analysis methods (local analysis around singularities) for passing from functional equations over generating functions to asymptotic expressions of their coefficients. Continuing with our example, we have : $N_{tree}(z) = -\frac{\sqrt{1-2z}\sqrt{1+2z}}{2z} + \frac{1}{2z}$

$N_{tree}(z)$ is analytic for $|z| < \frac{1}{2}$ and has two singularities : $z = \frac{1}{2}$ and $z = -\frac{1}{2}$.

Let us apply the Newton expansion : $[z^n]\,(1-kz)^{\alpha}$ † $= k^n(-1)^n \binom{n}{\alpha} = k^n \binom{n-\alpha-1}{-\alpha-1}$

One shows [Fla 87] that when $n \to +\infty$: $[z^n]\,(1-kz)^{\alpha} = \frac{n^{-\alpha-1}}{\Gamma(-\alpha)} k^n \left(1 + O\left(\frac{1}{n}\right)\right)$

(where Γ is the Euler Gamma function : $\Gamma(x) = \int_0^{+\infty} e^{-t} t^{x-1} dt$. We recall that : $\Gamma(x+1) = x\,\Gamma(x)$, $\Gamma(1) = 1$ -- hence $\Gamma(n) = (n-1)!$ when $n \in N$ -- and $\Gamma(\frac{1}{2}) = \sqrt{\pi}$).

The contributions of singularities with same module are added together, leading in our case to :

$$[z^n]\,N(z) = \frac{1}{\sqrt{2\pi}}\, 2^{n+1}\, n^{-3/2} \left(1 + O\left(\frac{1}{n}\right)\right) \text{ for n odd, and 0 for n even.}$$

A more systematic approach that uses "transfer lemmas" [Fla 87] is presented in section 2.

Now let us add a derived operator '_ ↑ _' (called "shuffle" since it is moving subtrees around) defined with the following rules :

(S_1) : $a \uparrow t \to a\,.\,t$, $\quad (t_1\,.\,t_2) \uparrow a \to (t_1\,.\,t_2)\,.\,a$, $\quad (t_1\,.\,t_2) \uparrow (u_1\,.\,u_2) \to (t_1 \uparrow u_1)\,.\,(t_2 \uparrow u_2)$

The cost of a term is the number of rewriting steps necessary to reduce it to its normal form for a given strategy ; the cost of an operator is defined as the cost of a term obtained by applying this operator to the terms in normal form. In all examples considered in this paper, it is insured that operator costs are independent from the evaluation strategy (cf. propsition in section 3).

Let us evaluate the cost functions for the operators of this specification : 'a' and '_ . _' being constructors, the corresponding cost functions are equal to zero. Computation of the cost function for the operation '_ ↑ _' will be done by means of the following generating function (cf. definition 3.4) :

$C^{\uparrow}(z) = \sum_{n\geq 0} C^{\uparrow}_n z^n$ where $C^{\uparrow}_n = \sum_{\substack{t,\,u \in T_{tree} \\ |t| + |u| = n}} \text{cost}(t \uparrow u)$ (where $|t|$ is the size of the term t, i.e. the

† $[z^n]\,\Phi(z)$ denotes the coefficient of z^n in $\Phi(z)$

total number of symbols that appear in t, and where T_{tree} is the set of terms in normal form, which is exactly here the set of terms built on the constructors 'a' and '_ . _').

Hence : $C^{\uparrow}(z) = \sum_{t,\, u \in T_{tree}} cost(t \uparrow u)\, z^{|t|+|u|}$

Let us use a case analysis for terms in normal form to compute $C^{\uparrow}(z)$:

$$C^{\uparrow}(z) = \sum_{t \in T_{tree}} cost\,(a \uparrow t)\, z^{1+|t|} + \sum_{t_1,t_2 \in T_{tree}} cost\,((t_1.t_2) \uparrow a)\, z^{|t_1.t_2|+1}$$
$$+ \sum_{t_1,t_2,u_1,u_2 \in T_{tree}} cost\,((t_1\,.\,t_2) \uparrow (u_1\,.\,u_2))\, z^{|t_1.t_2|+|u_1.u_2|}$$

since :
$$cost\,(a \uparrow t) = 1 + cost\,(a\,.\,t) = 1,$$
$$cost\,((t_1.t_2) \uparrow a) = 1 + cost\,((t_1.t_2)\,.\,a) = 1,$$
$$cost\,((t_1\,.\,t_2) \uparrow (u_1\,.\,u_2)) = 1 + cost\,((t_1 \uparrow u_1).(t_2 \uparrow u_2))$$
$$= 1 + cost\,(t_1 \uparrow u_1) + cost\,(t_2 \uparrow u_2)\,,$$

we have :

$$C^{\uparrow}(z) = z \sum_{t \in T_{tree}} z^{|t|} + z \sum_{t_1 \in T_{tree}} z^{|t_1|} \sum_{t_2 \in T_{tree}} z^{|t_2|} + z^2 \sum_{t_1 \in T_{tree}} z^{|t_1|} \sum_{t_2 \in T_{tree}} z^{|t_2|} \sum_{u_1 \in T_{tree}} z^{|u_1|} \sum_{u_2 \in T_{tree}} z^{|u_2|}$$
$$+ z^2 \sum_{t_1,u_1 \in T_{tree}} cost\,(t_1 \uparrow u_1)\, z^{|t_1|} z^{|u_1|} \sum_{t_2 \in T_{tree}} z^{|t_2|} \sum_{u_2 \in T_{tree}} z^{|u_2|}$$
$$+ z^2 \sum_{t_2,u_2 \in T_{tree}} cost\,(t_2 \uparrow u_2)\, z^{|t_2|} z^{|u_2|} \sum_{t_1 \in T_{tree}} z^{|t_1|} \sum_{u_1 \in T_{tree}} z^{|u_1|}$$
$$= z\, N_{tree}(z) + z^2\, N^2_{tree}(z) + z^2\, N^4_{tree}(z) + 2\, z^2\, N^2_{tree}(z)\, C^{\uparrow}(z)\,.$$

Hence :

$$C^{\uparrow}(z) = \frac{z\, N_{tree}(z) + z^2\, N^2_{tree}(z) + z^2\, N^4_{tree}(z)}{1 - 2\, z^2\, N^2_{tree}(z)} = \frac{z\, N_{tree}(z) + z\, N^3_{tree}(z)}{1 - 2\, z^2\, N^2_{tree}(z)} = \frac{N^2_{tree}(z)}{1 - 2\, z^2\, N^2_{tree}(z)} \;\dagger\dagger$$

Replacing in this expression $N_{tree}(z)$ by its value : $\dfrac{1-\sqrt{1-4z^2}}{2z}$ and using the same complex analysis method as for $N_{tree}(z)$ (development around singularities, Newton expansion, adding up the contributions of singularities) yields to : $C^{\uparrow}_{2p} = \dfrac{4}{\sqrt{\pi}} 2^{2p} p^{-3/2}(1 + O\,(\dfrac{1}{p}))$

Now the average cost is : $\overline{C}^{\uparrow}_{2p} = \dfrac{C^{\uparrow}_{2p}}{N_{2,2p}}$ where $N_{2,2p}$ (cf. section 2) is the number of tree couples (t_1, t_2) such that $|t_1| + |t_2| = 2p$. Computation of $N_{2,2p}$ yields :

$$C^{\uparrow}_{2p} = 4(1 + O\,(\frac{1}{p}))$$

in this case, the average cost is **constant** (asymptotically).

