数理解析研究所講究録
1096 巻 1999 年 51-67

Lagrange-Good Inversion from Trace

Ryu HASEGAWA[‡] (長谷川 立)

Graduate School of Mathematical Sciences, The University of Tokyo, Komaba 3-8-1, Meguro-ku, Tokyo 153, Japan

Abstract
We give the formula for categorical trace of normal functors, and derive the Lagrange-Good inversion formula from it.

1 Introduction

The purpose of this paper is to give a new proof of the Lagrange-Good inversion formula. The novelty of our proof is in the use of ideas fostered in theoretical computer science.

The Lagrange-Good inversion formula is a method to solve certain recursive equation on formal power series in several variables. As explained in section two, the formula gives a fixed point of certain operations on formal power series. The formula is used to compute the coefficients of the compositional inverse of formal power series in several variables as well as to find the generating functions related to problems of enumerative combinatorics.

For the case of a single variable, it dates back to Lagrange's work in the end of the eighteenth century. Many mathematicians tried to extend Lagrange's formula to several variables. After several extensions to specific number of variables, the general result for arbitrary n variables was settled in Good's paper [11]. The formula remains true even if the number of variables is infinite [7]. Gessel's paper [9] contains a short history of the formula.

There are many proofs of the formula. The proof by Good uses properties of analytic functions [11] (see also [17]). From the interest in enumerative combinatorics, many combinatorial proofs are produced, even in these several years [5, 7, 9, 18]. De Bruijn verified the formula by induction on the number of variables.

We give another proof of the Lagrange-Good inversion formula from complete different perspective. Recently Hasegawa [13, 14] provided the correspondence between fixed point operators and categorical trace in the sense of Joyal, Street and Verity [21] in cartesian categories. In this paper, we give a concrete formula

[‡] Email address: ryu@ms.u-tokyo.ac.jp

of the categorical trace in the category where the morphisms are normal functors [10], which are functorial generalization of formal power series. Then we show that the formula of the trace yields the Lagrange-Good inversion formula through the correspondence by Hasegawa.

What we want to emphasize is that the machinery needed for the proof has been developed in theoretical computer science. One of characteristic points of pure functional programming is that iteration of a procedure is realized by recursive call of functions. Traditionally such recursive call is interpreted as fixed point of an operation on programmes. This interpretation, however, endangers the mathematical foundation of functional programming, since we need an operator, called a fixpoint combinator, that yields the fixed point for every functional. It is a difficult problem to find a mathematical structure satisfying the property that every morphism has a fixed point. This implication leaded Scott to the discovery that certain continuous lattices satisfy the property [27], and to later development of domain theory [12] as the mathematical foundation of functional programming.

Various systems of lambda calculus [2] are upshots of the syntactical features in functional programming. The normal functor model [10] of lambda calculus is in some sense a variation of the mathematical models given in domain theory. An interesting point of normal functors is that they have the same form as formal power series, though the former are functors into the category of all sets. In fact, it is easy to see that normal functors are simply a special case of analytic functors introduced by Joyal as a functorial variation of generating functions in the context of enumerative combinatorics [18, 19]. As in usual domain theory, we have an interpretation of the fixpoint combinator as normal functors, thus in the form of formal power series.

The construction of the interpretation of the fixpoint combinator is inappropriate to compute the actual form of the interpretation. A more convenient form is derived from the correspondence between fixed point operators and categorical traces [13, 14]. Also this result is found in theoretical computer science. We often write a recursive programme informally a graph where the output is fed back to the input. Formally the latter is coded in syntax by what is called the letrec-operator. Joyal, Street and Verity generalized traces in linear algebra etc. to the context of monoidal categories [21]. One of fascinating features is that the trace is figured as a graph with feedback, and that proofs of equalities between formulas containing trace are depicted as intuitive graphical operation not changing their topologies. Hasegawa proved that the fixpoint operators satisfying so-called Bekič's formula in cartesian categories has a one-to-one correspondence to traces. We may say that this result provides a mathematical relation between recursive programmes and their intuitive graphic representations.

Hence, in place of computing fixpoint combinators, we may compute categorical traces. The latter turns out to be simpler, if we decompose the lambda calculus into intuitionistic linear logic [3]. Normal functors give also a model of the intuitionistic linear logic where the interpretation is quite reminiscent of linear algebra. Hence, in this model, we may define trace as diagonal sum as usual. Since what we want to have is a trace in the normal functor model of the lambda calculus rather than that of intuitionistic linear logic, we must modify the presentation of trace in an appropriate way. After the modification, we have a formula of the trace for normal functors, from which a formula of the fixpoint combinator is derived by Hasegawa's correspondence. We verify that the obtained formula yields the well-known Lagrange-Good inversion formula.

