UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Rapports de Recherche

N°1673

Programme 2

Calcul symbolique, Programmation et Génie logiciel

ASYMPTOTIC EXPANSIONS OF FUNCTIONAL INVERSES

Bruno SALVY
John SHACKELL

Mai 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

Asymptotic Expansions of Functional Inverses

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

John Shackell
University of Kent at Canterbury,
Canterbury,
Kent CT2 7NF,
England

Abstract

We study the automatic computation of asymptotic expansions of functional inverses. Based on previous work on asymptotic expansions, we give an algorithm which computes Hardy-field solutions of equations $f(y) = x$, with f belonging to a large class of functions.

Développements asymptotiques d'inverses fonctionnels

Résumé

Nous étudions le calcul automatique de développements asymptotiques d'inverses fonctionnels. À partir de résultats antérieurs sur les développements asymptotiques, nous donnons un algorithme qui calcule les solutions d'équations $f(y) = x$, pour des fonctions f appartenant à une classe étendue, et pourvu que ces solutions appartiennent à un corps de Hardy.

To appear in *Proceedings ISSAC'92*, P. Wang ed., ACM Press, 1992.

Asymptotic Expansions of Functional Inverses

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

John Shackell
University of Kent at Canterbury,
Canterbury,
Kent CT2 7NF,
England

Abstract

We study the automatic computation of asymptotic expansions of functional inverses. Based on previous work on asymptotic expansions, we give an algorithm which computes Hardy-field solutions of equations $f(y) = x$, with f belonging to a large class of functions.

Introduction

Asymptotics in symbolic computation have long been restricted to formal power series manipulations. In the recent years, new approaches have given rise to more and more general types of series (see [4, 14, 15]) and this opens up a new field of applications to computer algebra that we shall christen "symbolic asymptotics". In the same way as symbolic integration is based on differential algebra, the existing theory of symbolic asymptotics is based on Hardy fields. We shall review some properties of Hardy fields in Section 1. Two important notions for formal manipulations and for proving theorems were introduced in [19, 16], these are *nested forms* and *nested expansions*. Nested forms are expressions like

$$e^{\log^2 x \, e^{\sqrt{\log\log x}\,(l+\phi^{\langle 1\rangle}(x))}},$$

where l is a real constant and $\phi^{\langle 1\rangle}(x)$ tends to 0 when x tends to infinity (a formal definition will be given below). In certain cases, one can compute a nested form of $\phi^{\langle 1\rangle}(x)$, introducing a new function $\phi^{\langle 2\rangle}$ and then repeat the process, thus generating a sequence of nested forms; this sequence is called a *nested expansion*.

In previous papers by J. Shackell [15, 19, 16, 18, 17], algorithms are described that enable the computation of nested expansions for: *i)* exp-log functions, i.e. real functions of one variable composed by finitely many exponentials, logarithms and real algebraic functions, *ii)* Liouvillian functions and *iii)* solutions of algebraic differential equations (provided that these lie in a Hardy field).

In this paper, we give an algorithm to compute a nested expansion of $y(x)$, where $y(x)$ is a solution of

$$f(y) = x, \quad x \to \infty. \tag{1}$$

The class of allowable functions f, includes some which are not meromorphic at infinity, which means that the problem cannot always be solved by functional inversion of a power series. Equations of this type are encountered for instance when applying the saddle-point method to asymptotic estimates of Taylor coefficients of generating functions (see for instance [2, chap. 6]). In this context, one is interested by an integral of the form

$$\frac{1}{2i\pi}\oint \frac{F(z)}{z^{n+1}}\,dz,$$

and when the saddle-point method applies, the saddle-point is defined by

$$\rho\frac{F'(\rho)}{F(\rho)} - 1 = n.$$

One is then interested in the asymptotic behaviour of the solutions to this equation when n tends to infinity.

This paper is structured as follows: in Section 1, we recall the elementary theory of Hardy fields and prove a result concerning Hardy fields and functional inversion. Then in Section 2, we reduce the problem to the case when the inverse function tends to infinity. In Section 3 we show how to obtain the first asymptotic approximation to the inverse, and give a proof of its correctness. In the next section we describe the algorithm for the full asymptotic expansion and prove our main result which is that the algorithm computes the nested expansion of the inverse function. Then in Section 5, we describe the working of the algorithm on an example from de Bruijn's book [2, p. 25–28].

1 Basic properties of Hardy fields

We recall the part of the theory of Hardy fields that will be used in this paper. More information can be found in M. Rosenlicht's papers [10, 11, 12, 13].

Let $\mathcal{X}$ be the ring of germs at infinity of C^∞ functions. (Think of it as the set of possible asymptotic behaviours.) A *Hardy field* is a subring of $\mathcal{X}$ which is a field closed under differentiation.

