Meta-expansion of transseries*

Joris van der Hoeven

CNRS, Dépt. de Mathématiques (Bât. 425)
Université Paris-Sud
91405 Orsay Cedex
France
Email: joris@texmacs.org

December 9, 2008

The asymptotic behaviour of many univariate functions can only be expressed in generalized asymptotic scales, which are not merely formed of powers of a single variable. The computation of asymptotic expansions of functions in such generalized scales may lead to infinite cancellations, which complicate the design and implementation of practical algorithms. In this paper, we introduce a new heuristic technique of "meta-expansions", which is both simple and efficient in practice, even though the answers are not guaranteed to be correct in general.

Keywords: transseries, asymptotic expansion, algorithm, zero-test

A.M.S. subject classification: 40-04, 41A60, 68W30, 03C64, 03D60, 65B99

1. Introduction

The asymptotic behaviour of many univariate functions can only be expressed in generalized asymptotic scales, which are not merely formed of powers of a single variable.

It was already noticed by Hardy [Har10, Har11] that many interesting functions arising in combinatorics, number theory or physics can be expanded w.r.t. scales formed by so called exp-log functions or L-functions. An *exp-log function* is constructed from an indeterminate x and the real numbers using the field operations, exponentiation and logarithm. An *L-function* is defined similarly, by adding algebraic functions to our set of building blocks.

However, the class of functions which can be expanded with respect to a scale formed by exp-log functions (or L-functions) is not stable under several simple operations such as integration or functional inversion [Sha93, vdH97, MMvdD97]. More recently, exp-log functions have been generalized so as to allow for expressions with infinite sums, giving rise to the notion of transseries [DG86, É92]. In section 2, we briefly recall some of the most important definitions and properties. For more details, we refer to [vdH06c].

Given an explicit expression (such as an exp-log function), or the solution to an implicit equation, an interesting question is how to find its asymptotic expansion automatically. When working with respect to a generalized asymptotic scale, such as $\{x^\alpha \,\mathrm{e}^{\beta x} : \alpha, \beta \in \mathbb{R}\}$ at infinity ($x \to \infty$), even simple expressions can lead to infinite cancellations:

$$\begin{aligned}\frac{1}{1-\frac{1}{x}-\frac{1}{\mathrm{e}^x}}-\frac{1}{1-\frac{1}{x}} &= \left(1+\frac{1}{x}+\frac{1}{x^2}+\cdots+\frac{1}{\mathrm{e}^x}+\frac{1}{x\,\mathrm{e}^x}+\cdots\cdots\right)-\left(1+\frac{1}{x}+\frac{1}{x^2}+\cdots\right)\\ &\approx \frac{1}{\mathrm{e}^x}+\frac{1}{x\,\mathrm{e}^x}+\frac{1}{x^2\,\mathrm{e}^x}+\cdots.\end{aligned}$$

In many cases, the detection of infinite cancellations can be reduced to the zero-test problem in a suitable class of functions [Sha90, GG92, RSSvdH96, Sal91, vdH97].

∗. This work has partially been supported by the ANR Gecko project.

However, zero-test problems are often very hard. In the case of exp-log functions, a complete zero-test is only known if Schanuel's conjecture holds [Ric97, vdH98, vdHS06, Ric07]. If we want to expand more general expressions or more general solutions to differential or functional equations, the corresponding zero-test problem tends to get even harder. Consequently, the zero-testing tends to complicate the design of mathematically correct algorithms for more general asymptotic expansions. From the practical point of view, the implementation of robust zero-tests also requires a lot of work. Moreover, mathematically correct zero-tests tend to monopolize the execution time.

In this paper, we will investigate an alternative, more heuristic approach for the computation of asymptotic expansions. We will adopt a similar point of view as a numerical analyst who conceives a real number as the limit of a sequence of better and better approximations. In our case, the asymptotic expansion will be the limit of more and more precise polynomial approximations, where the monomials are taken in the asymptotic scale. As is often the case in numerical analysis, our approach will only be justified by informal arguments and the fact that it seems to work very well in practice.

Besides the analogy with numerical analysis there are a few additional interesting points which deserve to be mentioned. First of all, a finite sequence $\check{f}_{;0}, \check{f}_{;1}, \ldots$ of better and better approximations gives rise to a second sequence $\check{f}_0 = \check{f}_{;0}$, $\check{f}_1 = \check{f}_{;1} - \check{f}_{;0}$, $\check{f}_2 = \check{f}_{;2} - \check{f}_{;1}$, ..., which can itself be encoded by a generating series

$$\check{f}(z) = \sum_{n \in \mathbb{N}} \check{f}_n z^n.$$

The computation of the expansion of $f = \lim_{n \to \infty} \check{f}_{;n}$ can thus be re-interpreted as the computation of the expansion of $\check{f}$, which we therefore regard as the "meta-expansion" of f. This technique will be detailed in section 3. Additional complications arise in the context of transseries, because the elements of the asymptotic scale are themselves exponentials of other transseries. The computation of meta-expansions for transseries will be detailed in sections 4 and 5.

A second interesting aspect of meta-expansions is that we may operate on the meta-expansion without changing the underlying expansion. In a complex computation involving lots of auxiliary series, this provides some meta-control to the user. For instance, some subexpressions can be computed with more accuracy (or less accuracy) and one can focus on a specific range of terms. Techniques for the acceleration of convergence play a similar role in numerical analysis [PTVF07, Section 5.3]. Another operation, called "stabilization", removes those terms in the expansions $\check{f}_n$ which change at every few steps. After stabilization, we tend to compute only terms which occur in the final expansion of f, even though they usually appear in a different order. In particular, stabilization gives rise to a heuristic zero-test. Meta-operations on meta-expansions will be discussed in section 6.

One motivation behind the present paper was its application to the asymptotic extrapolation of sequences by transseries [vdH06a]. This application requires the computation of discrete sums and products of transseries. In section 7, we have included a small demonstration of our current implementation in the MATHEMAGIX system [vdH02b].

For the purpose of this application, we have mainly considered univariate transseries expansions so far. Of course, the approach of our paper generalizes to expansions in several variables. A natural next step for future developments would be to implement the Newton polygon method for rather general functional equations. Another interesting question is how to re-incorporate theoretically correct zero-tests in our mechanism and how much we really sacrifice when using our heuristic substitute. A few ideas in these directions will be given in section 8, some of which actually go back to [vdH94].

2. Transseries

In this section, we briefly survey some basic properties of transseries. For more details, we refer to [vdH06c, vdH97, É92, MMvdD97, MMvdD99].

Let R be a ring and $\mathfrak{M}$ a commutative monomial monoid which is partially ordered by an *asymptotic dominance relation* $\preccurlyeq$. A subset $\mathfrak{S} \subseteq \mathfrak{M}$ is said to be *well-based* if it well-quasi-ordered [Pou85, Mil85] for the opposite ordering of $\preccurlyeq$ and *grid-based* if it satisfies a bound of the form

$$\mathfrak{G} \subseteq \mathfrak{m}_1^{\mathbb{N}} \cdots \mathfrak{m}_k^{\mathbb{N}} \{\mathfrak{n}_1, \ldots, \mathfrak{n}_l\} \qquad (\mathfrak{m}_1, \ldots, \mathfrak{m}_k \prec 1). \tag{1}$$

A *well-based power series* is a formal sum $f = \sum_{\mathfrak{m} \in \mathfrak{M}} f_{\mathfrak{m}} \mathfrak{m}$, whose support $\operatorname{supp} f = \{\mathfrak{m} \in \mathfrak{M}: f_{\mathfrak{m}} \neq 0\}$ is well-based. It is classical [Hah07, Hig52] that the set $R[[\mathfrak{M}]]$ of well-based power series forms a ring. The subset $R[\![\mathfrak{M}]\!]$ of grid-based power series (i.e. with grid-based support) forms a subring of $R[[\mathfrak{M}]]$.

Example 1. Consider the series

$$\begin{aligned} f &= \frac{1}{1 - x^{-1} - x^{-\mathrm{e}}} = 1 + x^{-1} + x^{-2} + x^{-\mathrm{e}} + x^{-3} + x^{-\mathrm{e}-1} + x^{-4} + x^{-\mathrm{e}-2} + x^{-5} + x^{-2\mathrm{e}} + \cdots \\ g &= x^{-1} + g(x^{\pi}) = x^{-1} + x^{-\pi} + x^{-\pi^2} + x^{-\pi^3} + \cdots, \end{aligned}$$

with $\mathfrak{M} = x^{\mathbb{R}}$ for $x \to \infty$ (i.e. $x^{\alpha} \preccurlyeq x^{\beta} \Leftrightarrow \alpha \leqslant \beta$). Then the first series is grid-based and the second one only well-based.

A family $(f_I)_{i \in I}$ of series in $R[[\mathfrak{M}]]$ is said to be *well-based* if $\bigcup_{i \in I} \operatorname{supp} f_i$ is well-based and $\{i \in I: \mathfrak{m} \in \operatorname{supp} f_i\}$ is finite for every $\mathfrak{m} \in \mathfrak{M}$. In that case, the sum $g = \sum_{i \in I} f_i$ with $g_{\mathfrak{m}} = \sum_{i \in I} f_{i,\mathfrak{m}}$ is again in $R[[\mathfrak{M}]]$. A linear mapping $\varphi: R[[\mathfrak{M}]] \to R[[\mathfrak{N}]]$ is said to be *strong* it preserves well-based summation. Grid-based families and the corresponding notion of strong linearity are defined similarly.