We do not claim that our proof is superior to former proofs. What we think to be remarkable is the close connection between a purely mathematical formula and ideas in computer science. We omit the proofs of the results. They will be given in a forthcoming full paper [16].

2 Preliminaries

Lagrange-Good Inversion Formula

First we review the Lagrange-Good inversion formula. Let us consider n formal power series $g_i(x_1, x_2, \ldots, x_n)$ in n variables for $i = 1, 2, \ldots, n$ over a commutative ring. With a new set of variables $z_1, z_2, \ldots, z_n$, we are interested in the system of equations

$$\begin{aligned} x_1 &= z_1 \cdot f_1(x_1, x_2, \ldots, x_n) \\ x_2 &= z_2 \cdot f_2(x_1, x_2, \ldots, x_n) \\ &\vdots \\ x_n &= z_n \cdot f_n(x_1, x_2, \ldots, x_n), \end{aligned}$$

which we write $x = z \cdot g(x)$ simply. The Lagrange-Good inversion formula to find the solution of this equation in x is given as follows.

2.1 Theorem (Lagrange-Good inversion formula)
We consider a system of equations $x = zg(x)$ where $g(x)$ is an n-tuple of formal power series in n variables $x = x_1, x_2, \ldots, x_n$. Moreover, let $h(x)$ be an arbitrary formal power series in n variables.

The equations $x = zg(x)$ has a unique solution $x = a(z)$ in the formal power series ring. Moreover the formal power series a satisfies the formula

$$\frac{h(a(z))}{\det(E - M(a(z), z))} = \sum z^{\gamma}[x^{\gamma}]h(x)g(x)^{\gamma}$$

where γ ranges over all multi-indexes and $M(x,z)$ is the square matrix defined by $(z_i\partial g_i(x)/\partial x_k)_{ik}$. Here the notation x^γ equals $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ if the multi-index γ is $(k_1,k_2,\ldots,k_n)$ and the notation $[x^\gamma]f(x)$ gives the coefficient of x^γ in the formal power series f.

Remark: (i) In order to compute the solution $x = a(z)$ itself, it suffices to put $h(x) = x_i$ for $i = 1,2,\ldots,n$.

(ii) The theorem holds for formal power series over an arbitrary commutative ring. In fact, over an arbitrary commutative semiring, the theorem remains to be valid. Although the formula involves a determinant that may have a negative coefficients, the formal power series $1/\det(E - M(a(z),z))$ contains only nonnegative coefficients if we regard the coefficients of g as indeterminates.

(iii) The theorem remains to hold if $h(x)$ is a formal Laurent series [9], although $g(x)$ must be formal power series.

The form in the theorem above is not appropriate to compute the solution $x = a(z)$. For this purpose, we have the following equivalent formula:

$$h(a(z)) = \sum z^\gamma[x^\gamma]h(x)g(x)^\gamma \det(E - M(x, x/g(x)))$$

where the division $x/g(x)$ is componentwise, that is, the i-th component is $x_i/g_i(x_1,x_2,\ldots,x_n)$. To derive this formula, it suffices to replace $h(x)$ in the theorem by $h(x)\det(E - M(x,z))$ which is a formal power series over the ring of polynomials in z.

Remark: The determinant $\det(E - M(x, x/g(x)))$ can be written in use of Jacobian as $g_1(x)\cdots g_n(x)\times\partial(f_1,\ldots,f_n)/\partial(x_1,\ldots,x_n)$ where $f_i(x_1,x_2,\ldots,x_n) = x_i/g_i(x_1,x_2,\ldots,x_n)$.

Employing the Lagrange-Good inversion, we can compute the compositional inverse of systems $z = f(x)$ of formal power series over a field in the following form:

$$f_i(x_1,x_2,\ldots,x_n) = x_i(a_i + \text{higher degree terms}), \qquad a_i \neq 0$$

for $i = 1,2,\ldots,n$. In fact, the inverse $x = f^{-1}(z)$ should satisfy the equation $x = z\cdot(x/f(x))$. By the form of f, the formula $x/f(x)$ turns out to be a formal power series by the binomial theorem. So, if we put $g(x) = x/f(x)$ in the Lagrange-Good inversion formula, we can compute the inverse $x = f^{-1}(x)$ as the solution $x = a(z)$ of the equation $x = zg(x)$.