The main constraint here is that non-zero elements of Hardy fields have to be invertible, and thus cannot have arbitrarily large zeros. Consequently, since their derivatives belong to the field, they have to be ultimately monotonic and tend to a possibly infinite limit. Also, differences of two (germs of) functions of a Hardy field belong to the field and are ultimately of constant sign, so that this field is ordered.

Theorem 1 *Let f be an element of a Hardy field which tends to infinity. Then there exists a unique inverse function g of f. It tends to infinity and moreover belongs to a Hardy field.*

Theorem 1 should be compared to a question asked by G. H. Hardy in [6, p.87]:

> *Whether or not it is true that, given an L-function* [exp-log function] ϕ *and its inverse $\overline{\phi}$, there must be an L-function ψ, such that $\overline{\phi} \sim \psi$, I cannot say; and, as I said in [5], I am very doubtful whether this is so.*

Theorem 1 does not answer Hardy's question but proves the weaker property that there is a function ψ *in a Hardy field* such that $\overline{\phi} \sim \psi$.

Proof. Since f belongs to a Hardy field and tends to infinity, its derivative cannot be identically zero, and so $f'(y)$ has no zeros for y sufficiently large. The existence of a unique inverse then follows from standard results. To prove that g lies in a Hardy field, let P be a polynomial over $\mathbb{R}$ and suppose that there exists a sequence of points $\{x_k\}$ tending to infinity at which $P(g, g', \ldots, g^{(n)})$ is zero. We show that $P \equiv 0$.

A simple induction, starting from the relation $g' = (f' \circ g)^{-1}$ shows that for each $n = 1, 2, \ldots$, the n-th derivative $g^{(n)}$ may be expressed in the form $g^{(n)} = Q_n(f' \circ g, f'' \circ g, \ldots, f^{(n)} \circ g)$, where Q_n is a rational function over $\mathbb{R}$. On substituting these into P, we obtain

$$P(g, g', \ldots, g^{(n)}) = R(g, f' \circ g, \ldots, f^{(n)} \circ g),$$

with R rational over $\mathbb{R}$. If we regard R as a function of g (i.e. we make a change of variable $y = g(x)$), then R belongs to the Hardy field generated by $\mathbb{R}(y)$ and $f(y)$ and vanishes at the points $\{g(x_k)\}$. But $g(x_k) \to \infty$ and hence $R(y) \equiv 0$. Thus $P \equiv 0$. It then follows easily that any rational function of g and its derivatives must ultimately be of constant sign and so g belongs to a Hardy field as required. This completes the proof of Theorem 1. $\square$

If f and g are two elements of a Hardy field tending to infinity, they are said to be *comparable* when there exists a positive integer n such that

$$f < g^n \quad \text{and} \quad g < f^n,$$

where the order is that of the field. Extending this by saying that $\pm f$ and f^{-1} are comparable and that two elements tending to a non-zero finite limit are comparable yields a decomposition of the Hardy field into equivalence classes called *comparability classes*; the comparability class of f is denoted by $\gamma(f)$. Given two functions f and g in the field, with $f, g \to \infty$, we write $\gamma(f) > \gamma(g)$ if $f > g^n$ for all $n \in \mathbb{N}$. This relation is independent of the representatives of the class, and setting $\gamma(1)$ as the smallest class gives a total order on the set of comparability classes. One should think of these classes as basic functions of an asymptotic scale. Their possibly finite number minus one is called *the rank* of the field.

An important special type of Hardy field, one of finite rank, closed under $f \to f^c$ for all real c, was considered by M. Rosenlicht in [12]. In [19], such fields were called *Rosenlicht fields* and it was shown that *any element of a Rosenlicht field has a nested expansion.* We now define these latter objects more precisely.

We use the classical notations $l_k(x)$ (or sometimes just $l_k x$) for the logarithm iterated k times and likewise $e_k(x)$ for the iterated exponential. A *nested form* is then a finite sequence $\{(s_i, \epsilon_i, m_i, d_i, \phi_i), i = 1 \ldots n\}$, where s_i and m_i are non-negative integers, ϵ_i is ± 1, d_i is a real number, and ϕ_i is an element of a Hardy field. Such a sequence represents

$$\phi = e_{s_1}^{\epsilon_1}(l_{m_1}^{d_1}(x)\phi_1(x)),$$

and recursively

$$\phi_{i-1}(x) = e_{s_i}^{\epsilon_i}(l_{m_i}^{d_i}(x)\phi_i(x)), \quad i = 2, \ldots, n,$$

with the additional constraint that ϕ_n tends to a finite limit, that each ϕ_i is of a smaller order of growth than l_{m_i} (i.e. $\gamma(\phi_i) < \gamma(l_{m_i})$). We also impose some conditions ensuring, for example, that the expression cannot be reduced by simplifying an exp(log(.)); specifically we require $d_n \neq 1$ unless $s_n = 0$ or $m_n = 0$ and also $d_i > 0$ unless $s_i = 0$. The number n will be called *the length* of the nested form. Having thus defined a nested form, one defines a *nested expansion* as a sequence of nested forms F_k, such that F_{k+1} is the nested form of $\phi^{\langle k+1\rangle} = \phi^{\langle k\rangle}_{n_k} - \lim \phi^{\langle k\rangle}_{n_k}$, where we have set $\phi^{\langle 0\rangle} = \phi$.