In the case when R is a field and $\mathfrak{M}$ is totally ordered, then $R[\![\mathfrak{M}]\!]$ and $R[[\mathfrak{M}]]$ are also fields. Furthermore, any non-zero $f \in \mathbb{R}[[\mathfrak{M}]]^{\neq}$ admits a unique *dominant monomial* $\mathfrak{d}_f = \max_{\preccurlyeq} \operatorname{supp} f$ with corresponding *dominant coefficient* $c_f = f_{\mathfrak{d}_f}$ and *relative remainder* δ_f such that $f = c_f \mathfrak{d}_f (1 + \delta_f)$. We call $\tau_f = c_f \mathfrak{d}_f$ the *dominant term* of f and define $\tau_f = 0$ in the case when $f = 0$. The series f also admits a *canonical decomposition*

$$f = \underbrace{f_{\succ}}_{\sum_{\mathfrak{m} \succ 1} f_{\mathfrak{m}} \mathfrak{m}} + \underbrace{f_{\asymp}}_{\sum_{\mathfrak{m} \asymp 1} f_{\mathfrak{m}} \mathfrak{m}} + \underbrace{f_{\prec}}_{\sum_{\mathfrak{m} \prec 1} f_{\mathfrak{m}} \mathfrak{m}}$$

Here $\mathfrak{m} \asymp 1$ just means that $\mathfrak{m} = 1$; more generally, $\varphi \asymp \psi \Leftrightarrow \varphi \preccurlyeq \psi \preccurlyeq \varphi$. If f is grid-based, then so are δ_f, $f_{\succ}$ and $f_{\prec}$. If R is actually an ordered field, then so are $R[\![\mathfrak{M}]\!]$ and $R[[\mathfrak{M}]]$, by taking $f > 0 \Leftrightarrow c_f > 0$ for all f.

The field $\mathbb{T} = \mathbb{R}[\![x]\!]$ of grid-based transseries is a field of the form $\mathbb{T} = \mathbb{R}[\![\mathfrak{T}]\!]$ with additional operators exp and log. The set of *transmonomials* $\mathfrak{T}$ coincides with the set $\exp \mathbb{T}_{\succ}$ of exponentials of transseries f with $f_{\succ} = f$. More generally, we have

$$\log f = \log \mathfrak{d}_f + \log c_f + \log (1 + \delta_f) \tag{2}$$

$$\log \delta_f = \delta_f - \tfrac{1}{2} \delta_f^2 + \tfrac{1}{3} \delta_f^3 + \cdots, \tag{3}$$

for any $f \in \mathbb{T}^{>}$ (i.e. $f > 0$) and

$$\exp f = \exp f_{\succ} \exp f_{\asymp} \exp f_{\prec} \quad (4)$$
$$\exp f_{\prec} = 1 + f_{\prec} + \tfrac{1}{2} f_{\prec}^2 + \tfrac{1}{6} f_{\prec}^3 + \cdots. \quad (5)$$

The construction of $\mathbb{T}$ is detailed in [vdH06c, Chapter 4]. The construction of fields of well-based transseries is a bit more delicate [DG86, vdH97, Sch01], because one cannot simultaneously ensure stability under exponentiation and infinite summation. However, there is a smallest such field $\mathbb{R}[[[x]]]$, if we exclude transseries with arbitrarily nested logarithms or exponentials, such as $x + \log x + \log\log x + \cdots$.

Let $\mathbb{T}$ be one of the fields $\mathbb{R}\mathbb{\Pi}x\mathbb{\Pi}$ or $\mathbb{R}[[[x]]]$. Then $\mathbb{T}$ admits a lot of additional structure:

1. There exists a unique strong derivation $f \mapsto f'$ with $\mathbb{R}' = 0$, $x' = 1$ and $(\mathrm{e}^f)' = f'\mathrm{e}^f$ for all $f \in \mathbb{T}$.
2. There exists a unique strong integration $f \mapsto \int f$ with $(\int f)' = f$ and $(\int f)_{\asymp} = 0$ for all $f \in \mathbb{T}$.
3. For any positive, infinitely large transseries $g \in \mathbb{T}^{>,\succ}$, there exists a unique strongly linear right composition $f \mapsto f \circ g$, with $c \circ g$ ($c \in \mathbb{R}$), $x \circ g = g$ and $\mathrm{e}^f \circ g = \mathrm{e}^{f \circ g}$.
4. Each $g \in \mathbb{T}^{>,\succ}$ admits a unique functional inverse g^{inv}.
5. $\mathbb{T}$ is real closed. Even better: given a differential polynomial $P \in \mathbb{T}\{F\}$ and $f < g$ in $\mathbb{T}$ with $P(f)\,P(g) < 0$, there exists an $h \in \mathbb{T}$ with $f < h < g$ and $P(h) = 0$.

Furthermore, there exist very general implicit function theorems [vdH01, vdH06c], which can be used in order to solve functional equations, when put in a suitable normal form.

The field $\mathbb{T}$ is highly non-Archimedean. For what follows it will be introduce the asymptotic flatness relation

$$f \prec\!\!\!\prec g \iff \log f \prec \log g$$
$$f \asymp g \iff \log f \asymp \log g$$

For $g \succ 1$, one has $f \prec\!\!\!\prec g$ if and only if $f^\lambda \prec g$ for all $\lambda \in \mathbb{R}$.

3. META-EXPANSIONS

The intuitive idea behind meta-expansion is that, from the computational point of view, series usually arise as a sequence of successive approximations of an interesting object. We will make this simple idea operational by taking the approximations to be polynomials. In order to avoid repetitions, we will systematically work in the well-based setting; it is easy to adapt the definitions to the grid-based case.

Let R be an effective ring and $\mathfrak{M}$ an effective monomial monoid. Recall that $R[[\mathfrak{M}]]$ stands for the ring of well-based generalized power series and let $R[\mathfrak{M}]$ be the corresponding set of polynomials (i.e. series with finite support). We define an *expander* to be a computable well-based sequence $\check{f} = (\check{f}_n) \in R[\mathfrak{M}]^{\mathbb{N}}$ of polynomials. Its sum $f = \hat{\check{f}} = \sum_{n \in \mathbb{N}} \check{f}_n$ will be called the *result* of the expander and we say that $\check{f}$ is an expander *for* f. We also define an *approximator* to be a computable sequence $\check{f}_; = (\check{f}_{;n}) \in R[\mathfrak{M}]^{\mathbb{N}}$, such that $\bigcup_{n \in \mathbb{N}} \operatorname{supp} f_{;n}$ is well-based and such that for each $\mathfrak{m} \in \mathfrak{M}$, there exists an $n_0 \in \mathbb{N}$ for which $f_{;\mathfrak{m}} := \check{f}_{;n,\mathfrak{m}}$ is constant for $n \geqslant n_0$. In that case, the limit $f_; = \lim \check{f}_{;n} = \sum_{\mathfrak{m} \in \mathfrak{M}} f_{;\mathfrak{m}}\,\mathfrak{m}$ is called the *result* of the approximator.

Clearly, the notions of expander and approximator are variants of another: if $\check{f} = (\check{f}_n)$ is an expander, then $\check{f}_; = \Sigma\, \check{f}$ with $\check{f}_{;n} = \check{f}_0 + \cdots + \check{f}_n$ defines an approximator with the same result. Similarly, if $\check{f}_; = (\check{f}_{;n})$ is an approximator, then $\check{f} = \Delta\, \check{f}_;$ with $\check{f}_0 = \check{f}_{;0}$, $\check{f}_n = \check{f}_{;n} - \check{f}_{;n-1}$ $(n > 0)$ defines an expander with the same result. However, for certain purposes, expanders are more suitable, because an expander $\check{f}$ can be manipulated *via* its generating series

$$\check{f}(z) = \sum_{n \in \mathbb{N}} \check{f}_n\, z^n.$$

For other purposes though, approximators are the more natural choice. As far as notations are concerned, it is convenient to pass from expanders $\check{f}$ to approximators $\check{f}_;$ (and *vice versa*) by prefixing the index by a semicolon (resp. removing the semicolon).

We will denote by $R[[\mathfrak{M}]]^{\text{app}}$ the set of *approximable series* in $R[[\mathfrak{M}]]$ which admit an expander (or approximator). The corresponding set of expanders will be denoted by $\widecheck{R[[\mathfrak{M}]]^{\text{app}}}$. Given $f \in R[[\mathfrak{M}]]^{\text{exp}}$, we use the notation $\hat{f}$ to indicate that f represents $\hat{f}$. Given $f \in R[[\mathfrak{M}]]^{\text{app}}$, we will also use the notation $\check{f}$ to indicate that $\check{f} \in \widecheck{R[[\mathfrak{M}]]^{\text{app}}}$ is a representation for f. For more details on this convention, see [vdH07, Section 2.1].

In practice, expanders and approximators are usually implemented by pointers to an abstract class with a method to compute its coefficients. For more details on how to do this, we refer to [vdH02a]. Let us now show how to implement expanders and approximators for basic operations in $R[[\mathfrak{M}]]^{\text{app}}$.

Constructor. Given a polynomial $f \in R[\mathfrak{M}]$, an expander and approximator for f are given by

$$\begin{aligned} \check{f}(z) &= f \\ \check{f}_{;n} &= f \end{aligned}$$

It will be convenient to simply regard $R[\mathfrak{M}]$ as a subset of $\widecheck{R[[\mathfrak{M}]]^{\text{app}}}$.

Addition. Given $g, h \in R[[\mathfrak{M}]]^{\text{app}}$, we may compute an expander and an approximator for $g + h$ by

$$\begin{aligned} \check{f}(z) &= \check{g}(z) + \check{h}(z) \\ \check{f}_{;n} &= \check{g}_{;n} + \check{h}_{;n} \end{aligned}$$

Subtraction is treated in a similar way.

Multiplication. We may define expanders and approximators for products in the same way as for addition:

$$\check{f}(z) = \check{g}(z)\, \check{h}(z) \tag{6}$$
$$\check{f}_{;n} = \check{g}_{;n}\, \check{h}_{;n} \tag{7}$$

However, a subtlety occurs here. Consider for instance the case when $R[[\mathfrak{M}]] = R[[u]]$ and $\check{g} = \check{h} = 1/(1 - u\, z)$, so that $g = h = 1/(1-u)$ and $\check{g}_{;n} = \check{h}_{;n} = 1 + \cdots + u^n$. Contrary to what happened in the case of addition, the definitions (6) and (7) do not coincide in the sense that $\check{f} \neq \Sigma\, \check{f}_;$. Indeed, we respectively find

$$\begin{aligned} \check{f}_{;n} &= 1 + 2\, u + \cdots + n\, u^n \\ \check{f}_{;n} &= 1 + 2\, u + \cdots + n\, u^n + \cdots + 2\, u^{2n-2} + u^{2n-1} \end{aligned}$$

As a general rule, the manipulation of expanders tends to be a bit more economic from the computational point of view, its coefficients being smaller in size.