PCF and Intuitionistic Linear Logic

We define system *PCF* of typed lambda calculus with fixpoint combinator as well as arithmetic and Boolean operations [25, 24]. The types σ of PCF are given by the following Backus-Naur form:

$$\sigma \ ::= \ \iota \mid o \mid \sigma \Rightarrow \sigma.$$

The type ι is regarded as the type of natural numbers, and the type o as that of Boolean values. The terms M are given by the following syntax:

$$\begin{array}{rcl} M & ::= & x \mid MM \mid \lambda x^{\sigma}.M \mid \mathsf{fix}\,M \\ & & \mid \mathsf{succ}\,M \mid \mathsf{pred}\,M \mid \mathsf{zero?}\,M \\ & & \mid \mathsf{cond}\,MMM \mid \mathsf{t} \mid \mathsf{f} \mid n\,. \end{array}$$

Here x is a variable from a fixed countable set, and n ranges over the set of natural numbers $0, 1, \ldots$. The typing rules are obvious and we omit them. We consider the following reduction rules:

$$\begin{array}{lcl} (\lambda x^{\sigma}.M)N & \to & M[N/x] \\ \mathsf{fix}\,M & \to & M(\mathsf{fix}\,M) \end{array}$$

$$\begin{array}{lcllcl} \mathsf{succ}\,n & \to & n+1 & & & \\ \mathsf{pred}\,n+1 & \to & n & \mathsf{pred}\,0 & \to & 0 \\ \mathsf{zero?}\,n+1 & \to & \mathsf{t} & \mathsf{zero?}\,0 & \to & \mathsf{f} \\ \mathsf{cond}\,\mathsf{t}\,MN & \to & M & \mathsf{cond}\,\mathsf{f}\,MN & \to & N. \end{array}$$

We define the ***intuitionistic linear logic***, which is given as a system of typed calculus [3]. The types A are generated by the following form:

$$A \ := \ \alpha \mid A \otimes A \mid I \mid A \multimap A \mid\ !A$$

where α ranges over atomic types. For example, for the system to correspond to PCF, we may let α be either ι or o. The derivation rules for typing judgements are given in Tab. 2.2, which defines the terms of intuitionistic linear logic at the same time.

The reduction rules for the term calculus of intuitionistic linear logic are not completely established. See [4], for example. It does not matter which ones we take, for miscellaneous rules. The core rules are the β-reductions given by the following six:

$$\begin{array}{lcl} \mathsf{apply}(\lambda x^{A}.\,e)\ \mathsf{to}\ f & \to & e[f/x] \\ \mathsf{let}\,*\,\mathsf{be}\,*\,\mathsf{in}\ f & \to & f \\ \mathsf{let}\,d \otimes e\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ f & \to & f[d/x, e/y] \\ \mathsf{derelict}(\mathsf{promote}\,\vec{e}\ \mathsf{for}\ \vec{x}\ \mathsf{in}\ f) & \to & f[\vec{e}/\vec{x}] \\ \mathsf{discard}(\mathsf{promote}\,\vec{e}\ \mathsf{for}\ \vec{x}\ \mathsf{in}\ f)\ \mathsf{in}\ g & \to & \mathsf{discard}\,\vec{e}\ \mathsf{in}\ g \\ \mathsf{copy}(\mathsf{promote}\,\vec{e}\ \mathsf{for}\ \vec{x}\ \mathsf{in}\ f)\ \mathsf{as}\ y, z\ \mathsf{in}\ g & \to & \mathsf{copy}\,\vec{e}\ \mathsf{as}\ \vec{v}, \vec{w}\ \mathsf{in}\ g[c/y, d/z]. \end{array}$$

In the last rule, we put c to be $(\mathsf{promote}\,\vec{v}\ \mathsf{for}\ \vec{x}\ \mathsf{in}\ f)$ and d to be $(\mathsf{promote}\,\vec{w}$ for

Table 2.2 *Term assignment for linear natural deduction*

$$\frac{}{x:A \vdash x:A}$$

$$\frac{\Gamma,\ x:A \vdash e:B}{\Gamma \vdash (\lambda x^{A}.e):(A \multimap B)} \qquad \frac{\Gamma \vdash e:(A \multimap B) \qquad \Delta \vdash f:A}{\Gamma, \Delta \vdash (\mathsf{apply}\ e\ \mathsf{to}\ f):B}$$

$$\frac{}{\vdash *:I} \qquad \frac{\Gamma \vdash e:I \qquad \Delta \vdash f:A}{\Gamma, \Delta \vdash (\mathsf{let}\ e\ \mathsf{be}\ *\ \mathsf{in}\ f):A}$$

$$\frac{\Gamma \vdash e:A \qquad \Delta \vdash f:B}{\Gamma, \Delta \vdash (e \otimes f):(A \otimes B)} \qquad \frac{\Gamma \vdash e:(A \otimes B) \qquad \Delta,\ x:A,\ y:B \vdash f:C}{\Gamma, \Delta \vdash (\mathsf{let}\ e\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ f):C}$$