What makes these nested expansions very useful is that one can compute easily their ordering, their exponential and their logarithm.

2 Singularities of f

In the quest for the possible asymptotic forms of $y(x)$ satisfying (1), the first task is to find the singularities of f, since the possible limits of $y(x)$ form a subset of these. Unfortunately we know of no algorithm for doing this in the most general case, when f is defined by an algebraic differential equation. We deal with this problem by introducing a new constructor `SingularityOf`, akin to Maple's `RootOf`. However, there are several special cases where one can be more specific.

Rational and algebraic functions. Then the singularities are algebraic numbers and thus known "explicitly".

Exp-log functions. Then the set of possible singularities can be reduced to a set of `RootOf`s, by a direct recursive algorithm.

Solutions of linear differential equations with exp-log coefficients. Then it is known that the singularities of the solutions are either at infinity, or at roots of the leading coefficient, or at singularities of the coefficients. Thus the set of possible singularities can be obtained automatically in terms of `RootOf`s.

One particular problem that occurs when dealing with symbolic asymptotics is that of deciding whether a constant is zero or not. With exp-log functions, for example, this is related to some difficult conjectures in number theory and could be undecidable (see [7, 8, 9]). A fuller discussion of the constant problem is beyond the scope of this paper. It suffices to say that in view of the above, we require the use of an oracle which is able to decide the signs of constants. These constants will have to include the singularities of f given by the constructor `SingularityOf`. Note that having a solution in terms of `SingularityOf` is useful from the practical point of view, since one can then use numerical evaluation to get a partial answer to the constant problem, just as one often does when constants like π or γ are involved. For other cases, like $f(y) = \cos(\pi y) + 1/\sin(y)$, we assume that at some stage the program or the user will decide that one particular $k\pi$ is of interest.

Suppose we have found a candidate singularity s. Our next step will then be to change the variable depending on whether s is finite or not. If it is finite, we study both the changes of unknown function[1] $y = s \pm 1/Y$; if $s = -\infty$ then we change the sign of y. In all these cases, we have reduced the problem to the study of (1) when y tends to $+\infty$. Note that if s is a finite singularity and $f(s - 1/Y)$ belongs to a Hardy field, and $f \to \infty$ as $y \to s_-$ for example, it follows from Theorem 1, using the change of variable, that there is a unique Hardy field solution of equation (1) which tends to s_-. Similar remarks apply to s_+. However not all equations of type (1) have solutions lying in Hardy fields. Two simple examples where this is not the case are given by $-y^2 = x$ and $y^{-1} + \sin(y^{-1}) = x$.

[1] Everything we consider here lies in the real domain.

3 The First Nested Form

In this section, we restrict ourselves to the case when y tends to infinity in equation (1). We shall also require that f (or rather its germ at infinity) belongs to a Hardy field. We furthermore require that nested forms can be computed for f itself and various expressions containing it. To be precise, let $\mathcal{F}$ be the set of functions obtained from y and $f(y)$ by application of arithmetic operations and the operations $F \to \exp(F)$ and $F \to \log|F|$. Then we assume that we can compute nested forms for elements of $\mathcal{F}$. This will be the case if f is an exp-log function (see [15]) or more generally if f belongs to an asymptotic field (see [18] for the definition and details, and also [17]).

Algorithm: First Nested Form.
Input: a function $f(y) \in \mathcal{F}$ given as a nested form.
Output: the first nested form for $y(x)$, the Hardy-field solution of $f(y) = x$ tending to infinity at infinity.

Suppose then that

$$f(y) = e_s\left(l_m^d y \cdot f_1(y)\right),$$

with $\gamma(f_1) < \gamma(l_m)$. The exponent of e_s is $+1$ here since by hypothesis f tends to infinity. Inverting equation (1) we obtain an implicit equation

$$y = e_m(l_s^{1/d} x \cdot g_1(y)), \qquad (2)$$

where for simplification we have set $g_1(y) = f_1^{-1/d}(y)$. This last equation will be the basis of our iteration. The algorithm then outputs $(m, 1, s, 1/d)$ and calls the iteration step on $g_1(y)$. Note that $g_1 \in \mathcal{F}$ and that $\gamma(g_1) < \gamma(l_m)$.

Iteration step. The input of this step of the algorithm is a function $g_i(y) \in \mathcal{F}$ with $\gamma(g_i) < \gamma(l_m)$. Its output is the nested form of $g_i(y(x))$.

