UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Rapports de Recherche

N°1715

Programme 2

Calcul symbolique, Programmation et Génie logiciel

GENERAL ASYMPTOTIC SCALES AND COMPUTER ALGEBRA

Bruno SALVY

Juin 1992

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1)39 63 55 11

General Asymptotic Scales and Computer Algebra

Bruno Salvy
Algorithms Project,
INRIA Rocquencourt,
78153 Le Chesnay Cedex,
France

Abstract

In many natural applications, one encounters asymptotic expansions of a form more complicated than mere Puiseux series. Existing computer algebra systems lack good algorithms for handling such asymptotic expansions. We present tools that permit the representation and automatic handling of general exp-log asymptotic expansions.

Échelles Asymptotiques Générales et Calcul Formel

Résumé

Dans de nombreuses applications naturelles, les développements asymptotiques qui apparaissent sont d'une forme plus complexe que les séries de Puiseux. Les systèmes de calcul formel existant manquent de bons algorithmes pour manipuler de telles séries. Nous présentons des outils qui permettent la représentation et la manipulation des développements asymptotiques de type logarithmico-exponentiel.

To appear in *Proceedings NATO Advanced Research Workshop on Asymptotic-induced Numerical Methods for Partial Differential Equations, Critical Parameters, and Domain Decomposition*, H. Kaper ed., Kluwer Academic Publishers, 1992.

GENERAL ASYMPTOTIC SCALES AND COMPUTER ALGEBRA

Bruno Salvy
Algorithms Project
INRIA Rocquencourt
78153 Le Chesnay Cedex
France

ABSTRACT. In many natural applications, one encounters asymptotic expansions of a form more complicated than mere Puiseux series. Existing computer algebra systems lack good algorithms for handling such asymptotic expansions. We present tools that permit the representation and automatic handling of general exp-log asymptotic expansions.

1. Introduction

The process of computing an asymptotic expansion generally consists of two distinct parts: one is the application of a method (Laplace method, the saddle-point method, integration by parts, indeterminate coefficients,...), which usually requires some expertise, and the other part is a simple but tedious manipulation of asymptotic expansions, adding or multiplying them, computing their logarithm,.... The purpose of this paper is to describe the automation of this latter part, which must be prior to any attempt at automating the particular application considered. We shall focus on finding a good representation for asymptotic expansions, and on describing algorithms to perform elementary operations (sum, product, powering, exponential and logarithm) on objects stored with this representation. The emphasis is put on asymptotic expansions as *objects* as opposed to asymptotic expansion as a *process*.

Many more-or-less general definitions of asymptotic expansions have been given. We shall use the following.

DEFINITION 1 (Asymptotic Scales). *Let $\mathcal{E}$ be a set of points with a limit point $\mathcal{P}$, and let $\mathcal{S} = \{\phi_i\}_{i\in I}$ be a set of functions defined in a neighborhood* *of $\mathcal{P}$, taking their values in $\mathbb{C} \setminus \{0\}$. I is a totally ordered index set. Then $\mathcal{S}$ is an* asymptotic scale *at $\mathcal{P}$ if and only if*

$$\forall (i,j) \in I \times I, \quad i < j \Longrightarrow \frac{|\phi_j(x)|}{|\phi_i(x)|} \to 0, \quad x \to \mathcal{P} \text{ in } \mathcal{E}.$$

DEFINITION 2 (Asymptotic Expansions). *Let $\mathcal{S} = \{\phi_i\}_{i \in I}$ be an asymptotic scale. Then an* asymptotic expansion *in $\mathcal{S}$ is a formal sum of the form*

$$a_1(x)\phi_{i_1}(x) + a_2(x)\phi_{i_2}(x) + \cdots + a_k(x)\phi_{i_k}(x) + r(x),$$

where $i_1 < i_2 < \cdots < i_k$, and the $a_i(x)$ are functions from $\mathcal{E}$ to $\mathbb{C}$, whose modulus has a nonzero finite upper limit as $x \to \mathcal{P}$, and

$$|r(x)|/|\phi_{i_k}(x)| \to 0, \quad \text{as } x \to \mathcal{P} \text{ in } \mathcal{E}.$$