$$\frac{\Gamma \vdash e:(!A)}{\Gamma \vdash (\mathsf{derelict}\ e):A} \qquad \frac{\Gamma \vdash e:(!A) \qquad \Delta \vdash f:B}{\Gamma,\ \Delta \vdash (\mathsf{discard}\ e\ \mathsf{in}\ f):B}$$

$$\frac{\Gamma \vdash e:(!A) \qquad \Delta,\ x:(!A),\ y:(!A) \vdash f:B}{\Gamma,\ \Delta \vdash (\mathsf{copy}\ e\ \mathsf{as}\ x, y\ \mathsf{in}\ f):B}$$

$$\frac{\Gamma_i \vdash e_i:(!A_i) \quad (i = 1, \ldots, n) \qquad x_1:(!A_1),\ \ldots,\ x_n:(!A_n) \vdash f:B}{\Gamma_1,\ \ldots,\ \Gamma_n \vdash (\mathsf{promote}\ e_1, \ldots, e_n\ \mathsf{for}\ x_1, \ldots, x_n\ \mathsf{in}\ f):(!B)}$$

$\vec{x}$ in f). Here we use shorthands

$$\begin{aligned}
\mathsf{discard}\ e_1, e_2, \ldots, e_n\ \mathsf{in}\ g &= \mathsf{discard}\ e_1\ \mathsf{in}\ (\mathsf{discard}\ e_2\ \mathsf{in} \cdots (\mathsf{discard}\ e_n\ \mathsf{in}\ g) \cdots) \\
\mathsf{copy}\ e_1, e_2, \ldots, e_n\ \mathsf{as}\ \vec{v}, \vec{w}\ \mathsf{in}\ g &= \mathsf{copy}\ e_1\ \mathsf{as}\ v_1, w_1\ \mathsf{in}\ (\mathsf{copy}\ e_2\ \mathsf{as}\ v_2, w_2\ \mathsf{in} \\
&\qquad \cdots (\mathsf{copy}\ e_n\ \mathsf{as}\ v_n, w_n\ \mathsf{in}\ g) \cdots)
\end{aligned}$$

where in the latter the vector of variables $\vec{v} = v_1, v_2, \ldots, v_n$ and $\vec{w} = w_1, w_2, \ldots, w_n$ are used.

There is a standard translation of simply typed lambda calculus into intuitionistic linear logic. A type $A \Rightarrow B$ of typed lambda calculus is translated into the type $!A \multimap B$ of intuitionistic linear logic. We denote this translation by $A \rightsquigarrow A^*$. Accordingly, we have the translation of typing judgements as

$$\begin{aligned}
&x_1:A_1\ ,\ x_2:A_2\ ,\ \ldots\ ,\ x_n:A_n\ \vdash\ e:A \\
&\qquad\qquad \rightsquigarrow\ x_1:!A_1^*\ ,\ x_2:!A_2^*\ ,\ \ldots\ ,\ x_n:!A_n^*\ \vdash\ e^*:B^*
\end{aligned}$$

for an appropriate translation $e \rightsquigarrow e^*$ of terms (to be precise, definition e^* depends also on the environment).

Normal Functors

Girard introduced normal functors for the purpose of giving models of various systems of lambda calculi [10]. It turns out that normal functors are a special case of analytic functors in the sense of Joyal [19]. Namely the flat species [22] correspond to the normal functors. So they obtain the same concept from entirely different motivations. In [10], analytic functor is used as an alias of normal functor. To avoid confusion, we use the name of normal functors only.

2.3 Definition
Let $\mathbf{C}$ be a category.

A *normal functor* from $\mathbf{C}$ to $\mathbf{Set}$ is a coproduct $\sum_{i \in I} \mathbf{Set}(X_i, \text{-})$ of representable functors where I is a small set and all X_i are finitely presentable objects [1] of $\mathbf{C}$.

Remark: In this paper, we deal with only the case where the category $\mathbf{C}$ is of the form $\mathbf{Set}^A$ for a set A.

There are three equivalent characterizations of normal functors into $\mathbf{Set}$. For one of the characterizations, we need the following definition.

2.4 Definition
A *normal form* of an object A in a category $\mathbf{C}$ is an initial object $X \to A$ in the slice category $\mathbf{C}/A$. If every object has a normal form, the category $\mathbf{C}$ fulfills the *normal form property.*

2.5 Theorem
For a functor $f : \mathbf{Set}^A \to \mathbf{Set}$, *the following three conditions are equivalent:*

(i) f *is a normal functor.*
(ii) f *preserves all filtered colimits and all pullbacks (including infinite ones).*
(iii) *The category* $\mathrm{el}(f)$ *of elements enjoys the normal form property. Moreover,* X *is a finitely presentable object for every normal form* (X, x).