We develop in section 3, a general method for computing the coefficients of series such as $C^{\uparrow}(z)$, when the rules defining '_ $\uparrow$ _' constitute a *regular* rewriting system.

$\dagger\dagger$ using the fact that : $N_{tree}(z) = z\,(1 + N^2_{tree}(z))$

2. Enumerative series of the term algebra

We denote by T_{Constr} the set of terms built on the signature Constr. Let us denote by α_k the number of symbols in Constr of arity k. We always suppose that $\alpha_0 \neq 0$.

Definition 2.1

Let N_n stand for the number of terms in T_{Constr} of size n. The *enumerative series* of T_{Constr} is :

$$N(z) \;=\; \sum_{n\geq 0} N_n\, z^n \;=\; \sum_{t\in T_{Constr}} z^{|t|}.$$

Let $\Phi(X)$ stand for the polynomial : $\Phi(X) = \sum_{k=0}^{p} \alpha_k X^k$, where 'p' denotes the largest arity in Constr, and where there exists $\alpha_k > 0$ with $k \geq 2$.†

We have the following results :

(1) N(z) is a solution of the functional equation : $N(z) \;=\; z\,.\,\Phi\,(N(z))$ (cf. section 1 : transposition from structural equations to generating series).

(2) Let us suppose that there is no polynomial Ψ and integer $d\geq 2$ such that $\Phi(X) = \Psi\,(X^d)$; let τ be the smallest root of the equation $\Phi(X) = X\Phi'(X)$, and $\rho = \dfrac{\tau}{\Phi(\tau)}$; ρ is the convergence radius of N(z) and $0 < \rho < 1$. τ is the only real positive root of the equation $\Phi(X) = X\Phi'(X)$ such that $|X| = \tau$. Moreover, $\tau = N(\rho)$ and ρ is the only singularity of N(z) such that $|z| = \rho$. Then around $z = \rho$ [M&M 78] :

$$N(z) = \tau - \sqrt{\frac{2\Phi(\tau)}{\Phi''(\tau)}}\left(1-\frac{z}{\rho}\right)^{1/2} + \gamma 1\left(1-\frac{z}{\rho}\right) + O\left(\left(1-\frac{z}{\rho}\right)^{3/2}\right) \qquad \text{(F1)}$$

(3) **Transfer Lemmas** : when it is possible to have an asymptotic development of a series around the singularity that is closest to the origin, under some conditions (that are always fulfilled in the case of term algebras) an estimation of the series coefficients can be deduced from the series estimation through transfer lemmas [Fla 87].

We use the transfer lemmas in the following case :

Let $f(z) = h(z) + O(g(z))$ be the expansion of f around the singularity ρ, where f and g are standard functions of the type $\left(1-\dfrac{z}{\rho}\right)^{\alpha}$ and h is of higher order than g around ρ. Then : $[z^n]\, f(z) = [z^n]\, h(z) + O\,([z^n]\, g(z))$.

Considering the expression (F1) and applying the transfer lemmas, one gets :

$$N_n = -\sqrt{\frac{2\Phi(\tau)}{\Phi''(\tau)}}\,\frac{\rho^{-n}\, n^{-3/2}}{\Gamma(-1/2)} + O\left(\rho^{-n} n^{-5/2}\right) \qquad \text{with } \Gamma\,(-1/2) = -\,2\,\sqrt{\pi}$$

i.e. finally :

$$N_n \;=\; \sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\;\rho^{-n}\, n^{-3/2}\;\left(1 + O\left(\frac{1}{n}\right)\right).$$

† If this condition is not satisfied, this means that $N(z) = z(1+\alpha_1 N)$, and thus : $N(z) = \dfrac{z}{1-\alpha_1 z}$

Note 1 :

For average cost computations, we need to evaluate the quantity $N_{m,n}$, that is the number of terms $t_1,\dots,t_m \in T_{Constr}$ such that $|t_1|+\dots+|t_m| = n$. Then :

$$\sum_{n=0}^{\infty} N_{m,n}\, z^n = \sum_{n=0}^{\infty} \sum_{\substack{t_1,\dots,t_m \in T_{Constr} \\ |t_1|+\dots+|t_m|=n}} z^n = \sum_{t_1,\dots,t_m \in T_{Constr}} z^{|t_1|+\dots+|t_m|} = (N(z))^m.$$

Thus, $N_{m,n}$ is the coefficient of z^n in $(N(z))^m$, and using (F1) we have :

$$(N(z))^m = \tau^m + m\,\tau^{m-1}\sqrt{\frac{2\Phi(\tau)}{\Phi''(\tau)}}\left[1-\frac{z}{\rho}\right]^{1/2} + \gamma 2\left[1-\frac{z}{\rho}\right] + O\left(\left[1-\frac{z}{\rho}\right]^{3/2}\right).$$

Now, using the transfer lemmas, we obtain :

$$N_{m,n} = m\,\tau^{m-1}\sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\,\rho^{-n}\,n^{-3/2}\,\left(1 + O\left(\frac{1}{n}\right)\right)$$

and : $N_{m,n} = m\,\tau^{m-1}\,N_n = \frac{\partial}{\partial N}(N^m(z))\,|_{z=\rho} N_n$

More generally, given the polynomial on $N(z)$: $P(z) = \sum_{i=1}^{q} \alpha_i N^i(z)$ similarly : $[z^n]\,P(z) = N_n \frac{\partial P}{\partial N}\,|_{z=\rho}$.