When the functions a_i are constants, these expansions are *of Poincaré type*, but more general expansions are necessary, for instance, to express the asymptotic expansion of Bessel J functions. In this paper, "asymptotically" means "as $x \to \mathcal{P}$," and in most of our examples, the set $\mathcal{E}$ is $\mathbb{C}$ and $\mathcal{P}$ is either 0 or ∞.

Examples of asymptotic scales at 0 include $\{x^k, k \in \mathbb{Z}\}$, which gives (truncated) formal power series as corresponding asymptotic expansions. A more general scale is $\{x^r, r \in \mathbb{Q}\}$, corresponding to Puiseux expansions. Both of these scales are important in computer algebra, since their manipulation reduces to operations on polynomials in one variable. Our aim in this paper is to algorithmically reduce manipulations of more general asymptotic expansions to manipulations of polynomials, tasks that are routinely handled by computer algebra systems. Currently, most of these systems' asymptotic capabilities are reduced to formal power series manipulation. Maple is an exception, using a model of "hierarchical series" introduced in [5].

In Section 2, we separate computations on asymptotic expansions into operations on the coefficients and operations on the asymptotic scale. Section 3 deals with the former ones. The rest of the paper is concerned with operations on the asymptotic scale and on computing with the most general asymptotic scale. In Section 4, we study the case of a scale that is fixed *a priori*, thus generalizing computations with formal power series. This corresponds to restricting the logarithm and the exponential to expansions of particular valuations. Then, relaxing this limitation in Section 5, we give an algorithm to perform expansions in a scale that is determined dynamically during the computation.

2. Coefficients and Scale

The choice of a finite representation for mathematical objects depends heavily on the operations one wishes to perform on these objects. In the case of asymptotic expansions, the desired operations contain addition, product, inverse, exponential, and logarithm. Henceforth, we restrict ourselves to these operations, which we shall call *elementary*. We now spend some time on the machinery required to perform these elementary operations on asymptotic expansions, decomposing them into operations on the set of coefficients and operations on functions of the asymptotic scale.

Addition

Let A and B be the following asymptotic expansions:

$$\begin{aligned} A &= a_1\phi_{i_1} + a_2\phi_{i_2} + \cdots + a_k\phi_{i_k} + o(\phi_{i_k}), \\ B &= b_1\phi_{j_1} + b_2\phi_{j_2} + \cdots + b_k\phi_{j_p} + o(\phi_{j_p}). \end{aligned} \tag{1}$$

Then one computes $A + B$ as follows:

- Compare ϕ_{i_1} and ϕ_{j_1}.
- If one asymptotically dominates the other, then the corresponding term is output, and the same process is applied to the remainder of the expansions.
- Otherwise, $a_1 + b_1$ is compared to 0, and depending on the result, either one or zero term of the result have been found, and the same process is applied to the remainder of the expansions.
- When one of the $o()$ is reached, its argument is compared against the current ϕ_m in the other expansion, and a similar process applies.

The necessary operations are thus: ordering in the scale, addition of coefficients and comparison of a coefficient to 0. When the coefficients are not constants, by comparison to 0, we mean determination of the upper limit of the modulus, and then comparison of it to 0. This test is necessary to insure that the resulting expansion still obeys the definition of asymptotic expansions.

Product

With the same notation,

- If $A = a_1\phi_{i_1}$, then provided the functions $\phi_{i_1} \cdot \phi_{j_k}$ are in the scale, the result is

  $$a_1 b_1 \phi_{i_1}\phi_{j_1} + \cdots + o(\phi_{i_1}\phi_{j_k}).$$
- Otherwise, expand the product to reduce the computation to the previous case and to addition.