Remark: In [10], it is assumed, by definition, a normal functor preserves equalizers as well. This condition, however, follows from the preservation of filtered colimits and pullbacks.

By tupling normal functors into **Set**, we can define also a normal functor from $\mathbf{Set}^A$ to $\mathbf{Set}^B$ for sets A and B. Namely a normal functor f from $\mathbf{Set}^A$ to $\mathbf{Set}^B$ is a family of normal functors $f_b : \mathbf{Set}^A \to \mathbf{Set}$ where b ranges over the set B.

2.6 Definition (of $\mathbf{CAAcc}_{\mathbf{NF}}$)
The (large) category $\mathbf{CAAcc}_{\mathbf{NF}}$ has categories in the form $\mathbf{Set}^A$ for some set A as objects and all normal functors as morphisms.

Remark: **CAAcc** is an acronym of complete atomic accessible category. The justification of this terminology will be given in the full paper [16].

A *cartesian* natural transformation is a natural transformation $\nu : f \to g$ subject to the condition that the square diagram

$$\begin{tikzcd} fC \arrow[r, "\nu_C"] \arrow[d, "fk"'] & gC \arrow[d, "gk"] \\ fD \arrow[r, "\nu_D"'] & gD \end{tikzcd}$$

is a pullback for every morphism $C \xrightarrow{k} D$. We define $[\mathbf{Set}^A, \mathbf{Set}^B]_{\mathbf{NF}}$ as the category having all normal functors from $\mathbf{Set}^A$ to $\mathbf{Set}^B$ as objects and all cartesian natural transformations as morphisms.

A finitely presentable object of $\mathbf{Set}^A$ is identified with a finite multiset of members of A. If A is a finite set n, a presheaf Z in $\mathbf{Set}^n$ is regarded as a tuple $(Z_0, Z_1, \ldots, Z_{n-1})$ of sets. If a finitely presentable object of $\mathbf{Set}^n$ is given as a multiset γ containing $0, 1, \ldots, n-1$ with multiplicity $m_0, m_1, \ldots, m_{n-1}$ respectively, then the value $\mathbf{Set}^n(\gamma, Z)$ is exactly a monomial ${Z_0}^{m_0} {Z_1}^{m_1} \cdots {Z_{n-1}}^{m_{n-1}}$. For a general set A, in the same way, we obtain monomials of $\operatorname{card} A$ variables. So, as a sum of these monomials, a normal functor from $\mathbf{Set}^A$ to **Set** is a formal power series in $\operatorname{card} A$ variables.

2.7 Theorem
Category $\mathbf{CAAcc}_{\mathbf{NF}}$ *is cartesian closed by equivalence* $[\mathbf{Set}^{A+B}, \mathbf{Set}^C]_{\mathbf{NF}} \cong [\mathbf{Set}^A, \mathbf{Set}^{\exp B \times C}]_{\mathbf{NF}}$ *where* $\exp B$ *denotes the set of all finite multisets of members of* B.

2.8 Proposition
Let $f : \mathbf{Set}^A \times \mathbf{Set}^B \to \mathbf{Set}^B$ *be a binary normal functor, that is, a normal functor on* $\mathbf{Set}^{A+B}$

There is a normal functor $\mu f : \mathbf{Set}^A \to \mathbf{Set}^B$ *where* $(\mu f)(v)$ *is the object part of an initial algebra of the endofunctor* $f(v, \text{-})$ *for each presheaf* $v \in \mathbf{Set}^A$.

We may write the object part of an initial algebra $\mu x. f(x)$ in place of μf.

3 Fixpoint, Trace, and Inversion

Normal Functor Model

We give a model of PCF in the category $\mathbf{CAAcc}_{\mathrm{NF}}$. Types are interpreted as sets by the following definition:

$$\begin{array}{lcl} [\![\iota]\!] & = & \omega \\ [\![o]\!] & = & 2 \\ [\![\sigma \Rightarrow \tau]\!] & = & \exp[\![\sigma]\!] \times [\![\tau]\!] \end{array}$$

where ω is the set of natural numbers and 2 is the set $\{0, 1\}$.

For the interpretation of a term, we define it as a function of the pairs of environments $\Gamma = x_1 : \sigma_1,\ x_2 : \sigma_2,\ \ldots,\ x_n : \sigma_n$ and a term $M : \tau$ such that $\Gamma \vdash M : \tau$ is a correct typing judgement. The interpretation $[\![M]\!]_{\vec{x}}$ is a normal functor from $\mathbf{Set}^{A_1+A_2+\cdots+A_n}$ to $\mathbf{Set}^B$ where $A_i = [\![\sigma_i]\!]$ and $B = [\![\tau]\!]$. The definition of $[\![M]\!]_{\vec{x}}$ is by induction on construction of terms. What is the most interesting is the interpretation of the fixpoint combinator. But we start with easy ones.