If $g_i(y)$ has a finite non-zero limit c (at infinity), then $g_i(y) = c + G(y)$, with $G(y) = o(1)$ as $x \to \infty$, and the nested form for $g_i(y(x))$ is $g_i(y(x)) = c + G(y(x))$. The algorithm ouputs this and then stops.

If $g_i(y)$ tends to zero or infinity, we begin by computing the first part of the nested expansion of $g_i(y)$:

$$g_i(y) = e_\sigma^\epsilon[l_\mu^\delta y \cdot G(y)], \tag{3}$$

and we substitute (2) into this giving

$$g_i(y) = e_\sigma^\epsilon \left[l_\mu^\delta [e_m(l_s^{1/d} x \cdot g_1(y))] \cdot G(y) \right]. \tag{4}$$

We next note that since $\gamma(g_i) = \gamma(e_\sigma(l_\mu^\delta \cdot G)) < \gamma(l_m)$, we must in this case have $\mu > m$. We can therefore rewrite (4) as follows.

$$\begin{aligned} g_i(y) &= e_\sigma^\epsilon[l_{\mu-m-1}^\delta(\frac{1}{d} l_{s+1} x + \log g_1(y)) \cdot G(y)], \\ &= e_\sigma^\epsilon[l_{\mu-m+s}^\delta x \cdot h(x,y) \cdot G(y)], \end{aligned} \tag{5}$$

where for $\mu - m = 1$ we have

$$h(x,y) = \left(\frac{1}{d} + \frac{\log g_1(y)}{l_{s+1} x} \right)^\delta, \tag{6}$$

and for $\mu - m > 1$, we have the asymptotic estimate

$$h(x,y) = 1 + O\left(\frac{1}{l_{\mu-m+s} x} \right). \tag{7}$$

Note that $\log g_1(y)/l_{s+1}x \to 0$, since $\gamma(g_1(y)) < \gamma(l_m y) = \gamma(l_s x)$. The output of this step is $(\sigma, \epsilon, \mu - m + s, \delta)$ representing the estimate

$$g_i(y) = e_\sigma^\epsilon[l_{\mu-m+s}^\delta x \cdot g_{i+1}(y)],$$

where we have set $g_{i+1}(y) = G(y) \cdot h(f(y), y)$. We have $\gamma(g_{i+1}) < \gamma(g_i) < \gamma(l_m)$, and since

$$g_{i+1}(y) = \frac{l_\sigma(g_i^\epsilon(y))}{l_{\mu-m+s}^\delta(f(y))},$$

g_{i+1} belongs to the asymptotic field $\mathcal{F}$, and thus can be fed back into the algorithm.
End of the algorithm.

We have to prove that the process described above yields a nested form for $y(x)$ in a finite number of steps. We begin with the following lemma.

Lemma 1 *Let ϕ be an element of $\mathcal{F}$, and let its nested form be*

$$\phi(y) = e_{s_1}^{\epsilon_1}[l_{m_1}^{d_1} y \cdot e_{s_2}^{\epsilon_2}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n}^{d_n} y \cdot (c + \phi^{\langle 1 \rangle}(y))] \cdots]],$$

with $\phi^{\langle 1 \rangle}(y) = o(1)$. Let ψ be a member of $\mathcal{F}$ such that $\psi = o(\phi)$. Then the nested form of $\phi + \psi$ is identical to the nested form of ϕ except possibly in its last terms:

$$\phi(y) + \psi(y) = e_{s_1}^{\epsilon_1}[l_{m_1}^{d_1} y \cdot e_{s_2}^{\epsilon_2}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n}^{d_n} y \cdot (d + \theta(y))] \cdots]],$$

with d a non-zero constant, $\theta \to 0$ and $\theta \in \mathcal{F}$.

Proof. Note first that by rewriting $\phi^{\langle 1 \rangle}(y)$, one gets that $\phi^{\langle 1 \rangle} \in \mathcal{F}$. The proof is then by induction on n.
If $n = 0$, then $\phi(y) + \psi(y) = c + \phi^{\langle 1 \rangle}(y) + \psi(y)$ is a nested form of $\phi + \psi$, and $\phi^{\langle 1 \rangle} + \psi \in \mathcal{F}$.
If $n > 0$, let $\phi_1(y) = e_{s_2}^{\epsilon_2}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n}^{d_n} y \cdot (c + \phi^{\langle 1 \rangle}(y))] \cdots]$. Then

$$\phi(y) + \psi(y) = e_{s_1}^{\epsilon_1}[l_{m_1}^{d_1} y \cdot \phi(y)](1 + \psi(y)/\phi_1(y)).$$

If $s_1 = 0$, this can be rewritten

$$l_{m_1}^{d_1} y \cdot \left(\phi_1(y) + \frac{\phi_1(y)\psi(y)}{\phi(y)} \right),$$

and the induction hypothesis applies to the inner sum. Otherwise, having rewritten the sum as

$$\begin{aligned} \phi(y) + \psi(y) = e_1^{\epsilon_1}\{ & e_{s_1-1}[l_{m_1}^{d_1} y \cdot \phi(y)] + \\ & \epsilon_1 \log(1 + \psi(y)/\phi_1(y))\}, \end{aligned}$$

an induction on s_1 leads to the result. □

We need a similar lemma for the product by a function with a non-zero finite limit.