This operation necessitates that the scale is closed by product, and the product of coefficients is needed too.

Inverse

With the same notation as above, in order to compute the expansion $1/A$, the functions $1/\phi_{i_1}$ and the constant function 1 have to belong to the scale, and it is necessary that the lower limit of $|a_1|$ is nonzero. Then the inverse is computed as

$$\frac{1}{a_1}\frac{1}{\phi_{i_1}}\cdot[1-\psi+\psi^2-\psi^3+\cdots],$$

where

$$\psi=\frac{a_2}{a_1}\frac{\phi_{i_2}}{\phi_{i_1}}+\frac{a_3}{a_1}\frac{\phi_{i_3}}{\phi_{i_1}}+\cdots+o(\phi_{i_k}/\phi_{i_1}),$$

whose integral powers can be computed by successive products.

Besides those that have already been mentioned, the necessary operations in this case are taking an inverse in the scale, inversibility, and inverse of the coefficients.

Logarithm

We shall say that A *has valuation 0* when ϕ_{i_1} is the constant function 1. Then by analogy with formal power series, we define the logarithm of an asymptotic expansion only if this asymptotic expansion has valuation 0, and in this case the result is

$$\log a_1-\psi+\psi^2/2-\psi^3/3+\cdots,$$

with ψ as above. A generalization of this definition of the logarithm will be given in Section 5. With this definition, the new necessary operations concern only the coefficients. These are logarithm and product by rational numbers.

Exponential

Several choices have to be made in order to define an exponential of asymptotic expansions. We take a minimal approach here, and we shall give a more general version in Section 5.

If ϕ_{i_1} tends to 0 (which is computed by comparing it to the constant function 1), we shall say that the expansion has *positive valuation*, and the result is

$$1+A+A^2/2!+A^3/3!+\cdots,$$

which is computed with operations that have already been defined. Once again, by analogy with formal power series, we shall define the exponential in this case only.

With the restrictions we make on logarithm and exponential, we get the important property that *during the computation, the scale and the set of coefficients are handled independently from one another.* From a computational point of view, this means that one can adopt a very modular way of programming, implementing the part which concerns the coefficients on one side, the part that concerns the scale on the other one, and then the algorithm we have just outlined produces asymptotic expansions. We have thus implemented in `Gauss`[1] the prototype of a program whose input consists of a domain of coefficients and an asymptotic scale, and whose output is the corresponding domain of asymptotic expansions (see [14] for more details).

Putting together all the operations we have found necessary during this description, we can be rather precise concerning the scale and the coefficients:

> *The asymptotic scale has to be an effective ordered Abelian group and the set of coefficients has to be an effective $\mathbb{Q}$-algebra.*

By "effective ordered Abelian group," we mean a group whose law and order can be computed, as well as the inverse with respect to the law. Examples of this are $\mathbb{Z}$, corresponding to Laurent expansions; $\mathbb{Q}$, corresponding to Puiseux expansions; or $\mathbb{Z}^2$ ordered lexicographically, corresponding to expansions in the scale $\{x^p \log^q x\}$ (for instance).

Similarly, by "effective $\mathbb{Q}$-algebra," we mean an algebra whose operations can be computed, most notably inversibility, inverse, and product by rational numbers. Examples include $\mathbb{Q}$ itself, any algebraic extension of $\mathbb{Q}$, or for instance $\mathbb{Q}[\sin(x), \cos(x)]$.

3. Coefficient Domains

Since we are working with symbolic objects, most operations can be performed symbolically. For instance, the inverse of an object A is simply $1/A$. Working this way, the real difficulty has been moved to decision procedures: deciding whether a function of the scale is asymptotically larger than another one, or deciding whether a constant is 0 or not. The latter problem is classically called *the equivalence problem*. It is known to be decidable in any algebraic extension of $\mathbb{Q}$ (see, for instance, [8]), and it is known to be undecidable for sufficiently rich classes of expressions involving a variable [10].