Note 2 :

The above results were obtained for the case where ρ was the only singularity of $N(z)$ such that $|z| = \rho$. Let us now consider the case where there are more than one such singularity. If $\Phi(X) = \Psi(X^d)$, let τ still denote the unique real, positive root of the equation $\Phi(X) = X\Phi'(X)$. The other solutions of this equation are $e^{2ik\pi/d}\tau$, for $k = 1,2,\dots,d-1$. We now have :

$$N_n = d\sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\,\rho^{-n}\,n^{-3/2}\,\left(1 + O\left(\frac{1}{n}\right)\right) \quad \text{if } n \equiv 1 \text{ [mod d]},$$

$$N_n = 0 \quad \text{otherwise.}$$

and :

$$N_{m,n} = d\,m\,\tau^{m-1}\sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\,\rho^{-n}\,n^{-3/2}\,\left(1 + O\left(\frac{1}{n}\right)\right) \quad \text{if } n \equiv m \text{ [mod d]},$$

$$N_{m,n} = 0 \quad \text{otherwise.}$$

Example :

These results immediately apply to the example of section 1 where Constr = {a, _ . _ }. We have $\alpha_0 = 1$, $\alpha_1 = 0$, $\alpha_2 = 1$. Then $\Phi(X) = 1+X^2$, of the form $\Psi(X^2)$, with $\Psi(Y) = 1+Y$. τ satisfies : $1+\tau^2 = 2\tau^2$. Thus $\tau = 1$ and $\rho = \frac{\tau}{\Phi(\tau)} = 1/2$. , and we derive :

$$N_n = \frac{1}{\sqrt{2\pi}}\,2^{n+1}\,n^{-3/2}\,\left(1 + O\left(\frac{1}{n}\right)\right) \text{ for n odd, and 0 for n even ;}$$

$$N_{m,n} = \frac{m}{\sqrt{2\pi}}\,2^{n+1}\,n^{-3/2}\,\left(1 + O\left(\frac{1}{n}\right)\right) \text{ for } n \equiv m \text{ [mod d], and 0 otherwise.}$$

Notice that letting $m = 2$, $n = 2p$, we obtain : $N_{2,2p} = \frac{1}{\sqrt{2\pi}}\,2^{2p}\,p^{-3/2}\left(1 + O\left(\frac{1}{p}\right)\right)$ which leads to the same result as obtained in section 1.

3. Cost series for the derived operators

We suppose that the set of operator symbols Σ is partitioned into :

- a set Constr of *constructors* (these are functions that generate the set of terms, and for which the generating function N was computed hereabove) ;
- a set Der of *derived operators* (that realize computations on terms of T_{Constr}).

We wish to ensure that any term of T_{Constr} (i.e. built with constructors only) is irreducible, and that for $f \in$ Der and for $t_1,\dots,t_n \in T_{Constr}$, $f(t_1,\dots,t_n)$ rewrites into a unique term of T_{Constr} in a finite amount of rewrite steps. To this effect, we are going to restrict the form of the rules that are acceptable.

Definition 3.1

A *Constr-enumeration* associated to an operator $f \in$ Der, with $ar(f)=n$, is a finite family of n-tuples $(\vec{\varpi}_e)_{1\le e\le D(f)}$ of $(T_{Constr}(X))^n$ such that :

- each $\vec{\varpi}_e$ contains at least a constructor symbol
- for any $\vec{t} \in (T_{Constr})^n$, there exists a unique substitution $\sigma : X \to T_{Constr}$ and a unique e such that : $\vec{t} = \vec{\varpi}_e\sigma$

Thus, intuitively speaking, the $(\vec{\varpi}_e)$ form a description of the possible arguments of f, that is at the same time exhaustive and non-ambiguous. For instance, with Constr={0, s} and Der={+}, we can take : $\vec{\varpi}_1 = (x , 0)$ and $\vec{\varpi}_2 = (x , s(y))$, as a Constr-enumeration of the symbol '+'.

Let X_e stand for the variables that appear in $\vec{\varpi}_e$, and ξ_e for the number of constructors that appear in $\vec{\varpi}_e$ (let us recall that, according to Definition 3.1, $\xi_e \neq 0$). In the previous example, we have : $X_1 = \{x\}$, $\xi_1 = 1$ and $X_2 = \{x,y\}$, $\xi_2 = 1$.

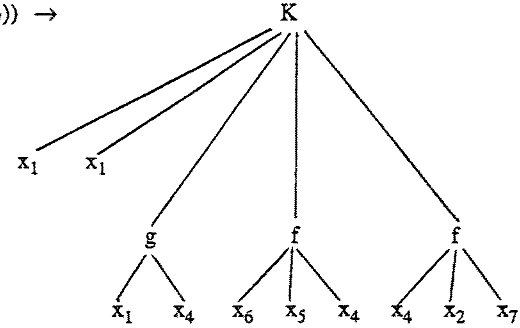
Definition 3.2

A *Constr-definition* of $f \in$ Der is a set of rewrite rules $R_f = (r_e)_{1\le e\le D(f)}$ such that :

- each rule is of the form $r_e : f(\vec{\varpi}_e) \to \rho_e$, where $(\vec{\varpi}_e)_{e\in D(f)}$ is a Constr-enumeration of f ;
- each ρ_e is of the form $\rho_e = K(x_1,\dots,x_n,\phi_1,\dots,\phi_m)$, where :
    - K is a context made of constructors only,
    - $\{x_1,\dots,x_n\} \subseteq X_e$,
    - each ϕ_k is of the form $g(y_1,\dots,y_{ar(g)})$, where g is a derived operator, $\{y_1,\dots,y_{ar(g)}\} \subseteq X_e$, and $y_i \neq y_j$ if $i \neq j$.

A typical rule thus looks like :

$f(\vec{\omega}(x_1,x_2,x_3,x_4,x_5,x_6,x_7)) \;\rightarrow$



Definition 3.3

A set of R of rewrite rules is *regular* if it is of the form $R = \cup_{f\in Der} R_f$, where R_f is a Constr-definition of f, for each $f \in Der$

For instance, the following systems are regular :

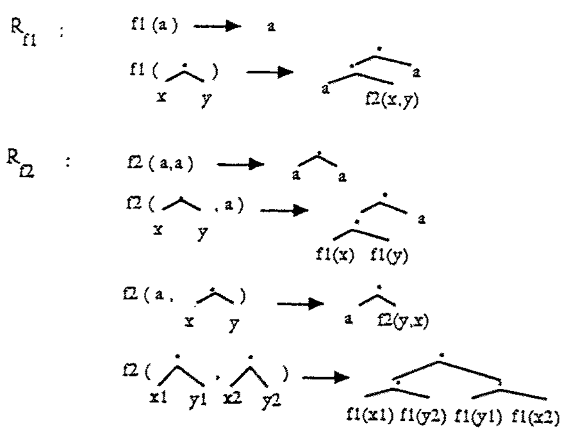
(S_2) Constr = {0, s}, Der = {+ , even} and

R_+ : $x + 0 \rightarrow x$, $x + s(y) \rightarrow s\,(x + y)$

R_{even} : $even(0) \rightarrow True$, $even(s(0)) \rightarrow False$, $even(s(s(x))) \rightarrow even(x)$

The Constr-enumeration associated to '+' is the one presented hereabove, while the one associated to 'even' is $\omega_1 = (0)$, $\omega_2 = (\ s(0)\ )$, $\omega_3 = (\ s(s(x))\ )$.