Lemma 2 *Let ϕ be as in Lemma 1, and let ψ be a member of $\mathcal{F}$ which tends to a finite limit $\lambda > 0$. Then the nested form of $\phi \cdot \psi$ is identical to the nested form of ϕ except possibly in its last terms. Moreover the error term belongs to $\mathcal{F}$.*

Proof. We have

$$\psi(y) = \lambda + \psi^{\langle 1 \rangle}(y),$$

where $\psi^{\langle 1 \rangle}(y) \to 0$, and we first note that it is sufficient to prove the lemma for $\psi(y) = \lambda$, because the result will then follow from Lemma 1. We now proceed by induction on the length n of the nested form of ϕ.
If $n = 0$, then

$$\lambda\phi(y) = \lambda c + \lambda\phi^{\langle 1 \rangle}(y)$$

is a nested form of the product.
If $n > 0$ then

$$\lambda\phi(y) = \lambda e_{s_1}^{\epsilon_1}[l_{m_1}^{d_1} y \cdot \phi_1(y)].$$

If $s_1 = 0$ then the induction applies to $\lambda\phi_1$, otherwise we rewrite the product as

$$e_1^{\epsilon_1}(e_{s_1-1}[l_{m_1}^{d_1} y \cdot \phi_1(y)] + \epsilon_1 \log\lambda),$$

and we can apply Lemma 1 to the inner sum. This completes the proof of the lemma. □

Next we examine the effect of substituting an identity of the form (2) into a nested form.

Lemma 3 *Let ϕ be as in Lemma 1. Let $y(x)$ satisfy $y = e_m(l_s^{1/d} x\, \lambda(y))$, with $\lambda \in \mathcal{F}$, $\gamma(\lambda(y(x))) < \gamma(l_s x)$ and $m_1 > m$. Then a nested form for $\phi(y(x))$ is*

$$\phi(y(x)) = e_{s_1}^{\epsilon_1}[l_{m_1-m+s}^{d_1} x\, e_{s_2}^{\epsilon_2}[l_{m_2-m+s}^{d_2} x\, e_{s_3}^{\epsilon_3}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n-m+s}^{d_n} x\,(d+\theta(x,y(x)))]\cdots]]],$$

where d is a non-zero constant, $\theta \to 0$ and $\theta(f(y),y) \in \mathcal{F}$.

Proof. Again we use induction on n. The case $n = 1$ is already clear from the presentation of the algorithm above and we therefore concentrate on the induction step.

Suppose then that the result holds when n is replaced by $n-1$ and let

$$\phi_1(y) = e_{s_2}^{\epsilon_2}[l_{m_2}^{d_2} y\, e_{s_3}^{\epsilon_3}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n}^{d_n} y\,(c+\phi^{\langle 1\rangle}(y))]\cdots]].$$

By the induction hypothesis,

$$\phi_1(y(x)) = e_{s_2}^{\epsilon_2}[l_{m_2-m+s}^{d_2} x\, e_{s_3}^{\epsilon_3}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n-m+s}^{d_n} x\,(d+\theta(x,y(x)))]\cdots],$$

with $\theta(f(y),y) \in \mathcal{F}$. But

$$\begin{aligned}\phi(y(x)) &= e_{s_1}^{\epsilon_1}(l_{m_1}^{d_1} y(x)\phi_1(y(x)))\\ &= e_{s_1}^{\epsilon_1}(l_{m_1-m+s}^{d_1} x(C+o(1))\phi_1(y(x))),\end{aligned}$$

as above (where C is a non-zero constant), and the result now follows from Lemmas 1 and 2. □

Theorem 2 *Let $f \in \mathcal{F}$. Then one can compute the nested form for any solution y of equation (1) in a neighbourhood of $+\infty$. In particular, if the nested form of f is*

$$f(y) = e_{s_1}[l_{m_1}^{d_1} y\, e_{s_2}^{\epsilon_2}[l_{m_2}^{d_2} y\, e_{s_3}^{\epsilon_3}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n}^{d_n} y\,(c+f^{\langle 1\rangle}(y))]\cdots]]], \tag{8}$$

with c a non-zero real number and $f^{\langle 1\rangle}(y) \to 0$ as $y \to \infty$, then the nested form of y is

$$y = e_{m_1}[l_{s_1}^{1/d_1} x\, e_{s_2}^{-\epsilon_2}[l_{m_2-m_1+s_1}^{d_2'} x\, e_{s_3}^{\epsilon_3}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n-m_1+s_1}^{d_n'} x\,(d+g^{\langle 1\rangle}(x,y(x)))]\cdots]]], \tag{9}$$

with d a non-zero real number and $g^{\langle 1\rangle}(x,y(x)) \to 0$ as $x \to \infty$. Moreover d_2' is equal to d_2/d_1 if $s_2 = 0$ and to d_2 otherwise, d_3' is equal to d_3/d_1 if $s_2 = s_3 = 0$ and to d_3 otherwise, and so on. Furthermore $g^{\langle 1\rangle}(f(y),y) \in \mathcal{F}$.