For classes of constants more general than algebraic extensions of $\mathbb{Q}$, the problem is difficult and related to conjectures in number theory such as Schanuell's conjecture. Some heuristics are known (see [11] for a review of these problems), and we shall assume here the existence of an "oracle" that

[1] Gauss is a program developed by Michael Monagan [9] in Maple, which implements domains and categories *à la* Axiom.

is able to decide whether a constant is 0 or not in the class of constants obtained by closing $\mathbb{Q}$ under exponential and logarithm. In practice, such an oracle can often be obtained by a numerical evaluation routine and a given ϵ threshold.

Another coefficient domain of use in practice is $\mathbb{Q}[\sin x, \cos x]$, where equivalence is decidable. One can even replace $\mathbb{Q}$ by any algebraic extension of $\mathbb{Q}$.

4. Effective Scales

In this section we consider effective ordered groups given by a finite set of generators and the required operations. We show that this forms the basis for handling very general asymptotic expansions purely automatically.

To fix the ideas, here are several examples of effective ordered groups:

- $\{x^\alpha; \alpha \in \mathbb{Q}\}$, which corresponds to Puiseux series;
- $\{x^r \log^k x; r \in \mathbb{Q}[\alpha], k \in \mathbb{N}\}$, which is the scale necessary to handle the expansions of singular solutions of linear differential equations in the neighborhood of a regular singular point;
- without any extra difficulty, more complicated scales also belong to this framework, here follows an example involving Euler's Γ function:
  $\{\Gamma^p(\Gamma(x)) \exp(qe^x)\Gamma^r(x)x^s \log^t x; (p, q, r, s, t) \in \mathbb{Q}^5\}$.

In each of these examples, the problem of comparing asymptotically two monomials in the scale (two elements of the group) is reduced to a lexicographic comparison of exponents. This will be the case in general.

4.1. SETS OF EXPONENTS

If we wish to handle as general scales as possible, it is necessary that the set of exponents be made as large as possible. Ideally, one would wish to work with $\mathbb{R}$, and we have to content ourselves with an effective ordered subgroup of $\mathbb{R}$.

Algorithms exist that permit computations in real algebraic extensions of $\mathbb{Q}$ [2]. Like in the constant case, more general, ordered, real constant structures are not yet completely effective and are related to long-standing conjectures in number theory. We shall therefore once again assume the existence of an oracle deciding the comparison of exp-log real constants, and we shall denote by $\mathcal{E}$ any ordered group given with an implementation.

4.2. FINITELY GENERATED SCALES

The following remark makes possible very general automatic computations. Its simplest special case is formal power series.

Given a finite set of functions $\{\phi_1, \ldots, \phi_k\}$ tending to infinity as $x \to \mathcal{P}$, and such that the following relations are known:

$$\forall i \in \{1,\ldots,k-1\}, \quad \log|\phi_{i+1}(x)| = o(\log|\phi_i(x)|), \quad \text{as } x \to \mathcal{P},$$

then the scale

$$\{\phi_1^{\alpha_1}\phi_2^{\alpha_2}\cdots\phi_k^{\alpha_k}, \alpha_i \in \mathcal{E}\}$$

is an effective scale. It is clearly a group, and the order is computed by lexicographical comparison of the exponents.

An important special case of this is as follows: for any polynomial $P \in \mathbb{Q}[\alpha][X^{p/q}]$, with α an algebraic number and p and q integers, the scale given by the generators $\{\exp(P(x)), x, \log x\}$, which is sufficient to express asymptotic expansions of singular solutions of linear differential equations about an irregular singular point.