The interpretation of the fragment of ordinary typed lambda calculus is induced from the structure of $\mathbf{CAAcc}_{\mathrm{NF}}$ as a cartesian closed category. The numerals of type ι are interpreted as the singletons of the corresponding numerals in ω. Namely, $[\![n]\!]$ for each numeral n is the singleton multiset $\{n\}$, which means the presheaf in $\mathbf{Set}^\omega$ carrying n to 1 and all other members of ω to $\emptyset$. The operation succ is interpreted as presheaves in $\mathbf{Set}^{\exp\omega\times\omega}$ taking the value 1 for $[\{n\}, n+1]$ and taking the value $\emptyset$ for all other members. In other words, $[\![\mathrm{succ}]\!]$ is the normal functor from $\mathbf{Set}^\omega$ to $\mathbf{Set}^\omega$ (so ω-indexed family of formal power series) satisfying that the n-th component is simply monomial x^{n+1}. Likewise we can define the interpretation of all Boolean and arithmetic operators.

What remains is the interpretation of the fixpoint combinator. It is given by initial algebra construction. Applying Prop. ??? to the evaluation, for each set X, we have a normal functor μ from $\mathbf{Set}^{\exp X\times X}$ into $\mathbf{Set}^X$ carrying each normal functor $f : \mathbf{Set}^X \to \mathbf{Set}^X$ to the object part of an initial algebra μf.

3.1 Theorem
The model of PCF *in category* $\mathbf{CAAcc}_{\mathrm{NF}}$ *is sound. Namely, if* $M \to N$, *then* $[\![M]\!]_{\vec{x}} = [\![N]\!]_{\vec{x}}$ *as normal functors.*

Trace

Categorical trace is introduced by Joyal, Street, and Verity as a generalization of several concepts including the usual trace in linear algebra [21]. It is given as an operation satisfying natural axioms of trace in a balanced monoidal categories [20]. We need only symmetric monoidal categories and, indeed, our main interest is in the case that the monoidal structure is given by the cartesian product.

3.2 Definition
A *traced monoidal category* is a symmetric monoidal category endowed with the family of operations

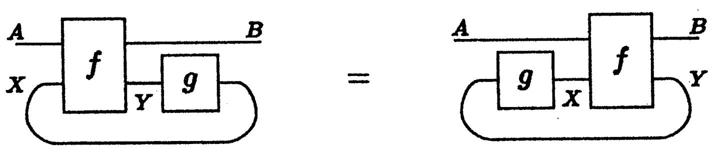
$$\frac{A \otimes X \xrightarrow{f} B \otimes X}{A \xrightarrow{\mathrm{tr}^X f} B}$$

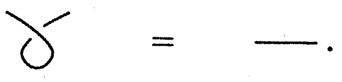
subject to the following conditions:

(vanishing)	$\mathrm{tr}^I f = f$	for $A \otimes I \xrightarrow{f} B \otimes I$
	$\mathrm{tr}^X(\mathrm{tr}^Y f) = \mathrm{tr}^{X \otimes Y} f$	for $A \otimes X \otimes Y \xrightarrow{f} B \otimes X \otimes Y$
(superposing)	$\mathrm{tr}^X(1 \otimes f) = 1 \otimes \mathrm{tr}^X f$	for $A \otimes X \xrightarrow{f} B \otimes X$
(yanking)	$\mathrm{tr}^X c = 1$	for $X \otimes X \xrightarrow{c} X \otimes X$
(left-tightening)	$\mathrm{tr}^X(f(g \otimes 1)) = (\mathrm{tr}^X f)g$	for $A \otimes X \xrightarrow{f} B \otimes X$ and $A' \xrightarrow{g} A$
(right-tightening)	$\mathrm{tr}^X((g \otimes 1)f) = g(\mathrm{tr}^X f)$	for $A \otimes X \xrightarrow{f} B \otimes X$ and $B \xrightarrow{g} B'$
(sliding)	$\mathrm{tr}^X((1 \otimes g)f) = \mathrm{tr}^Y(f(1 \otimes g))$	for $A \otimes X \xrightarrow{f} B \otimes Y$ and $Y \xrightarrow{g} X$

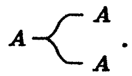
where $c = c_{X,X}$ is the symmetry. We omitted canonical isomorphisms, which should be clear from the context. For instance, the right hand side of the first rule of vanishing should be $\rho_B^{-1} \circ f \circ \rho_A$ with canonical isomorphisms $\rho_A : A \otimes I \to A$ and ρ_B.