(S_3) Constr = {a, _ . _}, Der = {f_1(_), f_2(_ , _)} and



Thus, regular systems can be mutually recursive.

We have the following result :

Theorem 3.1

A regular system is confluent and nœtherian. Moreover, it provides sufficiently complete and hierarchically consistent definitions of the derived operators w.r.t. the constructors.

Proof : confluence is because a regular system has no critical pair. Let us now prove nœtherianity. We order $T_{Constr \cup Der}$ by the *recursive path ordering* $>_{rpo}$ (cf. [Der 82,85]) such that all the constructors are equivalent, all the derived operators are equivalent, and the derived operators are greater than the constructors. Then consider a rule :

$$f(\vec{\omega}_i) \quad \rightarrow \quad K(x_1,...,x_n,\phi_1,...,\phi_m)$$

with the previous notations. Each ϕ_k is of the form $g(y_1,...,y_{ar(g)})$. Since by hypothesis $\vec{\omega}_i$ is not empty, the *multiset* $\{\vec{\omega}_i\}$ is strictly greater (for the associated ordering $\gg_{rpo}$) than the multiset $\{y_1,...,y_{ar(g)}\}$. Thus, $f(\vec{\omega}_i) >_{rpo} \phi_k$ for any k, and finally :

$$f(\vec{\omega}_i) \quad >_{rpo} \quad K(x_1,...,x_n,\phi_1,...,\phi_m)$$

This ends the proof of the termination of a regular system.
We notice that terms in T_{Constr} are irreducible, since no left-hand side admits a constructor at the root occurrence. This implies hierarchical consistence w.r.t. the constructors. Conversely, a term in normal form contains no derived operator, since otherwise a rule would apply to further reduce it. Thus, the system is also hierarchically complete.

In this article, we restrict attention to regular systems only. We then have the following result :

Proposition :

Let R be a regular system. For any term $t = f(t_1,...,t_n)$ with $f \in Der$ and $t_1,...,t_n \in T_{Constr}$, the number of rewrite steps between t and its normal from is independent of the rewriting strategy.

Proof : Let us denote by Γ the set of terms of $T_{Constr \cup Der}$ such that *at most one* derived operator appears from any path inside the term to the root. It is clear that, if R is regular, if $t \in \Gamma$ and $t \rightarrow^* t'$, then $t' \in \Gamma$. We are going to show, more generally, that for any $t \in \Gamma$, the cost of t does not depend of the evaluation strategy (which proves the previous lemma). *Ad absurdum*, suppose that this is not the case for a given t. We can write $t = K[f_1(\vec{\chi_1}),...,f_n(\vec{\chi_n})]$, where K and the $\vec{\chi_i}$'s are made of constructors only, and the d_i's are derived operators. We consider two cases :

Case 1 : if K is empty (and $t = f_1(\vec{\chi_1})$), then exactly one rule applies to t. We let $\phi(t)$ be the term such that $t \rightarrow \phi(t)$. Then, necessarily, the cost of $\phi(t)$ depends on the evaluation strategy.

Case 2 : else, if the cost of each $f_i(\vec{\chi_i})$ is equal to m_i, *whatever* the evaluation strategy, then the cost of t would be $m_1+...+m_n$, whatever the strategy, which would contradict the hypothesis. Thus, there exists an i such that the cost of $f_i(\vec{\chi_i})$ depends on the evaluation strategy. We then let $\phi(t) = f_i(\vec{\chi_i})$.

We define the ordering '>' by $t > t'$ *iff* either $t \rightarrow t'$ or t' is a strict subterm of t. '>' is well-founded. Now, the infinite chain :

$$t, \ \phi(t), \ \phi(\phi(t)), \ ...$$

is decreasing for '>', which yields the desired contradiction. This terminates the proof.

Note : the property is not true for elements outside of Γ. Consider for instance the regular system :

$x \mid 0 \rightarrow 0, \qquad x \mid s(y) \rightarrow s(x \mid y)$

The term '(0 | 0) | 0' (which is not in Γ) would be normalized in respectively 1 or 2 steps by an "outermost" or an "innermost" strategy.

We now provide results about the complexity of the rewriting of the derived operators.

Definition 3.4

Let $f \in Der$ of arity m. Let C_n^f stand for sum the number of rewrite steps of the term $f(t_1,...,t_m)$ to its normal form, where the $(t_i)_{1\le i\le m}$ range over T_{Constr} and are such that $|t_1| + ... + |t_m| = n$:

$$C_n^f = \sum_{\substack{t_1,...,t_m \in T_{Constr} \\ |t_1|+...+|t_m|=n}} cost(f(t_1,...,t_m))$$

The *cost series associated to f* is :

$$C^f(z) = \sum_{n\ge 0} C_n^f z^n = \sum_{t_1,...,t_m \in T_{Constr}} cost(f(t_1,...,t_m))\, z^{|t_1|+...+|t_m|}$$

From now on, we suppose that $Der = \{f_1,...,f_{NDer}\}$ where NDer is the number of derived operators. Given a regular system, with each f_i defined by a Constr-definition, we can write :

$$C^{f_i}(z) = \sum_{t_1,...,t_{ar(f_i)} \in T_{Constr}} cost(f_i(t_1,...,t_{ar(f_i)})) \quad z^{|t_1|+\cdots+|t_{ar(f_i)}|} = \sum_{e\in D(f_i)} \sum_{\sigma} cost(f_i(\varpi_e^i\sigma))\, z^{|\varpi_e^i\sigma|}.$$

We have : $cost(f_i(\varpi_e^i\sigma)) = 1 + \sum_{1\le k\le n_e^i} cost(\phi_{k,i,e}\sigma)$, in accordance with definition 3.2. Thus :

$$C^{f_i}(z) = \underbrace{\sum_{e\in D(f_i)} \sum_{\sigma} z^{|\varpi_e^i\sigma|}}_{A} + \underbrace{\sum_{e\in D(f_i)} \sum_{\sigma} \sum_{1\le k\le n_e} cost(\phi_{k,i,e}\sigma)\, z^{|\varpi_e^i\sigma|}}_{B}.$$

<-------A-------> + <------------------B----------------->

Where $A = N^{ar(f_i)}(z)$ is a constant part of this sum, and B is a recursive part.