4.3. A GENERAL EXP-LOG SCALE

To simplify notations, suppose $\mathcal{P} = \infty$. As usual in this context, let us denote by $e_n(x)$ the nth iterate of the exponential function, and likewise $l_n(x)$ for the logarithm. If $n < 0$ we define e_n to be l_{-n} and vice versa. These functions occur very frequently in asymptotic expansions. It is clear that any finite set of them can be handled as above. But more useful is to encode e_n as X^n and l_n as X^{-n}, and then use a group morphism from the scale into the additive group of Laurent polynomials in one variable. In other words, the monomial

$$e_{n_1}^{\alpha_1}(x)e_{n_2}^{\alpha_2}(x)\cdots e_{n_p}^{\alpha_p}(x)l_{m_1}^{\beta_1}(x)\cdots l_{m_k}^{\beta_k}(x)$$

is encoded as

$$\alpha_1 X^{n_1} + \cdots + \beta_k X^{-m_k}.$$

The order between two such polynomials is obtained by computing the sign of the leading coefficient of their difference.

This scale is very well suited to an implementation in computer algebra systems, whose polynomials are generally carefully implemented since this is crucial to their overall efficiency.

5. Dynamic Scales

With the restrictions we have made on the definition of the logarithm and the exponential in Section 2, there is no way the scale can be extended during the computation. We now consider defining the exponential function of asymptotic expansions in some cases when the argument does not have a positive valuation. Then independence between the algebra of coefficients and the scale no longer holds, and the scale gets richer as the computation proceeds. Adding a few restrictions, we show how such an extended algorithm can be handled.

5.1. EXTENSION STEP

To keep under control the relation between the scale and the algebra of coefficients, we shall restrict ourselves to cases where the algebra of coefficients is strictly ordered, so that it can also play the rôle of a set of exponents. A good example of an effective ordered $\mathbb{Q}$-algebra of coefficients is given by any real algebraic closure of $\mathbb{Q}$.

We now generalize the definition of exponential of Section 2. Consider again the asymptotic expansion A defined in equation (1). If there exists $j \in \{1, \ldots k\}$ such that ϕ_{i_j} has positive valuation, define ψ as

$$\psi = a_j \phi_{i_j} + \cdots + a_k \phi_{i_k} + o(\phi_{i_k}),$$

and extend the scale with the function

$$\Phi = \exp[a_1 \phi_{i_1} + \cdots + a_{j-1} \phi_{i_{j-1}}]. \tag{2}$$

The result is given by

$$\Phi \cdot (1 + \psi + \psi^2/2! + \psi^3/3! + \cdots),$$

this last product being computed by the same method as before. The remainder of this section addresses the extension of the scale.

In some cases, it may happen that the function Φ defined above already belongs to the scale. However, a famous theorem [4] states that given any asymptotic scale, it is possible to construct a function that is not asymptotically equivalent to any function of the scale. This means that there is no "universal" asymptotic scale and that we probably cannot avoid the problem of extending the scale.

It can be seen that the only extra operation needed to handle this extension is the exponential in the ordered group of the scale. Once again, the problem is in the decision procedure. We have to find an ordered group closed under exponential, where the order is computable.

5.2. USING SHACKELL'S ALGORITHM

The class of exp-log functions (functions whose expression is obtained by repeated application of logarithm, exponential and field operations starting from one variable x and rational numbers, with the restriction that any subexpression must be real for sufficiently large x) was studied from an asymptotic point of view in [7]. However, it was only in [3] that the comparison problem for exp-log functions was shown to be reducible to constant comparisons, and in [15] that an effective algorithm was produced to perform this reduction.

Although no complexity study of this comparison algorithm has been done, it is rather expensive in practice. We now describe a way to minimize the number of calls to this algorithm. The idea is to compute an extension of the exp-log scale we have described in Section 4, inserting rational powers of the variable X in the Laurent polynomials, so that eventually we work with Puiseux polynomials.