Proof. Inversion of (8) yields

$$y(x) = e_{m_1}(l_{s_1}^{1/d_1} x \cdot g_1(y(x))),$$

as in (2). On applying Lemma 3 with $\phi = g_1$, we obtain

$$y(x) = e_{m_1}[l_{s_1}^{1/d_1} x\, e_{s_2}^{-\epsilon_2/d_1}[l_{m_2-m_1+s_1}^{d_2} x\, e_{s_3}^{\epsilon_3}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n-m_1+s_1}^{d_n} x\,(b+\xi(x,y(x)))]\cdots],$$

where b is a non-zero constant, $\xi \in \mathcal{F}$ and $\xi \to 0$. If $s_2 \neq 0$, then

$$\begin{aligned}&e_{s_2}^{-\epsilon_2/d_1}[l_{m_2-m_1+s_1}^{d_2} x\, e_{s_3}^{\epsilon_3}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n-m_1+s_1}^{d_n} x\,(b+\xi(x,y(x)))]\cdots]] =\\ &\exp\{(-\epsilon_2/d_1)e_{s_2-1}[l_{m_2-m_1+s_1}^{d_2} x\, e_{s_3}^{\epsilon_3}[\cdots e_{s_n}^{\epsilon_n}[l_{m_n-m_1+s_1}^{d_n} x\,(b+\xi(x,y(x)))]\cdots]]\},\end{aligned}$$

and the result follows from Lemma 2. The proof of Theorem 2 is thus complete and it is clear that our algorithm gives a nested form for $y(x)$ in a finite number of steps. □

4 The full nested expansion of $y(x)$

We would like to be able to obtain the nested form for $g^{\langle 1\rangle}(y)$ in (9) and also subsequent forms in the nested expansion of $y(x)$. We shall denote these by $g^{\langle k+1\rangle}(y) = g_{n_k}^{\langle k\rangle}(f(y),y) - \lim g_{n_k}^{\langle k\rangle}(f(y),y)$. The obvious tactic is to substitute an existing asymptotic expression for $y(x)$ into $g^{\langle 1\rangle}(y)$. The only problem is that now when we substitute (2) into $g^{\langle 1\rangle}$—and more generally into $g_i^{\langle k\rangle}$—in the iteration step, the inequality $\mu > m$ may no longer hold. We therefore consider the modifications necessary to our arguments of the previous section when $\mu \le m$.

If $\mu = m$ then (4) becomes

$$g_i^{\langle k\rangle}(y) = e_\sigma^\epsilon[l_s^{\delta/d} x \cdot g_{i+1}^{\langle k\rangle}(y)], \tag{10}$$

where we have set $g_{i+1}^{\langle k\rangle}(y) = g_1^\delta(y)\,G(y)$, which is fed back to the algorithm. The output of this step is $(\sigma, \epsilon, s, \delta/d)$. Of course, if $\sigma > 0$, $s > 0$, $\delta = d$ and $g_{i+1}^{\langle k\rangle} \to a$, where $a \in \mathbb{R}\backslash\{0\}$, then we need to rewrite the expression for $g_i^{\langle k\rangle}$ as $e_{\sigma-1}^\epsilon[l_{s-1}^a \exp\{l_s(f(y))(g_{i+1}^{\langle k\rangle} - a)\}]$, so as to satisfy the conditions for a nested form. Likewise if $\sigma > 1$, $s > 1$, $a = 1$ and $\exp\{l_s(f(y))(g_{i+1}^{\langle k\rangle} - a)\}$ tends to a finite limit, then a further rewrite is necessary, and so on. However this causes no serious difficulty.

If $\mu < m$, then (4) gives

$$g_i^{\langle k\rangle}(y) = e_{\sigma+1}^\epsilon\left[e_{m-\mu-1}(l_s^{1/d} x \cdot g_1(y))\right.$$

$$\left(\delta + \frac{\log G(y)}{e_{m-\mu-1}(l_s^{1/d} x g_1(y))}\right)\Bigg],$$
$$= e^{\epsilon}_{\sigma+m-\mu}[l_s^{1/d} x \cdot g_1(y) \cdot h(x,y)].$$