In order to simplify the B part of C^{f_i}, we first notice that it is actually quantified over e corresponding to the rules with non-constant right handsides, that we denote $D_{nc}(f_i)$. Let $X_{e,i}$ stand for the variables of ϖ_e^i. Then, we may write $\phi_{k,i,e} = f_j(y_1,...,y_{ar(f_j)})$, for a certain j, and

$$X_{e,i} - \{y_1,...,y_{ar(f_j)}\} = \{w_1,...,w_{l(i,j,e)}\}, \quad \text{with } l(i,j,e) = |X_{e,i}| - ar(f_j)$$

where the w_i's are the variables that appear in the left handsides and not in the right handsides of the rules.

Let $\xi_{e,i}$ stand for the number of constructors appearing in ϖ_e^i. We suppose that σ restricted to the variables $X_{e,i}$ of ϖ_e^i is : $\{y_1:= t_1,...,y_{ar(f_j)}:= t_{ar(f_j)},\ w_1:= t'_1,...,w_{l(i,j,e)}:= t'_{l(i,j,e)}\}$.

We have : $|\varpi_e^i\sigma| = \xi_{e,i} + |t_1|+...+|t_{ar(f_j)}| + |t'_1|+...+|t'_{l(i,j,e)}|$

Then :

$$\sum_{e\in D(f_i)} cost(\phi_{k,i,e}\sigma)\, z^{|\varpi_e^i\sigma|} = z^{\xi_{e,i}} \sum_{e\in D_{nc}(f_i)} cost(f_j(t_1,...,t_{ar(f_j)}))\, z^{|t_1|+\cdots+|t_{ar(f_j)}|}\, z^{|t'_1|+\cdots+|t'_{l(i,j,e)}|}.$$

Let $\varepsilon_{e,j}^i$ stand for the number of occurrences of 'f_j' in the right handside of the rule r_e^i. The B part of C^{f_i} finally rewrites into :

$$\sum_{e\in D_{nc}(f_i)} z^{\xi_{e,i}} \sum_{1\le j\le NDer} \varepsilon_{e,j}^i \sum_{\substack{t_1,...,t_{ar(f_j)} \in T_{Constr} \\ t'_1,...,t'_{l(i,j,e)} \in T_{Constr}}} cost(f_j(t_1,...,t_{ar(f_j)}))\, z^{|t_1|+\cdots+|t_{ar(f_j)}|}\, z^{|t'_1|+\cdots+|t'_{l(i,j,e)}|} =$$

$$\sum_{e\in D_{nc}(f_i)} \sum_{1\le j\le NDer} \varepsilon_{e,j}^i\, z^{\xi_{e,i}}\, N^{l(i,j,e)}\, C^{f_j}(z) = \sum_{e\in D_{nc}(f_i)} \sum_{1\le j\le m} \varepsilon_{e,j}^i\, z^{\xi_{e,i}}\, N^{|X_{e,i}|-ar(f_j)}\, C^{f_j}(z).$$

Then, let $\vec{C}(z)$ stand for the vector $\begin{bmatrix} C^{f_1}(z) \\ \dots \\ C^{f_{NDer}}(z) \end{bmatrix}$ and $\vec{Y}(z)$ for the vector $\begin{bmatrix} N^{ar(f_1)}(z) \\ \dots \\ N^{ar(f_{NDer})}(z) \end{bmatrix}$.

We also define $M_{i,j}(z) = \sum_{e \in D_{ac}(f_i)} \varepsilon^{i}_{e,j}\, z^{\xi_{e,i}}\, N^{|X_{e,i}|-ar(f_j)}(z)$, and we let $M(z)$ denote the matrix $(M_{i,j}(z))_{1\le i,j\le NDer}$. We obtain the central result of this paper :

Theorem 3.2
The cost series satisfy the equation : $\vec{C}(z) = M(z)\,\vec{C}(z) + \vec{Y}(z)$. Thus, the expression of each cost series is :

$$C^{f_i}(z) = \frac{\det(\,(Id-M)^{[i]}(z)\,)}{\det(\,Id-M(z)\,)},$$

where Id is the identity matrix, and $(Id-M)^{[i]}(z)$ is the matrix Id-M(z), the i^{th} column of which being replaced by $\vec{Y}(z)$.

Since $z = \dfrac{N(z)}{\Phi(N(z))}$, each $C^{f_i}(z)$ may be rewritten into the following form :

$$C^{f_i}(z) = \frac{P^i(N(z))}{Q^i(N(z))},$$

where P and Q are respectively prime polynomials with integer coefficients. Now, in order to evaluate the $C_n^{f_i}$, we have to determine the smallest singularity of each $C^{f_i}(z)$. Its singularities are :

- either the singularities of N(z), the smallest being for $z = \rho$,
- or the z's such that $Q^i(N(z)) = 0$. Let us denote by ρ_0^i the smallest real positive root of $Q^i(N(\rho_0^i)) = 0$ (with the convention that $\rho_0^i = \infty$ if the equation $Q^i(N(z))$ has no root for $|z| \le \rho_0^i$).

We now have the following main theorem :

Theorem 3.3

(3) if $\rho < \rho_0^i$, then : $\quad \overline{C}_n^{f_i} = k_1\ (1 + O(\frac{1}{n}))$

(2) if $\rho = \rho_0^i$, then: $\quad \overline{C}_n^{f_i} = k_2 n^{m_2/2}\ (1+O(\frac{1}{\sqrt{n}}))$

(3) if $\rho > \rho_0^i$, then : $\quad \overline{C}_n^{f_i} = k_3 \left[\frac{\rho}{\rho_0^i}\right]^n n^{m_3+1/2}\ (1+O(\frac{1}{\sqrt{n}}))$

The k_i's are real numbers, and the m_i's are strictly positive integers ; all of them can be expressed simply (as shown hereafter - Results 3.4, 3.5 and 3.6).

We now proceed by proving theorem 3.3 by considering successively the three cases.