Suppose then that we start from a scale encoded by $\mathcal{S} = \mathbb{Z} \cup \{r_1, \ldots, r_k\}$ corresponding to a scale based on $\mathcal{B} = \{e_n(x), l_n(x); n \in \mathbb{N}\} \cup \{\psi_1, \ldots, \psi_k\}$, where each ψ_i is represented by X^{r_i} with $r_i \in \mathbb{Q} \setminus \mathbb{Z}$, such that if X^u and X^v are the representations of two functions ϕ_u and ϕ_v, then

$$u < v \Longrightarrow \log|\phi_u(x)| = o(\log|\phi_v(x)|), \quad \text{as } x \to \mathcal{P}. \tag{3}$$

Although the representation of the scale as an additive group of Puiseux polynomials is efficient computationally, to simplify the notation, we shall define e_r to be the member of $\mathcal{B}$ whose representative in $\mathcal{S}$ is r, and we shall denote the group of Puiseux polynomials multiplicatively. Thus a member of the scale is a function

$$e_{r_1}^{\alpha_1}(x) \cdot e_{r_2}^{\alpha_2}(x) \cdots e_{r_k}^{\alpha_k}(x).$$

When a new function Φ is added to the scale through computing the exponential of an asymptotic expansion of non positive valuation as in Equation (2), the algorithm first expands Equation (2) and associates a monomial to each $\exp(a_p\phi_{i_p})$, creating it if it does not belong to $\mathcal{B}$ yet. From Equation (3), this will be done by comparing logarithms of functions that are already in the scale to the logarithm of the function whose adjunction we are considering, this logarithm being simply $a_p\phi_{i_p}$ which is already a member of the scale. The following operations are performed:

1. From the representation of ϕ_{i_p} as a Puiseux polynomial
   $$e_{r_1}^{\alpha_1}(x) \cdot e_{r_2}^{\alpha_2}(x) \cdots e_{r_k}^{\alpha_k}(x),$$
   with $r_1 > r_2 > \cdots > r_k$ and $\alpha_1 > 0$, it is not difficult to see that a representation of $\exp(\phi_{i_p})$ as X^R must satisfy
   $$\lfloor r_1 \rfloor \leqslant R \leqslant \lceil r_1 \rceil + 1,$$
   these inequalities being strict when r_1 is not an integer.
2. Using Shackell's algorithm and a dichotomy among the *finite* set of rational numbers belonging to $(\lfloor r_1 \rfloor, \lceil r_1 \rceil + 1) \cap \mathcal{S}$, we reach a stage where either we have found two rational numbers r_m and r_{m+1} in $\mathcal{S}$, such that $r_m < R < r_{m+1}$, or we have found an index $r_m \in \mathcal{S}$ such that
   $$\lim_{x \to \mathcal{P}} \frac{\phi_{i_p}(x)}{\log \phi_{i_m}(x)} = c \in \mathbb{R} \setminus \{0\}. \tag{4}$$
3. In the first case, a new index $R = (r_m + r_{m+1})/2$ is inserted in $\mathcal{S}$, the corresponding function $\exp(\phi_{i_p})$ enters $\mathcal{B}$ and is represented as e_R.
4. In the second case, we claim that $\exp(\phi_{i_p})$ is not merely *equivalent* to $\phi_{i_m}^c$, but equal to it, and that $c = 1$. Then the representation of $\exp(\phi_{i_p})$ is e_{r_m}.

We sketch the proof of our last claim, which is an inductive proof based on the algorithm we have just outlined and on the way $\mathcal{B}$ is constructed. In Equation (4), since ϕ_{i_m} is represented by some r_m in $\mathcal{S}$, it has one of the types e_n or $\exp(U)$ with U a function of the scale. In both cases, this implies that $\log \phi_{i_m}$ is equal to a monomial

$$e_{i_1}^{\beta_1} e_{i_2}^{\beta_2} \cdots e_{i_s}^{\beta_s},$$

with the functions $e_{i_j} \in \mathcal{B}$. On the other hand, ϕ_{i_p} belonging to the scale, it is also equal to some monomial

$$e_{j_1}^{\alpha_1} e_{j_2}^{\alpha_2} \cdots e_{j_k}^{\beta_k},$$

with the functions e_{j_l} belonging to $\mathcal{B}$. From the property of $\mathcal{B}$ given in equation (3), we deduce that to satisfy (4), these two monomials have to be equal.