Left composition with power series. Let $g \in R[[t]]$ be a computable formal power series and let $h \in R[[\mathfrak{M}]]^{\mathrm{app}}_{\prec}$ be infinitesimal. Then we may compute an expander for $g \circ h$ using

$$\check{f}(z) \;=\; g(z\,\check{h}(z)). \tag{8}$$

Notice that the individual coefficients of $\check{h}$ need not be infinitesimal. Besides, the composition $g \circ f_0$ is usually not a polynomial. Therefore, we have forced $\check{h}$ to become infinitesimal using a multiplication by z. This is justified by the fact that

$$h = z\,\widehat{\check{h}}\,,$$

Multiplication of $\check{h}$ by z corresponds to delaying the approximation process of h.

Inversion. Assume now that R is a field and $\mathfrak{M}$ a totally ordered group. The inverse of a series $g \in R[[\mathfrak{M}]]$ may be computed using left composition with power series:

$$g = \frac{1}{\tau_g}\,(1 - \delta_g + \delta_g^2 - \delta_g^3 + \cdots) = \frac{1}{\tau_g}\,(1+t)^{-1} \circ \delta_g \tag{9}$$

Unfortunately, there exists no general algorithm for the computation of the dominant term τ_g. We will therefore assume the existence of an oracle for this purpose. In section 6, we will present a heuristic algorithm which can be used in practice.

Fixed points. A general technique for the resolution of functional equations is to rewrite then into the form

$$f = \Phi(f)$$

and apply a fixed-point theorem. In our case, this requires a well-based operator Φ: $R[[\mathfrak{M}]] \to R[[\mathfrak{M}]]$ for which we can prove that $\Phi(0)$, $\Phi(\Phi(0))$, $\Phi(\Phi(\Phi(0)))$, ... admits a well-based limit in $R[[\mathfrak{M}]]$. The obvious expander for f is given by

$$\check{f}_n = \Phi^n(0).$$

For some general fixed-point theorems for sequences $\check{f}_n$ of this kind, we refer to [vdH01] and [vdH06c, Chapter 6].

4. META-EXPANSION OF TRANSSERIES

In order to compute with well-based transseries in $\mathbb{R}[[[x]]] \subseteq \mathbb{R}[[\mathfrak{T}]]$, we take our coefficients in an effective subfield R of $\mathbb{R}$ and our monomials in an effective subgroup $\mathfrak{M}$ of $\mathfrak{T}$. Moreover, the monomials in $\mathfrak{M}$ which are not iterated logarithms are themselves exponentials of approximable transseries.

More precisely, elements in $\mathfrak{M}$ are represented by monomials $\check{\mathfrak{m}} \in \breve{\mathfrak{M}}$ which are of one of the following forms:

1. either $\check{\mathfrak{m}} = \mathfrak{m} = \log_l x$ for some $l \in \mathbb{N}$;
2. or $\check{\mathfrak{m}} = \exp \check{f}$, with $\check{f} \in \widetilde{R[[\mathfrak{M}]]_{\succ}}$.

In the first case, the exponential height $h_{\check{\mathfrak{m}}}$ of $\check{\mathfrak{m}} \in \breve{\mathfrak{M}}$ is defined to be zero and in the second case, we set $h_{\check{\mathfrak{m}}} = 1 + \max_{\check{\mathfrak{n}} \in \operatorname{supp} \log \check{\mathfrak{m}}} h_{\check{\mathfrak{n}}}$.

Elements $\mathfrak{m}, \mathfrak{n} \in \mathfrak{M}$ are multiplied using $\check{\mathfrak{m}}\,\check{\mathfrak{n}} = \exp(\log \check{\mathfrak{m}} + \log \check{\mathfrak{n}})$ and inverted using $\check{\mathfrak{m}}^{-1} = \exp(-\log \check{\mathfrak{m}})$. Here $\log \check{\mathfrak{m}} \in \widetilde{R[[\mathfrak{M}]]_{\succ}}$: if $\check{\mathfrak{m}} = \log_l x$, then $\log \check{\mathfrak{m}} = \log_{l+1} x$; if $\check{\mathfrak{m}} = \exp \check{f}$, then $\log \check{\mathfrak{m}} = \check{f}$. The asymptotic ordering $\preccurlyeq$ on $\mathfrak{M}$ is implemented using

$$\mathfrak{v} \preccurlyeq \mathfrak{w} \Longleftrightarrow \log \mathfrak{v} \leqslant \log \mathfrak{w} \Longleftrightarrow \tau_{\log(\mathfrak{v}/\mathfrak{w})} \leqslant 0,$$

and therefore relies on an oracle for the computation of $\tau_{\log(\mathfrak{v}/\mathfrak{w})}$. Setting $\mathfrak{m} = \mathfrak{v}/\mathfrak{w}$, we notice that heuristic algorithms for the computation of this dominant term recursively need to compare elements $\mathfrak{n}$ in $\operatorname{supp}\log\mathfrak{m}$ with respect to $\preccurlyeq$. The termination of the recursion is based on the fact that $h_{\check{\mathfrak{n}}} < h_{\check{\mathfrak{m}}}$.

The main additional operations for the manipulation of transseries are exponentiation and logarithm. Since exponentiation relies on canonical decompositions, we start with the general operation of restriction of support.

Restriction of support. Given $\mathfrak{S} \subseteq \mathfrak{M}$, we define the restriction of $f \in R[[\mathfrak{M}]]$ to $\mathfrak{S}$ by

$$f_{\mathfrak{S}} = \sum_{\mathfrak{m} \in \mathfrak{S}} f_{\mathfrak{m}}\, \mathfrak{m}$$

If the subset $\mathfrak{S}$ is a computable, i.e. $\mathfrak{S}$ admits a computable membership test, then the mapping $R[\mathfrak{M}] \to R[\mathfrak{M}]; f \mapsto f_{\mathfrak{S}}$ is computable. In particular, given $f \in R[[\mathfrak{M}]]^{\mathrm{app}}$, we may compute $f_{\mathfrak{S}}$ using

$$(\widetilde{f_{\mathfrak{S}}})_n = (\check{f}_n)_{\mathfrak{S}}.$$

Now making continued use of our oracle for the computation of dominant terms, the sets $\mathfrak{M}_{\succ} = \{\mathfrak{m} \in \mathfrak{M} : \mathfrak{m} \succ 1\}$, $\mathfrak{M}_{\asymp} = \{1\}$ and $\mathfrak{M}_{\prec} = \{\mathfrak{m} \in \mathfrak{M} : \mathfrak{m} \prec 1\}$ are computable. Consequently, we have algorithms to compute $f_{\succ} = f_{\mathfrak{M}_{\succ}}$, $f_{\asymp} = f_{\mathfrak{M}_{\asymp}}$ and $f_{\prec} = f_{\mathfrak{M}_{\prec}}$.

Logarithm and exponentiation. Assume that R is closed under logarithm (for positive elements) and exponentiation. Then the formulas (2–5) and our algorithms for canonical decomposition and left composition with power series yield a way to compute logarithms and exponentials of elements in $R[[\mathfrak{M}]]^{\mathrm{app}}$. The smallest subfield R of $\mathbb{R}$ which is stable under exponentiation and logarithm is called the field of *exp-log constants*. There exists a zero-test for this field which relies on Schanuel's conjecture for its termination [Ric97].

EXAMPLE 2. Consider the following example from [RSSvdH96]:

$$f = \log\log\left(x\,\mathrm{e}^{x\mathrm{e}^{x}} + 1\right) - \exp\exp\left(\log\log x + \frac{1}{x}\right).$$

When computing an expander $\check{f}$ with the routines presented so far, we obtain

$$\begin{aligned}
\check{f}_{;0} &= \log x \\
\check{f}_{;1} &= \log x + \frac{\log x}{x\,\mathrm{e}^{x}} + \frac{1}{x\,\mathrm{e}^{x}} \\
\check{f}_{;2} &= \frac{\log x}{x\,\mathrm{e}^{x}} + \frac{1}{x\,\mathrm{e}^{x}} - \frac{\log^2 x}{2\,x^2\,\mathrm{e}^{2x}} - \frac{\log x}{x^2\,\mathrm{e}^{2x}} - \frac{1}{2\,x^2\,\mathrm{e}^{2x}} \\
\check{f}_{;3} &= -\frac{\log x}{2\,x} + \frac{\log x}{x\,\mathrm{e}^{x}} + \frac{1}{x\,\mathrm{e}^{x}} - \frac{\log^2 x}{2\,x^2\,\mathrm{e}^{2x}} - \frac{\log x}{x^2\,\mathrm{e}^{2x}} - \frac{1}{2\,x^2\,\mathrm{e}^{2x}} + \frac{\log^3 x}{3\,x^3\,\mathrm{e}^{3x}} + \frac{\log^2 x}{x^3\,\mathrm{e}^{3x}} + \frac{\log x}{x^3\,\mathrm{e}^{3x}} + \\
&\quad \frac{1}{3\,x^3\,\mathrm{e}^{3x}} \\
\check{f}_{;4} &= -\frac{\log^2 x}{2\,x} - \frac{\log x}{2\,x} - \frac{\log x}{6\,x^2} + \frac{\log x}{x\,\mathrm{e}^{x}} + \frac{1}{x\,\mathrm{e}^{x}} - \frac{\log^2 x}{2\,x^2\,\mathrm{e}^{2x}} - \frac{\log x}{x^2\,\mathrm{e}^{2x}} - \frac{1}{2\,x^2\,\mathrm{e}^{2x}} + \frac{\log^3 x}{3\,x^3\,\mathrm{e}^{3x}} + \\
&\quad \frac{\log^2 x}{x^3\,\mathrm{e}^{3x}} + \frac{\log x}{x^3\,\mathrm{e}^{3x}} + \frac{1}{3\,x^3\,\mathrm{e}^{3x}} - \frac{\log^4 x}{4\,x^4\,\mathrm{e}^{4x}} - \frac{\log^3 x}{x^4\,\mathrm{e}^{4x}} - \frac{3\log^2 x}{2\,x^4\,\mathrm{e}^{4x}} - \frac{\log x}{x^4\,\mathrm{e}^{4x}} - \frac{1}{4\,x^4\,\mathrm{e}^{4x}} \\
\check{f}_{;5} &= -\frac{\log^2 x}{2\,x} - \frac{\log x}{2\,x} - \frac{\log^2 x}{2\,x^2} - \frac{\log x}{6\,x^2} - \frac{\log x}{24\,x^3} + \frac{\log x}{x\,\mathrm{e}^{x}} + \frac{1}{x\,\mathrm{e}^{x}} - \frac{\log^2 x}{2\,x^2\,\mathrm{e}^{2x}} - \frac{\log x}{x^2\,\mathrm{e}^{2x}} - \\
&\quad \frac{1}{2\,x^2\,\mathrm{e}^{2x}} + \frac{\log^3 x}{3\,x^3\,\mathrm{e}^{3x}} + \frac{\log^2 x}{x^3\,\mathrm{e}^{3x}} + \frac{\log x}{x^3\,\mathrm{e}^{3x}} + \frac{1}{3\,x^3\,\mathrm{e}^{3x}} - \frac{\log^4 x}{4\,x^4\,\mathrm{e}^{4x}} - \frac{\log^3 x}{x^4\,\mathrm{e}^{4x}} - \frac{3\log^2 x}{2\,x^4\,\mathrm{e}^{4x}} - \\
&\quad \frac{\log x}{x^4\,\mathrm{e}^{4x}} - \frac{1}{4\,x^4\,\mathrm{e}^{4x}} + \frac{\log^5 x}{5\,x^5\,\mathrm{e}^{5x}} + \frac{\log^4 x}{x^5\,\mathrm{e}^{5x}} + \frac{2\log^3 x}{x^5\,\mathrm{e}^{5x}} + \frac{2\log^2 x}{x^5\,\mathrm{e}^{5x}} + \frac{\log x}{x^5\,\mathrm{e}^{5x}} + \frac{1}{5\,x^5\,\mathrm{e}^{5x}}
\end{aligned}$$