The axioms of traced monoidal category are simulated by graphs. A morphism is drawn as a directed graph where the vertices are labeled by primitive morphisms and the edges are labeled by objects. For example, the sliding rule and the yanking rule amount to the following equalities between graphs:





We consider the traced monoidal categories where the monoidal structures are cartesian products. For cartesian product, there are diagonal maps $\Delta_A : A \to A \times A$ and unique morphisms $!_A : A \to 1$ to the terminal object. We denote Δ_A diagrammatically by



3.3 Definition

A *fixpoint operator* in a cartesian category is an operation $(\cdot)^\dagger$

$$\frac{A \times X \xrightarrow{f} X}{A \xrightarrow{f^\dagger} X}$$

of morphisms, natural in A and dinatural in X, this operation subject to the condition that $f^\dagger$ is equal to the composite $A \xrightarrow{\Delta_A} A \times A \xrightarrow{1 \times f^\dagger} A \times X \xrightarrow{f} X$

3.4 Definition

We consider a cartesian category with fixpoint operator $(\cdot)^\dagger$. Given two morphisms $A \times X \times Y \xrightarrow{f} X$ and $A \times X \times Y \xrightarrow{g} Y$, we put $h : A \times X \to X$ as $A \times X \xrightarrow{\Delta_{A \times X}} A \times X \times A \times X \xrightarrow{1_{A \times X} \times g^\dagger} A \times X \times Y \xrightarrow{f} X$ where Δ is the diagonal.

Bekič's formula is the equality $\langle f, g \rangle^\dagger = \langle h^\dagger, g^\dagger \langle 1_A, h^\dagger \rangle \rangle$ between morphisms from A to $X \times Y$.

Remark: It would be easier to understand, if we informally write h as $h(a, x) = f(a, x, g^\dagger(a, x))$ where a and x are parameters from A and X.

M. Hasegawa proved that, in a cartesian category, giving a trace is equivalent to giving a fixpoint operator satisfying Bekič's formula [13, 14].

3.5 Theorem

Let **C** *be a cartesian category.*

The category **C** *is a traced cartesian category iff* **C** *has a fixpoint operator satisfying Bekič's formula.*

A model of intuitionistic linear logic is given by the category $\mathbf{C}$ fulfilling the following structures: The multiplicative fragment $\otimes, I$ and $\multimap$ are interpreted by a symmetric monoidal closed category [23]. The exponential ! is interpreted as a symmetric monoidal functor [8], which is given as a triple $(!, \tilde{\varphi}, \varphi_0)$ where $! : \mathbf{C} \to \mathbf{C}$ is a functor, $!A \otimes !B \xrightarrow{\tilde{\varphi}_{A,B}} !(A \otimes B)$ is a natural transformation, and $I \xrightarrow{\varphi_0} !I$ is a morphism. To interpret discard and copy, we assume that each object of the form $!A$ is endowed with a commutative comonoid structure $(!A, e_A, d_A)$ where $!A \xrightarrow{e_A} I$ and $!A \xrightarrow{d_A} !A \otimes !A$ are monoidal natural transformations. To interpret derelict and promote, we assume that the functor ! takes part of the comonad $(!, \epsilon, \delta)$ where $!A \xrightarrow{\epsilon_A} A$ and $!A \xrightarrow{\delta_A} !!A$ are monoidal natural transformations. Moreover, we need several coherence conditions to make this model sound. See, for example, [4].

We modify the model of PCF in the category $\mathbf{CAAcc_{NF}}$ to construct a model of intuitionistic linear logic. In this model, morphisms should interpret linear terms. So we need the following definition:

3.6 Definition
A *linear normal functor* from $\mathbf{Set}^A$ to $\mathbf{Set}^B$ is a functor preserving all pullbacks and all colimits. The category $\mathbf{CAAcc_{LNF}}$ of complete atomic accessible categories and linear normal functors is induced as a subcategory of $\mathbf{CAAcc_{NF}}$.

Alternatively, linear normal functors are those normal functors $\mathbf{Set}^A \xrightarrow{f} \mathbf{Set}^B$ where, for every normal form (X, a) in $\mathrm{el}(f_b)$ for a member $b \in B$, the underlying finitely presentable object $X \in \mathbf{Set}^A$ corresponds to a singleton in $\exp A$. Hence, if we denote by $[\mathbf{Set}^A, \mathbf{Set}^B]_{\mathbf{LNF}}$ the category of linear normal functors and cartesian natural transformations, we have the categorical equivalence $[\mathbf{Set}^A, \mathbf{Set}^B]_{\mathbf{LNF}} \cong \mathbf{Set}^{A \times B}$.