Computation of the Logarithm. The computation of the logarithm has already been alluded to above. We can generalize the definition of the logarithm that was given in Section 2, so as to accommodate any asymptotic expansion. Consider the asymptotic expansion A defined in Equation (1). We can now define the logarithm in the case when the valuation of A is not zero: provided we can compute $\log \phi_{i_1}$, it clearly is

$$\log \phi_{i_1} + \log a_1 - \psi + \psi^2/2 - \psi^3/3 + \cdots,$$

with the same function ψ as before. The computation of $\log \phi_{i_1}$ is straightforward in this scale. The representation of ϕ_{i_1} as a Puiseux polynomial

$$e_{r_1}^{\alpha_1}(x) \cdot e_{r_2}^{\alpha_2}(x) \cdots e_{r_k}^{\alpha_k}(x),$$

gives for the logarithm

$$\alpha_1 \log(e_{r_1}) + \cdots + \alpha_k \log(e_{r_k}),$$

so that we the problem is reduced to computing logarithms of elements of $\mathcal{B}$, but this is obvious since an element of $\mathcal{B}$ is either of the type e_p with $p \in \mathbb{Z}$ or has been constructed by a former exponentiation of a function of the scale.

Example. We show how our program finds the asymptotic expansion of

$$\log\left(x + \frac{2e^{x \log^{3/2} x}}{x^x}\right), \quad \text{as } x \to \infty.$$

1. x is first translated into $e_0(x)$.
2. Then the second summand is considered, and the term $\exp(x \log^{3/2} x)$ is encountered. This is rewritten $\exp[e_0(x)e_{-1}^{3/2}(x)]$, which must lie between $e_1(x)$ and $e_2(x)$. As a consequence, this function enters the scale as $e_{3/2}(x)$.

3. The monomial $x^x = \exp(x \log x)$ is then handled. It is rewritten as $\exp[e_0(x)e_{-1}(x)]$. Thus it also has to lie between $e_1(x)$ and $e_2(x)$. Shackell's algorithm then helps deciding that this function grows faster than the previous one, and this function enters the scale as $e_{5/4}(x)$.
4. We then have to compute $\log(2e_{3/2}(x)/e_{5/4}(x) + e_0(x))$, which is
$$\log[e_{3/2}(x)] - \log[e_{5/4}(x)] + \log 2 - \sum_k \frac{(-1)^k}{k} \left(\frac{e_0(x)e_{5/4}(x)}{2e_{3/2}(x)}\right)^k,$$
the result is then obtained by recovering from their definitions the values of $e_{5/4}(x)$ and $e_{3/2}(x)$.

This is the actual output of our Maple program:

```
> gdev(log(x+2*exp(x*log(x)^(3/2))/x^x),x=infinity,5);
             3/2                                       x exp(x ln(x))
    (x ln(x)    ) + (- x ln(x)) + (ln(2)) + (1/2 ---------------)
                                                                3/2
                                                      exp(x ln(x)    )

                   2              2            3    x 3
                  x   exp(x ln(x))           x   (x  )
         + (- 1/8 ----------------) + (O(-----------------))
                              3/2 2                    3/2 3
                  exp(x ln(x)    )       exp(x ln(x)    )
```

5.3. OTHER SCALES

In a yet unpublished work, G. Gonnet and D. Gruntz [6] have devised an algorithm which performs asymptotic comparisons in very general scales. In one of their examples, they compare automatically

$$\exp(\exp(\exp(\Psi(\Psi(\Psi(x)))))) \text{ and } x,$$

where Ψ is the logarithmic derivative of Euler's Γ function. Such a comparison algorithm permits the construction of a very general asymptotic scale. However, it is not yet clear to the author what class exactly is handled by this new algorithm.