REMARK 3. In practice, it is useful to have efficient algorithms for the manipulation of transmonomials. In a similar way as in [RSSvdH96, vdH97, vdH06c], we therefore write transmonomials as power products $\mathfrak{m} = \mathfrak{b}_1^{\lambda_1} \cdots \mathfrak{b}_n^{\lambda_n}$ with respect to a *transbasis* $(\mathfrak{b}_1, ..., \mathfrak{b}_n)$ which is constructed incrementally during the computations. In our well-based setting, we merely require that the $\mathfrak{b}_i$ satisfy the hypotheses

TB1. $\mathfrak{b}_1 = \log_l x$ for some $l \in \mathbb{N}$.

TB2. $\mathfrak{b}_2, ..., \mathfrak{b}_n \in \exp R[[\mathfrak{M}]]_{\succ}$.

TB3. $\log \mathfrak{b}_1 \prec \cdots \prec \log \mathfrak{b}_n$.

In the grid-based setting **TB2** may be replaced by the stronger requirement that $\log \mathfrak{b}_{i+1}$ can be expanded w.r.t. $\mathfrak{b}_1, ..., \mathfrak{b}_i$ for all $i \in \{1, ..., n-1\}$.

5. OTHER OPERATIONS ON TRANSSERIES

Let us now come to the more interesting operations on transseries. Differentiation and composition rely on the general principle of extension by strong linearity.

Extension by strong linearity. Let $\varphi: \mathfrak{M} \to R[[\mathfrak{M}]]$ be a map such that each well-based subset $\mathfrak{S}$ of $\mathfrak{M}$ is mapped into a well-based family $(\varphi(\mathfrak{m}))_{\mathfrak{m} \in \mathfrak{S}}$. Then there exists a unique strongly linear extension $\Phi: R[[\mathfrak{M}]] \to R[[\mathfrak{M}]]$ of φ. If $\varphi: \mathfrak{M} \to R[[\mathfrak{M}]]^{\mathrm{com}}$ is computable, then we may compute the restriction of Φ to $R[[\mathfrak{M}]]^{\mathrm{app}}$ by

$$\Phi(\check{f})_n = \sum_{p+q=n} \sum_{\mathfrak{m} \in \operatorname{supp} \check{f}_p} \check{f}_{p,\mathfrak{m}} \widetilde{\varphi(\mathfrak{m})}_q.$$

Differentiation. The derivative of an approximable transseries in $R[[\mathfrak{M}]]^{\mathrm{app}}$ is computed using extension by strong linearity. The derivative of a transmonomial is computed recursively: $(\log_l x)' = 1/(x \cdots \log_{l-1} x)$ and $(\exp f)' = f' (\exp f)$.

Composition. Right composition with a fixed $g \in R[[\mathfrak{M}]]^{\mathrm{app},\succ,>}$ is done similarly. For arbitrary transseries in $R[[\mathfrak{M}]]^{\mathrm{app}}$, we use extension by strong linearity. Transmonomials are handled recursively: $(\log_l x) \circ g = \log_l g$ and $(\exp f) \circ g = \exp(f \circ g)$.

It can be shown [vdH06c, vdH97] that the derivation w.r.t. x admits a unique strongly linear right inverse $\int$ with the "distinguished property" that $(\int f)_{\asymp} = 0$ for all f. One way to construct $\int$ is to first compute its "trace" $T = T_{\int}: R[[\mathfrak{M}]] \to R[[\mathfrak{M}]]$ which is the unique strongly linear operator with $Tf = \tau_{\int f}$ on $R\mathfrak{M}$. We then compute $\int f$ by solving the implicit equation

$$\int f = Tf + \int (f - (Tf)'). \tag{10}$$

One may either apply (10) for monomials and extend by strong linearity, or apply it directly for arbitrary transseries f.

Trace of the distinguished integration. The trace $T\mathfrak{m} = T_{\int}\ \mathfrak{m}$ of a transmonomial is computed using the formula

$$T\mathfrak{m} = \begin{cases} \dfrac{\mathfrak{m}^2}{\tau_{\mathfrak{m}'}} & \text{if } \log \mathfrak{m} \succcurlyeq x \\ [T((x\,\mathfrak{m}) \circ \exp)] \circ \log & \text{otherwise} \end{cases}$$

We next extend by strong linearity.

Distinguished integration. We may rewrite (10) in operator form

$$\int = T(1 + (1 - \partial T) + (1 - \partial T)^2 + \cdots)$$

and define an expander for this operator:

$$\check{\int}(z) = \sum_{n=0}^{\infty} T\,(1-\partial T)^n\, z^n.$$

Then distinguished integration can be regarded as the application of this operator expander to another expander $\check{f}$:

$$(\widetilde{\int f})(z) = \check{\int}(z)\,\check{f}(z) = \sum_{n=0}^{\infty}\sum_{k=0}^{\infty} \check{\int}_n\,\check{f}_k\, z^{n+k}.$$

We also notice that $\int$ is a fixed-point of the operator

$$\int \longmapsto T + \int\,(1-\partial T).$$

Adapting our general mechanism for the computation of fixed points for operators instead of series, we find $z\,\check{\int}(z)$ as the natural expander of $\int$.

Functional inversion of transseries can be done using formulas in [vdH06c, Section 5.4] and we will not detail this operation here. Two other interesting operations are *finite differences* and *discrete summation*:

$$\begin{aligned}\Delta f &= f\circ(x+1) - f\\ \Sigma f &= \Delta^{-1} f.\end{aligned}$$

We implemented these operations because they are critically needed in an algorithm for the asymptotic extrapolation of sequences [vdH06a]. Modulo compositions with exp and log, they are related to *finite quotients* and *discrete products*:

$$\begin{aligned}\mathrm{Q}\, f &= \frac{f\circ(x+1)}{f} = \exp\Delta\log f\\ \Pi f &= \exp\Sigma\log f.\end{aligned}$$

Our algorithm for right-composition clearly yields a way to compute Δf for $f\in R[[\mathfrak{M}]]^{\mathrm{app}}$. The distinguished summation Σ is the unique distinguished strongly linear right-inverse of Δ, i.e. $\Delta\Sigma f = f$ and $(\Sigma f)_{\asymp} = 0$ for all f. It therefore suffices to show how to compute $\Sigma\mathfrak{m}$ for monomials $\mathfrak{m}\in\mathfrak{M}$. Three different cases need to be distinguished :

Flat discrete summation. Assuming $\mathfrak{m}\prec\!\!\!\prec \mathrm{e}^x$ (i.e. $\log\mathfrak{m}\prec x$), we compute $\Sigma\mathfrak{m}$ by solving the equation

$$\Delta f = (\mathrm{e}^{\partial}-1)\,f = \left(\partial + \frac{1}{2}\,\partial^2 + \frac{1}{6}\,\partial^3 + \cdots\right) f = \mathfrak{m},$$

which yields a solution

$$f = \int \mathfrak{m} + \frac{1+\partial-\mathrm{e}^{\partial}}{\partial\,(\mathrm{e}^{\partial}-1)}\,\mathfrak{m} = \int \mathfrak{m} + \left(-\frac{1}{2} + \frac{1}{12}\,\partial - \frac{1}{720}\,\partial^3 + \frac{1}{30240}\,\partial^5 + \cdots\right)\mathfrak{m}.$$

The application of the operator

$$\Phi(\partial) = \frac{1+\partial-\mathrm{e}^{\partial}}{\partial\,(\mathrm{e}^{\partial}-1)}$$

to $\mathfrak{m}$ is computed in a similar way as in the case of distinguished integration. In fact, the expander $\check{\Phi}(z) = \Phi(z\,\partial)$ can directly be applied to expanders $\check{f}$ with $\mathfrak{m}\prec\!\!\!\prec \mathrm{e}^x$ for all $\mathfrak{m}\in\operatorname{supp} f$. Moreover, this application preserves grid-basedness.

Moderate discrete summation. In the case when $\mathfrak{m}\succ\!\!\!\succ \mathrm{e}^x$ (i.e. $\log\mathfrak{m}\succ\!\!\!\succ x$), let $c\in R$ be such that $\mathfrak{m} = \mathfrak{n}\,\mathrm{e}^{cx}$ with $\mathfrak{n}\prec\!\!\!\prec \mathrm{e}^x$. We now search for a solution to $\Delta f = \mathfrak{m}$ of the form $f = g\,\mathrm{e}^{cx}$, which leads to the equation

$$\mathrm{e}^c\, g\circ(x+1) - g = \mathfrak{n}, \tag{11}$$

We rewrite this equation in operator form

$$(\mathrm{e}^c\,\mathrm{e}^{\partial}-1)(g)=\left(\mathrm{e}^c-1+\mathrm{e}^c\,\partial+\frac{\mathrm{e}^c}{2}\,\partial^2+\cdots\right)=\mathfrak{n}$$

and we invert the operator $\mathrm{e}^c\,\mathrm{e}^{\partial}-1$ as in the flat case. No integration is needed this time, since $\mathrm{e}^c-1\neq 0$. Again, the grid-based property is preserved by moderate discrete summation.