Study of case (1)

We can write, using Taylor-Izritch expansion formula :

$$\frac{P_i(N(z))}{Q_i(N(z))} = \frac{P_i(\tau)}{Q_i(\tau)} + \frac{\partial}{\partial N}\left[\frac{P_i}{Q_i}\right]_{|N=\tau}(N(z)-\tau) + \frac{\partial^2}{\partial N^2}\left[\frac{P_i}{Q_i}\right]_{|N=\tau}(N(z)-\tau)^2 + O(|N(z)-\tau|^2).$$

Applying the transfer lemmas of section 2, and using the approximation :

$$N(z) = \tau - \sqrt{\frac{2\Phi(\tau)}{\Phi''(\tau)}}\left[1-\frac{z}{\rho}\right]^{1/2} + \gamma 1\left[1-\frac{z}{\rho}\right] + O\left(\left[1-\frac{z}{\rho}\right]^{3/2}\right) \qquad \text{(F1)}$$

we get :

$$C_n^f = k_1\sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\rho^{-n}n^{-3/2}\ \left(1 + O\left(\frac{1}{n}\right)\right), \quad \text{with } k_1 = \frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\frac{\partial}{\partial N}\left[\frac{P_i}{Q_i}\right]_{|N=\tau},$$

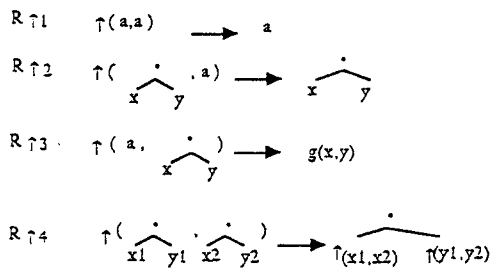
and finally :

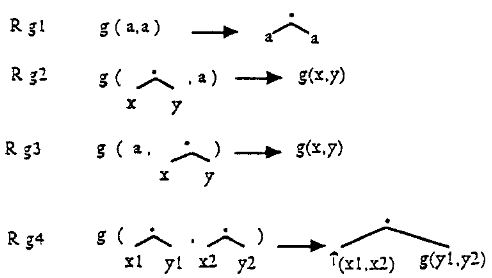
Result 3.4

$$\overline{C}_n^f = k_1\ \left(1 + O\left(\frac{1}{n}\right)\right), \quad \text{with } k_1 = \frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\frac{\partial}{\partial N}\left[\frac{P_i}{Q_i}\right]_{|N=\tau} \qquad \square$$

Example 3.1

We consider, on binary trees built as previously, another version of a shuffle function on trees, defined by the following set of rules :





The matrix M(z) associated to the ↑ and g is :

$\begin{bmatrix} 2z^2N^2(z) & z^2 \\ z^2N^2(z) & 2z^2+z^2N^2(z) \end{bmatrix}$, and $Y(z) = \begin{bmatrix} N^2(z) \\ N^2(z) \end{bmatrix}$. This yields, after computation of the determinants, and replacement of z by $\frac{N(z)}{1+N^2(z)}$ (these computations have been performed with assistance of the

MAPLE program [MAPLE 85]) :

$$C^{\uparrow}(z) = \frac{(1+N^2(z)+N^4(z))\,N^2(z)\,(1+N^2(z))^2}{1+2N^2(z)-N^4(z)-N^6(z)} = \frac{P_1(N(z))}{Q_1(N(z))}$$

$$C^{g}(z) = \frac{(1+2N^2(z)+N^6(z))\,N^2(z)\,(1+N^2(z))^2}{1+2N^2(z)-N^4(z)-N^6(z)} = \frac{P_2(N(z))}{Q_2(N(z))}$$

The denominator $(1+2N^2-N^4-N^6)$ has no root for $N\in[0,1]$. We are therefore in the current case (1). Computation gives : $\frac{\partial}{\partial N}\left[\frac{P_1}{Q_1}\right]_{|N=1} = 144$ and $\frac{\partial}{\partial N}\left[\frac{P_2}{Q_2}\right]_{|N=1} = 200$. Finally :

$$\overline{C}_n^{\uparrow} = 72\,(1 + O\,(\frac{1}{n}))$$

$$\overline{C}_n^{g} = 100\,(1 + O\,(\frac{1}{n})).$$

End of example 3.1

Study of case (2)

We can write :

$$\frac{P_i(N(z))}{Q_i(N(z))} = \frac{1}{(N(z)-\tau)^s}\,\frac{P_i(N(z))}{\overline{Q}_i(N(z))},$$

where s is a strictly positive integer, and $\overline{Q}_i$ is an integer polynomial such that $\overline{Q}_i(\tau) \neq 0$. Then :

$$\frac{P_i(N(z))}{Q_i(N(z))} = \frac{P_i(\tau)}{\overline{Q}_i(\tau)}(N(z)-\tau)^{-s} + \gamma_2(N(z)-\tau)^{-s+1/2} + O(|N(z)-\tau|^{-s+1}).$$

Applying the transfer lemmas of section 2, and using the approximation (F1), we obtain :

$$C_n^{f} = k'_2\,\rho^{-n} n^{\frac{s}{2}-1}\;(1+O(\frac{1}{n^{1/2}})),\text{ with } k'_2 = (-1)^s\frac{P_i(\tau)}{\overline{Q}_i(\tau)}\left[\frac{2\Phi(\tau)}{\Phi''(\tau)}\right]^{-\frac{s}{2}}\frac{1}{\Gamma(s/2)},$$

and thus :

$$\overline{C}_n^{f} = k_2 n^{\frac{s+1}{2}}\;(1+O(\frac{1}{n^{1/2}})),\text{ with } k_2 = (-1)^{s-1}\frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\,\frac{P_i(\tau)}{\overline{Q}_i(\tau)}\left[\frac{2\Phi(\tau)}{\Phi''(\tau)}\right]^{-\frac{(s+1)}{2}}\frac{\Gamma(-1/2)}{\Gamma(s/2)},$$

Finally, using $\Gamma(1/2) = \sqrt{\pi}$ and $\Gamma(s/2) = \begin{cases} (p-1)! & \text{if } s=2p \\ \dfrac{(2p-1)!}{(p-1)!}\,\dfrac{1}{2^{2p+1}} & \text{if } s=2p+1 \end{cases}$, we obtain :