Another class of scales can be obtained by extending the exp-log scale we have presented, appealing to the theory of Hardy fields. Briefly, Hardy fields provide an algebraic setting to express and prove results about formal asymptotic expansions. Relevant work can be found in [1, 12, 13, and 16].

Conclusion

In this paper we have presented algorithms that perform manipulations of asymptotic expansions. The first step is to help make the program modular by reducing the operations on asymptotic expansions to operations on the set of coefficients and operations on the scale. The second step is to get as general an asymptotic scale as possible. Such a scale has been presented for exp-log functions. Its important features are that it is created dynamically and that it necessitates a minimal amount of comparisons (a costly operation which is handled by Shackell's algorithm).

Acknowledgements

This work was supported in part by the ESPRIT III Basic Research Action Programme of the E.C. under contract ALCOM II (#7141).

References

[1] Bourbaki, N. (1961) *Éléments de Mathématiques*, Chap. V: Fonctions d'une variable réelle (appendice), 2nd ed., Hermann, Paris, 36–55.
[2] Coste, M., and Roy, M. F. (1988) 'Thom's lemma, the coding of real algebraic numbers and the topology of semi-algebraic sets', *J. Symbolic Computation*, **5**, 121–129.
[3] Dahn, B. I., and Göring, P. (1986) 'Note on exponential-logarithmic terms', *Fundamenta Mathematicae*, **127**, 45–50.
[4] du Bois-Reymond, P. (1875) 'Ueber asymptotische Werthe, infinitäre Approximationen und infinitäre Auflösung von Gleichungen', *Mathematische Annalen*, **VIII**, 363–414, 574–576.
[5] Geddes, K. O., and Gonnet, G. H. (1989) 'A new algorithm for computing symbolic limits using hierarchical series', in P. Gianni (ed.), *Symbolic and Algebraic Computation*, vol. 358 of *Lecture Notes in Computer Science*, Springer-Verlag, New York, 490–495. Proceedings ISSAC'88, Rome.
[6] Gonnet, G., and Gruntz, D. (1992). Personal communication.
[7] Hardy, G. H. (1910) 'Orders of infinity', *Cambridge Tracts in Mathematics*, **12**.
[8] Loos, R. (1982) 'Computing in algebraic extensions', in B. Buchberger, G. E. Collins, and R. Loos (eds.), *Computer Algebra. Symbolic and Algebraic Computation*, vol. 4 of *Computing Supplementum*, Springer-Verlag, New York, 173–188.
[9] Monagan, M. B. (1989) *Signatures + Abstract Data Types = Computer Algebra – Intermediate Expression Swell*, PhD thesis, University of Waterloo.
[10] Richardson, D. (1968) 'Some undecidable problems involving elementary functions of a real variable', *J. Symbolic Logic*, **33**, **4**, 514–520.
[11] Richardson, D. (1992) 'The elementary constant problem', in P. Wang (ed.), *ISSAC'92*, ACM Press. To appear.
[12] Rosenlicht, M. (1983) 'Hardy fields', *J. Math. Anal. Appls.*, **93**, 297–311.
[13] Rosenlicht, M. (1987) 'Growth properties of functions in Hardy fields', *Trans. AMS*, **299**, **1**, 261–272.
[14] Salvy, B. (1991) *Asymptotique automatique et fonctions génératrices*, PhD thesis, École Polytechnique.
[15] Shackell, J. (1990) 'Growth estimates for exp-log functions', *J. Symbolic Computation*, **10**, 611–632.
[16] Shackell, J. (1991). 'Rosenlicht fields'. *Trans. AMS*, to appear.