Steep discrete summation. In the case when $\mathfrak{m}\gg\mathrm{e}^x$ (i.e. $\log\mathfrak{m}\succ x$), we have to solve the equation

$$f\circ(x+1)-f=\mathfrak{m}.$$

If $\mathfrak{m}\prec 1$, then this is done by computing a fixed point for the operator

$$f\longmapsto-\mathfrak{m}+f\circ(x+1).$$

If $\mathfrak{m}\succ 1$, then we compute a fixed point for the operator

$$f\longmapsto\mathfrak{m}\circ(x-1)+f\circ(x-1).$$

It can be shown that Σf is grid-based if f is grid-based and there exists a k such that $\mathfrak{m}\ll\mathrm{e}^{x^k}$ for all $\mathfrak{m}\in\operatorname{supp} f$.

6. META-OPERATIONS ON EXPANDERS

So far, we have not really exploited the extra level of abstraction provided by expanders. In this section, we will describe several "meta-operations" on expanders. These operations do not affect the series being represented, but rather concern qualitative aspects of the approximation process: they guide the rate of convergence, the terms which appear first, etc. Based on the process of "stabilization", we will also describe a heuristic zero-test and a heuristic method for the computation of dominant terms.

Shortening. In the algorithm for left composition with formal power series, we have already observed that the expanders $\check{f}(z)$ and $z\,\check{f}(z)$ represent the same series. More generally, given a computable function $\phi\colon\mathbb{N}\to\mathbb{N}$ with $n-\phi(n)\to\infty$, we define the *shortening* operator sh_ϕ by

$$(\mathrm{sh}_\phi\,\check{f})_{;n}=\begin{cases}0 & \text{if } n<\phi(n)\\ \check{f}_{;n-\phi(n)} & \text{otherwise}\end{cases}$$

In the case when $\phi(n)=k\in\mathbb{N}$ is a constant function, we have

$$(\mathrm{sh}_k\,\check{f})(z)=z^k\,\check{f}(z).$$

The shortening operator is typically used for the expansion of expressions which involve an expander $\check{f}$, such that the expression size of $\check{f}_{;n}$ tends to grow very rapidly with n. For instance, we may prefer to compute a sum $f+g$ using the expander $\mathrm{sh}_\phi\,\check{f}+\check{g}$ instead of $\check{f}+\check{g}$.

Lengthening. Given a computable function $\phi\colon\mathbb{N}\to\mathbb{N}$, the *lengthening* operator le_ϕ is defined by

$$(\mathrm{le}_\phi\,\check{f})_{;n}=\check{f}_{;n+\phi(n)}.$$

In the case when $\phi(n)=k\in\mathbb{N}$ is a constant function, we have

$$(\mathrm{le}_k\,\check{f})(z)=\frac{\check{f}(z)-\check{f}_{k-1}\,z^{k-1}-\cdots-\check{f}_0}{z^k}.$$

During the expansion of an expression, the lengthening operator may for instance be used in order to boost the precision of a subexpression. We may also use it as a substitute for the order parameter of a typical expansion command. E.g., we would simply display $\mathrm{le}_{100}\, f$ in order to show an additional 100 terms of f.

Stabilization. An even more interesting meta-operation is *stabilization*. Given a computable function $\phi: \mathbb{N} \to \mathbb{N}$, we define it by

$$\begin{aligned}(\mathrm{stab}_\phi\, f)_{;n} &= \check{f}_{\mathfrak{S}_{\check{f},\phi,n}} = \sum_{\mathfrak{m}\in\mathfrak{S}_{\check{f},\phi,n}} \check{f}_{;n,\mathfrak{m}}\,\mathfrak{m}\\ \mathfrak{S}_{\check{f},\phi,n} &= \{\mathfrak{m}\in\operatorname{supp}\check{f}_{;n}:\ \check{f}_{;n,\mathfrak{m}} = \check{f}_{;n+1,\mathfrak{m}} = \cdots = \check{f}_{;n+\phi(n),\mathfrak{m}}\}.\end{aligned}$$

The stabilization operator removes all terms from the expansion $\check{f}_{;n}$ which are still subject to changes during the next $\phi(n)$ approximations. Even for small values of $k\in\mathbb{N}$, such as $k=3$, we usually have

$$(\mathrm{stab}_k\,\check{f})_{;0} \sqsubseteq (\mathrm{stab}_k\,\check{f})_{;1} \sqsubseteq (\mathrm{stab}_k\,\check{f})_{;2} \sqsubseteq \cdots, \tag{12}$$

where

$$\varphi\sqsubseteq\psi \iff (\psi-\varphi)_{\operatorname{supp}\varphi} = 0.$$

In particular, the successive approximations $(\mathrm{stab}_k\,\check{f})_{;n}$ usually only contain terms which occur in the final result f.

EXAMPLE 4. Let us reconsider the function f from example 2. When approximating f using $\check{g} = \mathrm{stab}_3\,\check{f}$ instead of $\check{f}$, we get:

$$\begin{aligned}\check{g}_0 &= 0\\ \check{g}_1 &= \frac{\log(x)}{x\,\mathrm{e}^x} + \frac{1}{x\,\mathrm{e}^x}\\ \check{g}_k &= \check{f}_k \qquad (k=2,3,4,5)\end{aligned}$$

EXAMPLE 5. An example for which (12) is *not* satisfied for any finite $k\in\mathbb{N}$ is the expander

$$\check{f}(z) = \frac{1}{1-\frac{z}{x^2}} - \frac{1}{1-\frac{z^2}{x^2}},$$

which arises during the computation of

$$f = \frac{1}{1-\frac{1}{x\,x}} - \frac{1}{1+\frac{1}{x}}\,\frac{1}{1-\frac{1}{x}}.$$

Indeed, the first terms of $\check{f}$ are given by

$$\begin{aligned}\check{f}_0 &= 0\\ \check{f}_1 &= x^{-2}\\ \check{f}_2 &= x^{-4}\\ \check{f}_3 &= x^{-4} + x^{-6}\\ \check{f}_4 &= x^{-6} + x^{-8}\\ \check{f}_5 &= x^{-6} + x^{-8} + x^{-10}\end{aligned}$$

In this kind of situations, it may be necessary to consider more powerful stabilizations of the form $\mathrm{stab}_{n\mapsto\alpha n+\beta}$.

Dominant terms. In the case when (12) holds, we have

$$\tau_f = \tau_{(\mathrm{stab}_k \check{f})_{;n}} \tag{13}$$

for a sufficiently large value of n. When taking k and n fixed, the formula (13) also provides us with a reasonable heuristic for the computation of τ_f (which implies a zero-test for f). Of course, the values k and n should not be taken too small, so as to provide sufficient robustness. On the other hand, large values of k and n may lead to unacceptable computation times. Our current compromise $k = n = 3$ has worked for all practical examples we have tried so far.

REMARK 6. Our claim that relatively small values of k and n provide sufficient robustness may seem very surprising at first sight and is indeed *the* remarkable feature which make meta-expansions so useful in our opinion. The intuitive justification lies in the fact that we expand in a really massive way all our operations and all our parameters. On the one hand, canceling terms usually change after every step before they vanish, and are thereby "stabilized out". On the other hand, deeper combinations of parameters which lead to a genuine non-canceling contribution can usually be detected after a few steps. In particular, small power series expressions with large valuations [vdH06b] tend to be less harmful in our context.

REMARK 7. Let $\mathcal{E}$ be the class of expanders which are obtained by applying our expansion algorithms to exp-log expressions. From a theoretical point of view, it might be interesting to investigate the existence of a simple computable function $\phi: \mathbb{N} \to \mathbb{N}$ such that, for any $\check{f} \in \mathcal{E}$, there exists a k with

$$(\mathrm{stab}_\phi \check{f})_{;k} \sqsubseteq (\mathrm{stab}_\phi \check{f})_{;k+1} \sqsubseteq (\mathrm{stab}_\phi \check{f})_{;k+2} \sqsubseteq \cdots.$$

Generalizing example 5, we see that we must take $\phi(n) \succ n$. Would $\phi(n) = n^2$ be sufficient?

Printing. Another application of the stabilization operator is printing. The default printing method of an expander $\check{f}$ might for instance be to print $(\mathrm{stab}_k \check{f})_n$ for suitable values of n and k (e.g. $n = 5$ and $k = 3$). This method can be further improved as follows: first compute $\varphi = (\mathrm{stab}_{k+1} \check{f})_n$ and $\psi = (\mathrm{stab}_k \check{f})_{n+1}$ with $\varphi \sqsubseteq \psi$. When considering the successive terms of ψ in decreasing order for $\succcurlyeq$, we may decompose the expansion ψ in blocks

$$\psi = \psi_1 + \psi^1 + \psi_2 + \psi^2 + \cdots + \psi_p + \psi^p, \tag{14}$$

with $\psi_i \sqsubseteq \varphi$, $\psi^i \sqsubseteq \psi - \varphi$, $\psi_2 \neq 0, \ldots, \psi_p \neq 0$ and $\psi^1 \neq 0, \ldots, \psi^{p-1} \neq 0$. In (14), we now replace each non-zero ψ^i by the expression $O(\mathfrak{d}_{\psi^i})$, and print the result. For instance, if

$$\begin{aligned}
\varphi &= 1 + \frac{1}{x} + \frac{1}{x^2} + \frac{1}{\mathrm{e}^x} + \frac{1}{x\,\mathrm{e}^x} + \frac{1}{\mathrm{e}^{2x}} \\
\psi &= 1 + \frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \frac{1}{x^4} + \frac{1}{\mathrm{e}^x} + \frac{1}{x\,\mathrm{e}^x} + \frac{1}{x^2\,\mathrm{e}^{2x}} + \frac{1}{\mathrm{e}^{2x}} + \frac{1}{x\,\mathrm{e}^{2x}} + \frac{1}{\mathrm{e}^{3x}},
\end{aligned}$$

then we print

$$1 + \frac{1}{x} + \frac{1}{x^2} + O\left(\frac{1}{x^3}\right) + \frac{1}{\mathrm{e}^x} + \frac{1}{x\,\mathrm{e}^x} + O\left(\frac{1}{x^2\,\mathrm{e}^{2x}}\right) + \frac{1}{\mathrm{e}^{2x}} + O\left(\frac{1}{x\,\mathrm{e}^{2x}}\right).$$

An interesting feature of this way of printing is that it allows us to see some of the remaining terms after the first ω leading terms. In certain cases, such as

$$\frac{1 - x^{-100}}{1 - x^{-1}} = 1 + \frac{1}{x} + \frac{1}{x^2} + O\left(\frac{1}{x^3}\right) + \frac{1}{x^{100}} + \frac{1}{x^{101}} + O\left(\frac{1}{x^{102}}\right),$$

one might prefer to suppress some of these extra terms. One criterion for suppression could be the following: given the last term τ_1 of some ψ_i and any term τ_2 of ψ^i, suppress all terms $\tau_3 \prec \tau_2$ with $\tau_3 \succ \tau_1 (\tau_2/\tau_1)^k$ for some $k \in \mathbb{R}$.