Result 3.5

$$\overline{C}_n^{f_i} = k_2 n^{\frac{s+1}{2}}\;(1+O(\frac{1}{n^{1/2}})),$$

with

$$k_2 = -\frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\,\frac{P_i(\tau)}{\overline{Q}_i(\tau)}\quad\left[\frac{2\Phi''(\tau)}{\Phi(\tau)}\right]^{-(p+1)}\frac{\sqrt{\pi}\,(p-1)!}{(2p-1)!\,4^{p-1}}\qquad \text{if } s = 2p$$

$$k_2 = \frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\,\frac{P_i(\tau)}{\overline{Q}_i(\tau)}\quad\left[\frac{2\Phi''(\tau)}{\Phi(\tau)}\right]^{-\frac{2p+1}{2}}\frac{2\sqrt{\pi}}{(p-1)!}\qquad \text{if } s = 2p+1 \quad \square$$

Example 3.2

We consider the same system as in example 3.1, except that rule $R_{\uparrow 2}$ is replaced by rule $R_{\uparrow 2'}$:

$$R\uparrow 2' \qquad \uparrow(\ \overset{\circ}{\underset{x\quad y}{\wedge}}\ , a) \longrightarrow \quad g(x,y)$$

The matrix M(z) associated to the $\uparrow$ and g is now :

$$\begin{bmatrix} 2z^2N^2(z) & 2z^2 \\ z^2N^2(z) & 2z^2+z^2N^2(z) \end{bmatrix}.$$

We obtain, after replacement of z by $\dfrac{N(z)}{1+N^2(z)}$:

$$\det(I\text{-}M(z)) = (1-N(z))\,\frac{(N^2(z)+2)\,(N(z)+1)}{(1+N^2(z))^3}$$

which yields :

$$C^{\uparrow}(z) = \frac{1}{1-N(z)}\,\frac{N^2(z)\,(1+N^2(z))}{1+N(z)} = \frac{1}{1-N(z)}\,\frac{P_1(N(z))}{\overline{Q}_1(N(z))}$$

$$C^{g}(z) = \frac{1}{1-N(z)}\,\frac{N^2(z)\,(1+N^2(z))}{1+N(z)} = \frac{1}{1-N(z)}\,\frac{P_2(N(z))}{\overline{Q}_2(N(z))}$$

We are in the current case (1). Computation gives : $\dfrac{P_1(1)}{\overline{Q}_1(1)} = \dfrac{P_2(1)}{\overline{Q}_2(1)} = 1$, and finally :

$$\overline{C}_n^{\uparrow} = \sqrt{\frac{\pi}{2}}\,n^{1/2}\ (1+O(\frac{1}{\sqrt{n}}))$$

$$\overline{C}_n^{g} = \sqrt{\frac{\pi}{2}}\,n^{1/2}\ (1+O(\frac{1}{\sqrt{n}}))$$

End of example 3.2

Study of case (3)

Let $\tau_0^i = N(\rho_0^i)$. We have : $0 < \rho_0^i < \rho$, and $0 < \tau_0^i < \tau$. We can write :

$$\frac{P_i(N(z))}{Q_i(N(z))} = \frac{1}{(N(z)-\tau_0^i)^s}\,\frac{P_i(N(z))}{\overline{Q}_i(N(z))},$$

where s is a strictly positive integer, and $\overline{Q}_i$ is an integer polynomial such that $\overline{Q}_i(\tau_0^i) \neq 0$. Then, using a Taylor-Izritch expansion of $z = \dfrac{N(z)}{\Phi(N(z))}$ in the neighbourhood of ρ_0^i, we get :

$$z = \frac{N(z)}{\Phi(N(z))} = \frac{\tau_0^i}{\Phi(\tau_0^i)} + \frac{\partial}{\partial N}\left[\frac{N}{\Phi(N)}\right]_{|N=N_0}(N-\tau_0^i) + O(|z-\rho_0^i|^2),$$

from which we derive :

$$N - \tau_0^i = \frac{1}{\dfrac{\partial}{\partial N}\left[\dfrac{N}{\Phi(N)}\right]_{|N=N_0}}\,(z-\rho_0^i) + O(|z-\rho_0^i|^2)$$

$$= \frac{-1}{\frac{1}{\tau_0^i} - \frac{\Phi'(\tau_0^i)}{\Phi(\tau_0^i)}} \left(1 - \frac{z}{\rho_0^i}\right) + O\left(\left|1 - \frac{z}{\rho_0^i}\right|^2\right).$$

Using this developement in the previous expression of $\frac{P_i(N(z))}{Q_i(N(z))}$ yields :

$$\frac{P_i(N(z))}{Q_i(N(z))} = (-1)^s\left(1-\frac{z}{\rho_0^i}\right)^{-s}\left[\frac{1}{\tau_0^i} - \frac{\Phi'(\tau_0^i)}{\Phi(\tau_0^i)}\right]^s \frac{P_i(\tau_0^i)}{\overline{Q}_i(\tau_0^i)} + O\left[\left|1-\frac{z}{\rho_0^i}\right|^{-s+\frac{1}{2}}\right].$$

Thus,

$$C_n^f = (-1)^s\left[\frac{1}{\tau_0^i} - \frac{\Phi'(\tau_0^i)}{\Phi(\tau_0^i)}\right]^s \frac{P_i(\tau_0^i)}{\overline{Q}_i(\tau_0^i)}(\rho_0^i)^{-n}\frac{n^{s-1}}{\Gamma(s-1)} \quad \left(1+O\left(\frac{1}{\sqrt{n}}\right)\right),$$

and finally :

Result 3.6

$$\overline{C}_n^{f_i} = k_3\left[\frac{\rho}{\rho_0^i}\right]^n n^{s+\frac{1}{2}}\left(1+O\left(\frac{1}{\sqrt{n}}\right)\right),$$

with $k_3 = \frac{(-1)^s}{(s-1)!}\frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\left[\frac{1}{\tau_0^i} - \frac{\Phi'(\tau_0^i)}{\Phi(\tau_0^i)}\right]^s \frac{P_i(\tau_0^i)}{\overline{Q}_i(\tau_0^i)}\sqrt{\frac{2\pi\Phi''(\tau_0^i)}{\Phi(\tau_0^i)}}$ □

Example 3.3

We consider the same system as in example 3.1 (or example 3.2), except that rule $R_{\uparrow 2}$ (or rule $R_{\uparrow 2'}$) is replaced by rule $R_{\uparrow 2''}$:

R ↑ 2''

The matrix M(z) associated to the ↑ and g is now :

$$\begin{bmatrix} 2z^2N^2(z) & 3z^2 \\ z^2N^2(z) & 2z^2+z^2N^2(z) \end{bmatrix}.$$