If $m-\mu = 1$, then

$$h(x,y) = \delta + \frac{\log G(y)}{l_s^{1/d} x g_1(y)},$$

while if $m-\mu > 1$, we have the asymptotic estimate

$$h(x,y) = 1 + O\left(\frac{1}{l_s^{1/d} x g_1(y)}\right).$$

The output of this step is $(\sigma+m-\mu, \epsilon, s, 1/d)$ representing the estimate

$$g_i^{\langle k \rangle}(y) = e^{\epsilon}_{\sigma+m-\mu}[l_s^{1/d} x \cdot g_{i+1}^{\langle k \rangle}(y)], \tag{11}$$

where we have set $g_{i+1}^{\langle k \rangle}(y) = g_1(y) \cdot h(f(y), y)$. The proof that $g_{i+1}^{\langle k \rangle} \in \mathcal{F}$ is easily done by rewriting $g_{i+1}^{\langle k \rangle}$ from (11) as a function of $g_i^{\langle k \rangle}(y)$ and $f(y)$. We can therefore feed $g_{i+1}^{\langle k \rangle}$ back into the algorithm as before.

We can thus summarize our full algorithm as follows.

Algorithm: Full Nested Form.

1. Determine the singularities of f to be investigated. Treat each singularity in turn, replacing y by a new variable where appropriate, to reduce to the case when $y \to \infty$.

2. Compute the first part of the nested form of f. Check that it tends to infinity, otherwise reject the singularity.

3. Invert the first part of the nested form of f to obtain an expression of the form (2). Then substitute (2) into itself, as in Algorithm: First Nested Form. This yields a nested form and an error term $l_0 + g^{\langle 1 \rangle}(y)$.

4. Apply the iteration step to $g^{\langle k \rangle}(y)$, yielding a nested form, a non-zero limit l_k and a function $g^{\langle k+1 \rangle}(y)$, for which this step can be repeated.

Note that in the case of functional inversion of a function which is meromorphic at infinity, our algorithm reduces to formal inversion of formal power series, and this means that it can compute (although not very efficiently) inversion of power series.

We now give our main theorem.

Theorem 3 *The above algorithm computes each nested form of the nested expansion of $y(x)$ in a finite number of iterations.*

Proof. We have already proved that the algorithm successfully computes the first nested form. As regards the later nested forms we have to ensure two things. Firstly we must show in (10) and (11) that $\gamma(g_{i+1}^{\langle k \rangle}(y(x))) < \gamma(l_s(x))$, and secondly we must prove that after a finite number of stages, we reach the stage $i = n_k$ when $g_{n_k}^{\langle k \rangle}$ tends to a finite non-zero limit.

When $\mu = m$, we have

$$\begin{aligned}\gamma(g_{i+1}^{\langle k \rangle}(y)) &= \gamma(g_1^{\delta}(y) G(y)),\\ &\le \max(\gamma(g_1(y)), \gamma(G(y))).\end{aligned}$$

On the other hand, $\gamma(l_s^{\delta/d} x) = \gamma(l_s x) = \gamma(l_m^d y g_1(y)) = \gamma(l_m y)$. By definition of the nested form, we have $\gamma(l_m y) > \gamma(g_1(y))$ and since $\mu = m$ we also have $\gamma(l_m y) > \gamma(G(y))$. So $\gamma(g_{i+1}^{\langle k \rangle}(y(x))) < \gamma(l_s(x))$ as required.

For the case when $\mu < m$, we already have that $\gamma(g_1(y)) < \gamma(l_s(x))$ and because of the form of $h(x,y)$, we immediately obtain that $\gamma(g_{i+1}^{\langle k \rangle}(y(x))) < \gamma(l_s(x))$ in this case also.

The fact that the computation of each particular nested form in the expansion of $y(x)$ terminates in a finite number of steps can be seen as follows. When we are computing a particular nested form suppose that at some stage we substitute $y = e_m(l_s x\, g_1(y))$ into a form $g_i^{\langle k \rangle}(y) = e_\sigma(l_\mu y\, G(y))$. Then the input of the next stage is $g_{i+1}^{\langle k \rangle}(y)$ with $\gamma(g_{i+1}^{\langle k \rangle}) < \gamma(l_\mu)$ and so the value of μ strictly increases as we go from one substitution to the next. Since m remains fixed, we must eventually reach the stage when $\mu > m$, unless the computation first ceases by reaching the stage when some $g_i^{\langle k \rangle}$ tends to a non-zero constant. But once $\mu > m$, Lemma 3 applies and so we always get termination for each nested form in a finite number of stages. □

5 Example

We shall work out the example given in de Bruijn's book [2] on pages 25–26:

$$ye^y = x.$$

In this case, the nested form of f is computed by the algorithm in [15] as

$$f(y) = e^{y[1+\log y/y]}.$$

The estimate produced by the first step of the algorithm is

$$y = \log x \left[1 - \frac{\log y}{y + \log y}\right], \tag{12}$$

which yields the first level of the nested expansion:

$$y_1 = \log x(1 + o(1)).$$

The second step starts by computing a nested form of $\log y/(y + \log y)$ as $y \to \infty$:

$$\frac{\log y}{y + \log y} = \frac{\log y}{y}\left(1 - \frac{\log y}{y + \log y}\right). \tag{13}$$