Dominant bias. If you are mainly interested in the first ω terms of an expansion, then you may want to give early terms a higher priority during the computations. Given an expander $\check{f}$, let $\check{f}_{n,k}$ denote the k-th term in the polynomial $\check{f}_n$ (in decreasing order for $\succcurlyeq$). If k is larger than the number of terms of $\check{f}_n$, then we set $\check{f}_{n,k} = 0$. Now for each computable increasing function $\phi: \mathbb{N}^2 \to \mathbb{N}$, we define

$$(\mathrm{bias}_\phi \check{f})_l = \sum_{\phi(n,k)=l} \check{f}_{n,k}.$$

In the case when $\phi(n,k)$, then we have

$$\begin{aligned} (\mathrm{bias}_\phi \check{f})(z) &= \tilde{f}(z,z) \\ \tilde{f}(u,z) &= \sum_{k,n} \check{f}_{n,k}\, u^k\, z^n. \end{aligned}$$

We call bias_ϕ a *dominant bias* operator. We may typically apply it on auxiliary series $\check{f}$ with a sharp increase of the number of terms of $\check{f}_n$ with n. More generally, it is possible to define operators which favour terms at the tail, in the middle, or close to a specified monomial. However, these generalization do not seem to have any practical applications, at first sight.

7. Expansion gallery

Most of the algorithms described in this paper have been implemented inside the Mathemagix system [vdH02b]. Below, we illustrate the implementation with a sample session.

7.1. Exp-log functions

```
Mmx] use "numerix"; use "algebramix"; use "multimix"; use "symbolix";
Mmx] x == infinity ('x);
Mmx] 1 / (x + 1)
```

$$\frac{1}{x} - \frac{1}{x^2} + \frac{1}{x^3} - \frac{1}{x^4} + O\left(\frac{1}{x^5}\right)$$

```
Mmx] 1 / (x + log x + log log x)
```

$$\frac{1}{x} - \frac{\log(x)}{x^2} - \frac{\log(\log(x))}{x^2} + \frac{\log(x)^2}{x^3} + \frac{2\log(x)\log(\log(x))}{x^3} + \frac{\log(\log(x))^2}{x^3} - \frac{\log(x)^3}{x^4} - \frac{3\log(x)^2\log(\log(x))}{x^4} - \frac{3\log(x)\log(\log(x))^2}{x^4} - \frac{\log(\log(x))^3}{x^4} + O\left(\frac{\log(x)^4}{x^5}\right)$$

```
Mmx] 1 / (1 + 1/x + 1/exp x)
```

$$1 - \frac{1}{x} + \frac{1}{x^2} - \frac{1}{x^3} + O\left(\frac{1}{x^4}\right) - \frac{1}{\mathrm{e}^x} + \frac{2}{x\,\mathrm{e}^x} - \frac{3}{x^2\,\mathrm{e}^x} + O\left(\frac{1}{x^3\,\mathrm{e}^x}\right) + \frac{1}{\mathrm{e}^{2x}} - \frac{3}{x\,\mathrm{e}^{2x}} + O\left(\frac{1}{x^2\,\mathrm{e}^{2x}}\right) - \frac{1}{\mathrm{e}^{3x}} + O\left(\frac{1}{x\,\mathrm{e}^{3x}}\right)$$

```
Mmx] 1 / (1 + 1/x + 1/exp x) - 1 / (1 + 1/x)
```

$$\frac{-1}{\mathrm{e}^{x}}+\frac{2}{x\,\mathrm{e}^{x}}-\frac{3}{x^{2}\,\mathrm{e}^{x}}+O\left(\frac{1}{x^{3}\,\mathrm{e}^{x}}\right)+\frac{1}{\mathrm{e}^{2x}}-\frac{3}{x\,\mathrm{e}^{2x}}+O\left(\frac{1}{x^{2}\,\mathrm{e}^{2x}}\right)-\frac{1}{\mathrm{e}^{3x}}+O\left(\frac{1}{x\,\mathrm{e}^{3x}}\right)$$

```
Mmx] exp (x + exp (-exp x)) - exp (x)
```

$$\frac{1}{\mathrm{e}^{\mathrm{e}^{x}-x}}+\frac{1}{2\,\mathrm{e}^{2\mathrm{e}^{x}-x}}+\frac{1}{6\,\mathrm{e}^{3\mathrm{e}^{x}-x}}+O\left(\frac{1}{\mathrm{e}^{4\mathrm{e}^{x}-x}}\right)$$

```
Mmx] exp (exp (x) / (x + 1))
```

$$\mathrm{e}^{\frac{\mathrm{e}^{x}}{x}-\frac{\mathrm{e}^{x}}{x^{2}}+\frac{\mathrm{e}^{x}}{x^{3}}-\frac{\mathrm{e}^{x}}{x^{4}}+O\left(\frac{\mathrm{e}^{x}}{x^{5}}\right)}$$

7.2. Calculus

```
Mmx] derive (exp (exp (x) / (x + 1)), x)
```

$$\frac{\mathrm{e}^{\frac{\mathrm{e}^{x}}{x}-\frac{\mathrm{e}^{x}}{x^{2}}+\frac{\mathrm{e}^{x}}{x^{3}}-\frac{\mathrm{e}^{x}}{x^{4}}+O\left(\frac{\mathrm{e}^{x}}{x^{5}}\right)+x}}{x}-\frac{2\,\mathrm{e}^{\frac{\mathrm{e}^{x}}{x}-\frac{\mathrm{e}^{x}}{x^{2}}+\frac{\mathrm{e}^{x}}{x^{3}}-\frac{\mathrm{e}^{x}}{x^{4}}+O\left(\frac{\mathrm{e}^{x}}{x^{5}}\right)+x}}{x^{2}}+\frac{3\,\mathrm{e}^{\frac{\mathrm{e}^{x}}{x}-\frac{\mathrm{e}^{x}}{x^{2}}+\frac{\mathrm{e}^{x}}{x^{3}}-\frac{\mathrm{e}^{x}}{x^{4}}+O\left(\frac{\mathrm{e}^{x}}{x^{5}}\right)+x}}{x^{3}}-\frac{4\,\mathrm{e}^{\frac{\mathrm{e}^{x}}{x}-\frac{\mathrm{e}^{x}}{x^{2}}+\frac{\mathrm{e}^{x}}{x^{3}}-\frac{\mathrm{e}^{x}}{x^{4}}+O\left(\frac{\mathrm{e}^{x}}{x^{5}}\right)+x}}{x^{4}}+O\left(\frac{\mathrm{e}^{\frac{\mathrm{e}^{x}}{x}-\frac{\mathrm{e}^{x}}{x^{2}}+\frac{\mathrm{e}^{x}}{x^{3}}-\frac{\mathrm{e}^{x}}{x^{4}}+O\left(\frac{\mathrm{e}^{x}}{x^{5}}\right)+x}}{x^{5}}\right)$$

```
Mmx] integrate (exp (x^2), x)
```

$$\frac{\mathrm{e}^{x^{2}}}{2\,x}+\frac{\mathrm{e}^{x^{2}}}{4\,x^{3}}+\frac{3\,\mathrm{e}^{x^{2}}}{8\,x^{5}}+\frac{15\,\mathrm{e}^{x^{2}}}{16\,x^{7}}+O\left(\frac{\mathrm{e}^{x^{2}}}{x^{9}}\right)$$

```
Mmx] integrate (x^x, x)
```

$$\frac{\mathrm{e}^{x\log(x)}}{\log(x)}-\frac{\mathrm{e}^{x\log(x)}}{\log(x)^{2}}+\frac{\mathrm{e}^{x\log(x)}}{\log(x)^{3}}-\frac{\mathrm{e}^{x\log(x)}}{\log(x)^{4}}+O\left(\frac{\mathrm{e}^{x\log(x)}}{\log(x)^{5}}\right)+\frac{\mathrm{e}^{x\log(x)}}{x\log(x)^{3}}-\frac{3\,\mathrm{e}^{x\log(x)}}{x\log(x)^{4}}+\frac{6\,\mathrm{e}^{x\log(x)}}{x\log(x)^{5}}+O\left(\frac{\mathrm{e}^{x\log(x)}}{x\log(x)^{6}}\right)+\frac{\mathrm{e}^{x\log(x)}}{x^{2}\log(x)^{4}}-\frac{\mathrm{e}^{x\log(x)}}{x^{2}\log(x)^{5}}+O\left(\frac{\mathrm{e}^{x\log(x)}}{x^{2}\log(x)^{6}}\right)+\frac{2\,\mathrm{e}^{x\log(x)}}{x^{3}\log(x)^{5}}+O\left(\frac{\mathrm{e}^{x\log(x)}}{x^{4}\log(x)^{6}}\right)$$

```
Mmx] sum (x^4, x)
```

$$\frac{x^{5}}{5}-\frac{x^{4}}{2}+\frac{x^{3}}{3}-\frac{x}{30}$$

```
Mmx] product (x, x)
```

$$\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}+\frac{\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}}{12\,x}+O\left(\frac{\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}}{x^{2}}\right)$$

```
Mmx] lengthen (product (x, x), 8)
```

$$\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}+\frac{\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}}{12\,x}+\frac{\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}}{288\,x^{2}}-\frac{139\,\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}}{51840\,x^{3}}-\frac{571\,\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}}{2488320\,x^{4}}+\frac{163879\,\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}}{209018880\,x^{5}}+O\left(\frac{\mathrm{e}^{x\log(x)-x-\frac{\log(x)}{2}}}{x^{6}}\right)$$

```
Mmx] product (log x, x)
```

$$\mathrm{e}^{x\log(\log(x))-\frac{x}{\log(x)}-\frac{x}{\log(x)^2}-\frac{2x}{\log(x)^3}+O\left(\frac{x}{\log(x)^4}\right)-\frac{\log(\log(x))}{2}}+\frac{\mathrm{e}^{x\log(\log(x))-\frac{x}{\log(x)}-\frac{x}{\log(x)^2}-\frac{2x}{\log(x)^3}+O\left(\frac{x}{\log(x)^4}\right)-\frac{\log(\log(x))}{2}}}{12\,x\log(x)}+O\left(\frac{\mathrm{e}^{x\log(\log(x))-\frac{x}{\log(x)}-\frac{x}{\log(x)^2}-\frac{2x}{\log(x)^3}+O\left(\frac{x}{\log(x)^4}\right)-\frac{\log(\log(x))}{2}}}{x^2\log(x)^2}\right)$$