This yields, after computation of the determinants, and replacement of z by $\frac{N(z)}{1+N^2(z)}$:

$$C^{\uparrow}(z) = \frac{(1+N^2(z))^2\,(1+3N^2(z))\,N^2(z)}{1+2N^2(z)-N^4(z)-3N^6(z)}$$

$$C^{g}(z) = \frac{(1+N^2(z))^2\,(1+2N^2(z))\,N^2(z)}{1+2N^2(z)-N^4(z)-3N^6(z)}$$

The expression $(1+2N^2(z)-N^4(z)-3N^6(z))$ admits a root for $N_0 \sim 0.93336$, which gives $\tau_0 \sim 0.49881$ ($< \tau = 1/2$). We are therefore in the current case (3), and computation gives finally :

$$\overline{C}_n^{\uparrow} = k_3\left[\frac{\rho}{\rho_0^i}\right]^n \sqrt{n}\ \left(1+O\left(\frac{1}{\sqrt{n}}\right)\right)$$

$$\overline{C}_n^g = k'_3 \left[\frac{\rho}{\rho_0^i}\right]^n \sqrt{n}\ (1+O(\frac{1}{\sqrt{n}}))$$

with $k_3 \sim 0.27901$, $k'_3 \sim 0.21234$, and $\frac{\rho}{\rho_0^i} \sim 1.00238$

End of example 3.3

We notice that, simply modifying *one* rule between examples 3.1, 3.2 and 3.3, induces respectively constant, polynomial or exponential cost. This illustrates the great sensitivity of the cost of rewriting w.r.t. mild modifications within the rewrite rules.

Thus, in each particular case, the asymptotic developments are obtained with very few computations, that just rely on the "geometry" of the system. The user of our methods actually never needs to manipulate formal series ; (s)he simply has to apply theorems 3.3, 3.4 or 3.6 (according to the singularity closest to the origin of the $Q_i(N(z))$'s.

4. Conclusion

For the class of the *regular* term rewriting systems, we have provided ways of obtaining asymptotic evaluations of the cost series. The user does not need to actually manipulate formal series, since our results are given under the form of ready-to-use formulae. These results solely depend on physical characteristics of the system, easily obtainable : number of variables and of constructors in the left-hand sides, occurrences of derived operators in the right-hand sides. Then, the average cost is constant, polynomial or exponential, according to the position of the singularity of the expressions $Q_i(N(z))$ closest to the origin.

Acknowledgements

We thank Jean-Pierre Jouannaud for contribution to the proof of Theorem 3.1 and Philippe Flajolet for fruitfull discussions. This work has been partially supported by the C.N.R.S. P.R.C. de Programmation, and the ESPRIT Meteor project.

5. References

[Bel 85] Belhassen S., "Séries formelles de complexité dans les types abstraits algébriques", Rapport de DEA, Orsay 1985.

[BBWT 81] Bergstra J.A., Broy M., Wirsing M., Tucker J.V., "On the power of algebraic specifications", Proc. of the M.F.C.S. Conference, L.N.C.S. 118, Springer Verlag, 1981.

[BCV 85] Bidoit M., Choppy C., Voisin F., "The Asspegique specification environment : motivations and design", Proc. 3rd Workshop on Theory and Applications of Abstract data types", Bremen, Nov.

1984. Recent Trends on Data Type Specification (H.-J. Kreowski ed.), Informatik Fachberichte 116, Springer-Verlag, Berlin-Heidelberg, 1985.

[C&K 83] Choppy C., Kaplan S., "Complexity calculus for abstract data types", LRI Report no 147, Nov. 1983.

[CLR 80] Choppy C., Lescanne P., Rémy J.-L., "Improving abstract data type specification by appropriate choice of constructors", Proc. Intern. Workshop on Program Construction, Bonas, France, 1980, Mac Millan, A. Biermann, G. Guiho and Y. Kodratoff (eds), 1983.

[Der 82] N. Dershowitz, "Orderings for term-rewriting systems", T.C.S. vol. 17.3, March 1982.

[Der 85] N. Dershowitz, "Termination", Proc. of the RTA'85 Conference, L.N.C.S. 202, 1985.

[E&M 81] Ehrig H., Mahr B., "Complexity of algebraic implementations for abstract data types", J. of Computer and System Sciences, vol 23, no 2, Oct, 1981, pp. 223-253.

[Fla 87] Flajolet P., "The symbolic operator method", *in* Mathematical methods in the analysis of algorithms and data structures, L.N.C.S., Springer-Verlag, to appear 1987.

[F&S 82] Flajolet P., Steyaert J.M., "A complexity calculus for classes of recursive search programs over tree structures", Proc. 22nd IEEE Symp. on Foundations of Computer Science, Nashville, 1982, pp. 386-393.

[F&G 84] Forgaard R., Guttag J.V., "REVE : a term rewriting system generator with failure-resistant Knuth-Bendix", Proc. of an NSF Workshop on the rewrite rule laboratory, Report no 84GEN008, General Electric, Apr. 1984.

[FGJM 84] Futatsugi K., Goguen J.A., Jouannaud J.P., Meseguer J., "Principles of OBJ2", CRIN Report 84-R-066.

[GSF 86] Gaudel M.C., Soria M., Froidevaux C., "Types de données et algorithmes - vol 1 : Analyse d'algorithmes, Définition des types de données", Collection Didactique, INRIA, 1986.

[GHW 85] Guttag J.V., Horning J.J., Wing J.M., "Larch in five easy pieces" Digital System Research Center Report, Jul. 1985.

[Kap 86] Kaplan S., "A compiler for conditional term rewriting systems" Proc. of the RTA'87 Conference, L.N.C.S. this volume, also : L.R.I. Report no 315, Dec. 1986.

[Knu 68] Knuth D., "The art of computer programming : Fundamental algorithms", Addison Wesley, Reading, 1968.

[Knu 73] Knuth D., "The art of computer programming : Sorting and searching", Addison Wesley, Reading, 1973.

[M&M 78] Meir A., Moon J.W., "On the altitude of nodes in random trees", Canadian Journal of Math. 30, 1978, pp. 997-1015.

[MAPLE 85] Char B.W., Geddes K.O., Gonnet G.H., Watt S.M., "MAPLE : Reference Manual", University of Waterloo, 1985.

[S&S 84] Soria M., Steyaert J.M., "Average efficiency of pattern matching on Lisp expressions", Proc. of the CAAP 84 Conference, also : LRI Report no 178, May 1984.

[Stey 84] Steyeart J.M., "Complexité et structure des algorithmes", Thèse d'Etat, Université de Paris 7, 1984.