Now the algorithm substitutes (12) into the dominant part of (13) yielding the exact form:

$$y = \log x\left[1 - \frac{\log\log x}{\log x}\left(1 - \frac{\log[1 + \log y/y]}{\log(y + \log y)}\right)\right]. \tag{14}$$

This gives us the second level of the nested expansion:

$$y_2 = \log x - \log\log x(1 + o(1)),$$

and we apply once again the same process. We get a nested form for the last quotient by the algorithm of [15]:

$$\frac{\log[1 + \log y/y]}{\log(y + \log y)} = \frac{1}{y}(1 + \phi(y)),$$

with $\phi(y)$ a function tending to 0 at infinity, whose expression is too messy to be reproduced here. Then we substitute (12) into the dominant part of this, which gives an exact form:

$$y = \log x\left[1 - \frac{\log\log x}{\log x}\left(1 - \frac{1}{\log x}(1 + \psi(y))\right)\right],$$

hence the third level of the nested expansion:

$$y_3 = \log x - \log\log x + \frac{\log\log x}{\log x}(1 + o(1)),$$

and so on, each new step producing one more term.

Although this method is clearly more cumbersome than the approach described in [2], it has the advantage of being purely automatic and general. Besides, the algorithm for nested forms in [15] was not designed with complexity in mind, and will hopefully be improved complexity wise in the future.

Conclusion

The calculus of nested form which has been started in [15] proves to be both a computational and a mathematical tool. It becomes possible to state and prove algorithms on a class of asymptotic expansions much larger than mere formal power series. We feel that symbolic asymptotics is a young field of symbolic computation which has plenty of potential applications. Functional inversion is one of them and others should follow, such as asymptotic expansions of integrals or of coefficients of generating functions (see [14] for ideas on that and [3] for applications to the automatic analysis of algorithms).

Acknowledgements. The second author would like to thank INRIA-Rocquencourt, and in particular Philippe Flajolet and Bruno Salvy, for their hospitality during the week 24-30 November, when the groundwork for this paper was laid. 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. *Éléments de Mathématiques.* Hermann, 1951, ch. V: Fonctions d'une variable réelle. Appendice, pp. 36–55. Second edition, 1961.

[2] de Bruijn, N. G. *Asymptotic Methods in Analysis.* Dover, 1981. A reprint of the third North Holland edition, 1970 (first edition, 1958).

[3] Flajolet, P., Salvy, B., and Zimmermann, P. Automatic average–case analysis of algorithms. *Theoretical Computer Science, Series A 79*, 1 (Feb. 1991), 37–109.

[4] Geddes, K. O., and Gonnet, G. H. A new algorithm for computing symbolic limits using hierarchical series. In *Symbolic and Algebraic Computation* (1989), vol. 358 of *Lecture Notes in Computer Science*, pp. 490–495. Proceedings ISSAC'88, Rome.

[5] Hardy, G. H. Orders of infinity. *Cambridge Tracts in Mathematics 12* (1910).

[6] Hardy, G. H. Properties of logarithmico-exponential functions. *Proceedings of the London Mathematical Society 10*, 2 (1911), 54–90.

[7] Moses, J. Algebraic simplification: a guide for the perplexed. *Communications of the ACM 14*, 8 (Aug. 1971), 527–537.

[8] Richardson, D. Some undecidable problems involving elementary functions of a real variable. *The Journal of Symbolic Logic* (1968), 514–520.

[9] Richardson, D. Towards computing non algebraic cylindrical decompositions. In *ISSAC'91* (New York, 1991), ACM Press, pp. 247–255.

[10] Rosenlicht, M. Hardy fields. *Journal of Mathematical Analysis and Applications 93* (1983), 297–311.

[11] Rosenlicht, M. The rank of a Hardy field. *Transactions of the American Mathematical Society 280*, 2 (1983), 659–671.

[12] Rosenlicht, M. Rank change on adjoining real powers to Hardy fields. *Transactions of the American Mathematical Society 284*, 2 (1984), 829–836.

[13] Rosenlicht, M. Growth properties of functions in Hardy fields. *Transactions of the American Mathematical Society 299*, 1 (1987), 261–272.

[14] Salvy, B. *Asymptotique automatique et fonctions génératrices.* Ph. D. thesis, École Polytechnique, 1991.

[15] Shackell, J. Growth estimates for exp-log functions. *Journal of Symbolic Computation 10* (Dec. 1990), 611–632.

[16] Shackell, J. Computing asymptotic expansions for elements of hardy fields. Preprint, 1991.

[17] Shackell, J. Extensions of asymptotic fields via analytic functions. Preprint, 1991.

[18] Shackell, J. Limits of liouvillian functions. Preprint, 1991.

[19] Shackell, J. Rosenlicht fields. To appear in *Transactions of the American Mathematical Society*, 1991.