7.3. Functional equations

```
Mmx] fixed_point (f :-> log x + f @ (log x))
```

$$\log(x)+\log(\log(x))+\log(\log(\log(x)))+\log(\log(\log(\log(x))))+O(\log(\log(\log(\log(\log(x))))))$$

```
Mmx] la == derive (fixed_point (f :-> log x + f @ (log x)), x)
```

$$\frac{1}{x}+\frac{1}{x\log(x)}+\frac{1}{x\log(x)\log(\log(x))}+\frac{1}{x\log(x)\log(\log(x))\log(\log(\log(x)))}+O\left(\frac{1}{x\log(x)\log(\log(x))\log(\log(\log(x)))\log(\log(\log(\log(x))))}\right)$$

```
Mmx] mu == la * la + 2 * derive (la, x)
```

$$\frac{-1}{x^2}-\frac{1}{x^2\log(x)^2}+O\left(\frac{1}{x^2\log(x)^2\log(\log(x))^2}\right)$$

```
Mmx] fixed_point (f :-> 1/x + f @ (x^2) + f @ (x^x))
```

$$\frac{1}{x}+\frac{1}{x^2}+\frac{1}{x^4}+\frac{1}{x^8}+O\left(\frac{1}{x^{16}}\right)+\frac{1}{\mathrm{e}^{x\log(x)}}+\frac{1}{\mathrm{e}^{2x\log(x)}}+\frac{1}{\mathrm{e}^{4x\log(x)}}+O\left(\frac{1}{\mathrm{e}^{8x\log(x)}}\right)+\frac{1}{\mathrm{e}^{2x^2\log(x)}}+\frac{1}{\mathrm{e}^{4x^2\log(x)}}+O\left(\frac{1}{\mathrm{e}^{8x^2\log(x)}}\right)+\frac{1}{\mathrm{e}^{4x^4\log(x)}}+O\left(\frac{1}{\mathrm{e}^{8x^4\log(x)}}\right)+\frac{1}{\mathrm{e}^{x\log(x)\mathrm{e}^{x\log(x)}}}+\frac{1}{\mathrm{e}^{2x\log(x)\mathrm{e}^{x\log(x)}}}+O\left(\frac{1}{\mathrm{e}^{4x\log(x)\mathrm{e}^{x\log(x)}}}\right)+\frac{1}{\mathrm{e}^{2x\log(x)\mathrm{e}^{2x\log(x)}}}+O\left(\frac{1}{\mathrm{e}^{4x\log(x)\mathrm{e}^{2x\log(x)}}}\right)+\frac{1}{\mathrm{e}^{2x^2\log(x)\mathrm{e}^{2x^2\log(x)}}}+O\left(\frac{1}{\mathrm{e}^{4x^2\log(x)\mathrm{e}^{2x^2\log(x)}}}\right)+\frac{1}{\mathrm{e}^{x\log(x)\mathrm{e}^{x\log(x)\mathrm{e}^{x\log(x)}+x\log(x)}}}+O\left(\frac{1}{\mathrm{e}^{2x\log(x)\mathrm{e}^{x\log(x)\mathrm{e}^{x\log(x)}+x\log(x)}}}\right)$$

8. Towards more robustness

Even though the expansion algorithms developed so far are usually sufficient for applications, they lack robustness in several ways. First of all, we have used heuristic algorithms for zero-testing and the computation of dominant terms. Some of our algorithms crucially depend on the correctness of these heuristic algorithms. For instance, our algorithm for the computation of an inverse $f = g^{-1}$ yields an erroneous result if τ_g is computed incorrectly. Finally, expanders $(\check{f}_n)$ only asymptotically tend to f. Even if we know that a given monomial $\mathfrak{m} \in \mathfrak{M}$ is in the support of f, we do not know how large n should be in order to guarantee that $f_{\mathfrak{m}} = \check{f}_{;n,\mathfrak{m}}$. In this section, we describe a few ideas which may be used to increase the robustness of our algorithms.

Auto-correction. The strategy of *auto-correction* can be used to reduce the impact of incorrect answers of heuristic algorithms. For instance, in order to invert a series g, we started with the computation of τ_g. Instead, we might imagine that the expander of $f = g^{-1}$ is allowed to adjust its initial value of τ_g at later stages of the approximation. More precisely, for $n \in \mathbb{N}$ and a suitable ϕ, let τ_n be the dominant term of $(\mathrm{stab}_\phi\, g)_{;n}$ and consider the expander

$$\check{\varphi}_n = \begin{cases} \tau_n^{-1} \frac{1}{1 + z\,(\tau_n^{-1}\, g - 1)} & \text{if } \tau_n \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Then we may define a new expander $\check{f}$ by taking the diagonal

$$\check{f}_{;n} = \check{\varphi}_{n;n}.$$

In order to have $f = g^{-1}$, it now suffices to have $\tau_n = \tau_f$ $(n \geqslant n_0)$, instead of $\tau_0 = \tau_f$. Of course, in nasty cases, it might still happen that $\tau_n \neq \tau_f$ for all n. In other words, the strategy of auto-correction produces no miracles and does not substitute for a genuine zero-test. Even in the case when the stabilization operator stab_ϕ is sufficiently powerful to guarantee $f = g^{-1}$ for a certain class of expanders, one should keep in mind that the result only becomes correct at the limit; we still don't know how many terms need to be computed.

From a practical point, the strategy of auto-correction is easy to implement on the series level from section 3: in our example of inversion, the expander $\check{f}$ may simply keep $\check{\varphi}_n$ in memory for the largest n considered so far and only update its value when τ_n changes. Implementations become more involved when considering recursive transseries expansions, as in section 4. Indeed, in this more general setting, we also need to correct erroneous outcomes for the asymptotic ordering $\preccurlyeq$, which recursively relies on the orderings $\leqslant$ and $\preccurlyeq$ for expanders of lower exponential height. In order to generalize the idea, one thus has to define sequences of approximations $\preccurlyeq_n$ and $\leqslant_n$ for $\preccurlyeq$ and $\leqslant$, and systematically work with the relations $\preccurlyeq_n$ and $\leqslant_n$ when making decisions for expansions at stage n.

Reduction to zero-testing. In the case when we are interested in expansions of functions inside a given class $\mathcal{F}$, it sometimes happens that $\mathcal{F}$ admits a zero-test. For instance, if $\mathcal{F}$ is the class of exp-log functions, then a zero-test can be given whose correctness relies on Schanuel's conjecture [vdH98, Ric97].

An interesting question is whether we can use a zero-test in $\mathcal{F}$ in order to design a non-heuristic algorithm for the computation of dominant terms. In order to make this work, we have to be able to detect infinite cancellations of terms which occur in expressions such as $\log(1 + x^{-1} + \mathrm{e}^{-x}) - \log(1 + x^{-1})$. A general mechanism for doing this is to refine the mechanism of expanders by indexing over a suitable well-based set.

More precisely, given an abstract well-based set $\mathfrak{A}$ (i.e. a set $\mathfrak{A}$ which is well-quasi-ordered for the opposite ordering of $\preccurlyeq$), we define a sequence $\mathfrak{A}_0, \mathfrak{A}_1, \ldots$ of finite subsets of $\mathfrak{A}$ by $\mathfrak{A}_i = \max_\preccurlyeq \mathfrak{A} \setminus (\mathfrak{A}_0 \cup \cdots \cup \mathfrak{A}_{i-1})$. In general, the sequence $(\mathfrak{A}_\alpha)$ is transfinite, but if we have $\mathfrak{A} = \bigcup_{n \in \mathbb{N}} \mathfrak{A}_n$, then we say that $\mathfrak{A}$ is *accessible*. We say that $\mathfrak{A}$ is *computable*, if for any finite subset $\mathfrak{F} \subseteq \mathfrak{A}$, we can compute $\max_\preccurlyeq \{\mathfrak{a} \in \mathfrak{A} : \forall \mathfrak{f} \in \mathfrak{F}, \mathfrak{a} \prec \mathfrak{f}\}$. In particular, this implies the sequence $n \mapsto \mathfrak{A}_n$ to be computable.

A *well-based expander* is a computable well-based family $\check{f} = (\check{f}_\mathfrak{a})_{\mathfrak{a} \in \mathfrak{A}} \in C[\mathfrak{M}]^\mathfrak{A}$, indexed by a computable accessible well-based set $\mathfrak{A}$. A well-based expander is the natural refinement of an expander (g_n) in the usual sense, by regrouping terms $\check{g}_n = \check{f}_{\mathfrak{A}_n} = \sum_{\mathfrak{a} \in \mathfrak{A}_n} \check{f}_\mathfrak{a}$. We say that $\check{f}$ is a *termwise well-based expander* if each $\check{f}_\mathfrak{a}$ is of the form $\check{f}_\mathfrak{a} = c_\mathfrak{a}\, \mathfrak{m}_\mathfrak{a} \in R\,\mathfrak{M}$ and the mapping $\mathfrak{a} \mapsto \mathfrak{m}_\mathfrak{a}$ is increasing. Notice that $\check{f}$ is automatically well-based if $\mathfrak{a} \mapsto \mathfrak{m}_\mathfrak{a}$ is increasing.

Recall that the *initial segment* generated by a finite subset $\mathfrak{F} \subseteq \mathfrak{A}$ is defined by $(\mathfrak{F}) = \{\mathfrak{a} \in \mathfrak{A}: \exists \mathfrak{f} \in \mathfrak{F}, \mathfrak{a} \preccurlyeq \mathfrak{f}\}$. Now consider a termwise well-based expander $(\check{f}_\mathfrak{a})_{\mathfrak{a}\in\mathfrak{A}}$ such that $\check{f}_{(\mathfrak{F})} \in \mathcal{F}$ for any finite subset $\mathfrak{F} \subseteq \mathfrak{A}$. If the mapping $\mathfrak{F} \mapsto \check{f}_{(\mathfrak{F})}$ is effective, then we call $\check{f}$ an *expander over* $\mathcal{F}$. For the operations we have considered in this paper, it should be possible to replace the usual notion of expander by expanders of $\mathcal{F}$ (assuming that $\mathcal{F}$ is stable under the operation). This was already shown in [vdH94] for the basic operations from section 3 and still waits to be worked out for the other ones.

Given an expander $(\check{f}_\mathfrak{a})_{\mathfrak{a}\in\mathfrak{A}}$ over $\mathcal{F}$, the zero-test in $\mathcal{F}$ may now be used in order to compute the set $\mathfrak{D}_f = \max_\preccurlyeq \operatorname{supp} f$ of dominant monomials of f. The algorithm again goes back to [vdH94]:

1. Let $\mathfrak{S} := \max_\preccurlyeq \mathfrak{A}$.
2. Replace $\mathfrak{S}$ by a minimal subset $\mathfrak{T}$ with $\check{f}_{(\mathfrak{S})} = \check{f}_{(\mathfrak{T})}$.
3. Let $\mathfrak{N} := \max_\preccurlyeq \{\mathfrak{d}_{\check{f}_\mathfrak{s}} : \mathfrak{s} \in \mathfrak{S}\}$ and $\mathfrak{S}_\mathfrak{n} := \{\mathfrak{s} \in \mathfrak{S}: \mathfrak{d}_{\check{f}_\mathfrak{s}} = \mathfrak{n}\}$ for each $\mathfrak{n} \in \mathfrak{N}$.
4. If there exists an $\mathfrak{n} \in \mathfrak{N}$ with $\check{f}_{\mathfrak{S}_\mathfrak{n}} = 0$, then set
$$\mathfrak{S} := \max_\preccurlyeq \left[(\mathfrak{S} \setminus \mathfrak{S}_\mathfrak{n}) \cup \max_\preccurlyeq \{\mathfrak{a} \in \mathfrak{A}: \exists \mathfrak{s} \in \mathfrak{S}_\mathfrak{n}, \mathfrak{a} \prec \mathfrak{s}\} \right],$$
and go to step 2.
5. Return $\mathfrak{S}$.

Computable well-based series. Our notion of approximable well-based series is the natural counterpart of the concept of approximable real numbers: a real number x is said to be *approximable* if there exists a computable sequence $\tilde{x}: \mathbb{N} \to \mathbb{Q}$, which converges to x. A stronger and more robust notion is the one of computable real numbers: we say that $x \in \mathbb{R}$ is *computable*, if there exists a computable function $\check{x}: \mathbb{Q}^> \to \mathbb{Q}$ which takes $\varepsilon \in \mathbb{Q}^>$ on input and produces an approximation $\tilde{x} = \check{x}(\varepsilon) \in \mathbb{Q}$ with $|\tilde{x} - x| < \varepsilon$. It is natural to search for an analogue notion of computable well-based series.

There are really two aspects to a computable well-based series f. On the one hand, we should be able to compute its coefficient $f_\mathfrak{m}$ for any monomial $\mathfrak{m}$. On the other hand, its support should be sufficiently effective. In the case of ordinary power series $f \in R[[z]]$, the second issue does not really arise, because the support is necessarily included in the well-based set $z^\mathbb{N}$. In general, one might require that f is given by a termwise well-based expander, which yields quite a lot of information about $\operatorname{supp} f$.

As to the computation of coefficients $f_\mathfrak{m}$, consider the case of a product $f = g\,h$, where $\mathfrak{M}$ is totally ordered and g and h are given by $g = 1 + \mathfrak{v}_1 + \mathfrak{v}_2 + \cdots$ and $h = 1 + \mathfrak{w}_1 + \mathfrak{w}_2 + \cdots$ with $1 + \mathfrak{v}_1 \succ \mathfrak{v}_2 \succ \cdots$ and $1 + \mathfrak{w}_1 \succ \mathfrak{w}_2 \succ \cdots$. Given $\mathfrak{m} \in \mathfrak{M}$, we hit the problem that we don't have any *a priori* information on the asymptotic behaviour of the $\mathfrak{v}_i$ and the $\mathfrak{w}_i$. In order to design an algorithm for the computation of $f_\mathfrak{m}$, we need more control over this asymptotic behaviour.

In the grid-based setting, we are really computing with multivariate power series, and no real difficulties arise. In the well-based setting, things get more involved. When we restrict our attention to transseries, there are ways to represent and compute with *monomial cuts* [vdH06c, Chapter 9]. A monomial cut is the analogue of a Dedekind cut for the set of transmonomials instead of $\mathbb{Q}$. Given a termwise well-based expander $(\check{f}_\mathfrak{a})_{\mathfrak{a}\in\mathfrak{A}}$, any initial segment $(\mathfrak{F})$ naturally induces a transseries $\check{f}_{(\mathfrak{F})}$ and a monomial cut, called the width of $\check{f}_{(\mathfrak{F})}$. For a fully satisfactory definition of computable well-based series, one should be able to compute these widths. However, we have not investigated this matter in detail yet.

BIBLIOGRAPHY

[DG86] B. I. Dahn and P. Göring. Notes on exponential-logarithmic terms. *Fundamenta Mathematicae*, 127:45–50, 1986.

[É92] J. Écalle. *Introduction aux fonctions analysables et preuve constructive de la conjecture de Dulac*. Hermann, collection: Actualités mathématiques, 1992.

[GG92] G.H. Gonnet and D. Gruntz. Limit computation in computer algebra. Technical Report 187, ETH, Zürich, 1992.

[Hah07] H. Hahn. Über die nichtarchimedischen Größensysteme. *Sitz. Akad. Wiss. Wien*, 116:601–655, 1907.

[Har10] G.H. Hardy. *Orders of infinity*. Cambridge Univ. Press, 1910.

[Har11] G.H. Hardy. Properties of logarithmico-exponential functions. *Proceedings of the London Mathematical Society*, 10(2):54–90, 1911.

[Hig52] G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc.*, 2:326–336, 1952.

[Mil85] E. C. Milner. Basic wqo- and bqo- theory. In Rival, editor, *Graphs and orders*, pages 487–502. D. Reidel Publ. Comp., 1985.

[MMvdD97] A. Macintyre, D. Marker, and L. van den Dries. Logarithmic-exponential power series. *Journal of the London Math. Soc.*, 56(2):417–434, 1997.

[MMvdD99] A. Macintyre, D. Marker, and L. van den Dries. Logarithmic exponential series. *Annals of Pure and Applied Logic*, 1999. To appear.

[Pou85] M. Pouzet. Applications of well quasi-ordering and better quasi-ordering. In Rival, editor, *Graphs and orders*, pages 503–519. D. Reidel Publ. Comp., 1985.

[PTVF07] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes, the art of scientific computing*. Cambridge University Press, 3rd edition, 2007.

[Ric97] D. Richardson. How to recognise zero. *JSC*, 24:627–645, 1997.

[Ric07] D. Richardson. Zero tests for constants in simple scientific computation. *MCS*, 1(1):21–37, 2007.

[RSSvdH96] D. Richardson, B. Salvy, J. Shackell, and J. van der Hoeven. Expansions of exp-log functions. In Y.N. Lakhsman, editor, *Proc. ISSAC '96*, pages 309–313, Zürich, Switzerland, July 1996.

[Sal91] B. Salvy. *Asymptotique automatique et fonctions génératrices*. PhD thesis, École Polytechnique, France, 1991.

[Sch01] M.C. Schmeling. *Corps de transséries*. PhD thesis, Université Paris-VII, 2001.

[Sha90] J. Shackell. Growth estimates for exp-log functions. *Journal of Symbolic Computation*, 10:611–632, 1990.

[Sha93] J. Shackell. Inverses of Hardy L-functions. *Bull. of the London Math. Soc.*, 25:150–156, 1993.

[vdH94] J. van der Hoeven. Outils effectifs en asymptotique et applications. Technical Report LIX/RR/94/09, LIX, École polytechnique, France, 1994.

[vdH97] J. van der Hoeven. *Automatic asymptotics*. PhD thesis, École polytechnique, Palaiseau, France, 1997.

[vdH98] J. van der Hoeven. Generic asymptotic expansions. *AAECC*, 9(1):25–44, 1998.

[vdH01] J. van der Hoeven. Operators on generalized power series. *Journal of the Univ. of Illinois*, 45(4):1161–1190, 2001.

[vdH02a] J. van der Hoeven. Relax, but don't be too lazy. *JSC*, 34:479–542, 2002.

[vdH02b] J. van der Hoeven et al. Mathemagix, 2002. `http://www.mathemagix.org`.

[vdH06a] J. van der Hoeven. Algorithms for asymptotic interpolation. Technical Report 2006-12, Univ. Paris-Sud, 2006. Submitted to JSC.

[vdH06b] J. van der Hoeven. Counterexamples to witness conjectures. *JSC*, 41:959–963, 2006.

[vdH06c] J. van der Hoeven. *Transseries and real differential algebra*, volume 1888 of *Lecture Notes in Mathematics*. Springer-Verlag, 2006.

[vdH07] J. van der Hoeven. On effective analytic continuation. *MCS*, 1(1):111–175, 2007.

[vdHS06] J. van der Hoeven and J.R. Shackell. Complexity bounds for zero-test algorithms. *JSC*, 41:1004–1020, 2006.