A presheaf in $\mathbf{Set}^{A \times B}$ is regarded as a matrix with the columns indexed by the members of A and the rows indexed by the members of B such that each entry is a set. If we have two matrices $M \in \mathbf{Set}^{A \times B}$ and $N \in \mathbf{Set}^{B \times C}$, the composite of the corresponding linear normal functors is represented by multiplication of matrices NM where the entry of index $(a, c) \in A \times C$ is the set $\sum_{b \in B} N[b, c]M[a, b]$. Here coproduct in $\mathbf{Set}^{A \times C}$ is denoted by $\sum$ and cartesian product by concatenation.

We show that the category $\mathbf{CAAcc_{LNF}}$ forms a model of intuitionistic linear logic. First we define the symmetric monoidal closed structure. Tensor of $\mathbf{Set}^A$ and $\mathbf{Set}^B$ is given by $\mathbf{Set}^{A \times B}$, and unit I by $\mathbf{Set}^1$ where 1 is a singleton. The right adjoint of tensor is given also by product as $\mathbf{Set}^A \multimap \mathbf{Set}^B = \mathbf{Set}^{A \times B}$.

The monoidal functor to interpret the exponential ! is defined as follows: On objects, $!A$ is the set $\exp A$ of all finite multisets of members of A. On morphisms, we define the following operation associating the matrix $\wp M \in \mathbf{Set}^{\exp A \times \exp B}$ to a matrix $M \in \mathbf{Set}^{A\times B}$. First we note the categorical equivalence $[\mathbf{Set}^A, \mathbf{Set}]_{\mathbf{NF}} \cong \mathbf{Set}^{\exp A}$. The mapping $g \mapsto g \circ {}^tM$ defines a functor from $[\mathbf{Set}^A, \mathbf{Set}]_{\mathbf{NF}}$ to $[\mathbf{Set}^B, \mathbf{Set}]_{\mathbf{NF}}$ where ${}^tM$ is the usual transpose of the matrix M. This functor is linear, so it determines a linear normal functor $\wp M$ from $\mathbf{Set}^{\exp A}$ to $\mathbf{Set}^{\exp B}$. By definition, it is obvious that $\wp$ is functorial, preserving identities and composition. This construction $\wp M$ appears in the tensor representation in a polynomial ring [26].

The natural transformations involved in linear category are given as the inverse image f^* of appropriate function f. The inverse image $f^* : \mathbf{Set}^B \to \mathbf{Set}^A$ is a linear normal functor, and its matrix $M \in \mathbf{Set}^{B\times A}$ satisfies the condition that $M[b, a]$ equals 1 if $f(a) = b$; otherwise equals 0. For instance, the morphism $\tilde{\varphi}$: $!A \otimes !B \multimap !(A \otimes B)$ is the functor f^* in $[\mathbf{Set}^{\exp A\times \exp B}, \mathbf{Set}^{\exp(A\times B)}]_{\mathbf{LNF}}$ corresponding to the function $\exp(A\times B) \xrightarrow{f} \exp A \times \exp B$ carrying $\{\langle a_1, b_1\rangle, \langle a_2, b_2\rangle, \ldots, \langle a_n, b_n\rangle\}$ to the pair of $\{a_1, a_2, \ldots, a_n\}$ and $\{b_1, b_2, \ldots, b_n\}$.

Trace of Normal Functors

As observed above, the model of intuitionistic linear logic has similarity to linear algebras, although the entries of matrices are sets rather than numbers. So we can define the trace of a linear normal functor $\mathbf{Set}^A \xrightarrow{f} \mathbf{Set}^A$ by the diagonal sum $\sum_{a\in A} M[a, a]$ where M is the matrix in $\mathbf{Set}^{A\times A}$ associated to f. With this definition, the category $\mathbf{CAAcc}_{\mathbf{LNF}}$ of linear normal functors turns out to be a traced monoidal category.

However, what we want to have is the trace in the category $\mathbf{CAAcc}_{\mathbf{NF}}$ of normal functors. A normal functor $\mathbf{Set}^A \to \mathbf{Set}^A$ corresponds to a matrix in $\mathbf{Set}^{\exp A\times A}$. We cannot take the diagonal sum, since this is not a square matrix. In the following, we show how to modify the trace of linear normal functors to the trace of normal functors.

A straightforward idea is the following. If we have a normal functor $A \xrightarrow{f} A$, that is, a linear normal functor $!A \overset{f}{\multimap} A$, we have the promotion $!A \overset{pf}{\multimap} !A$, corresponding to a square matrix in $\mathbf{Set}^{\exp A\times \exp A}$. Hence we can take the diagonal sum.

More generally, if a normal functor in $[\mathbf{Set}^{A+X}, \mathbf{Set}^{B+X}]_{\mathbf{NF}}$ is given, we may write it a pair of $!A \otimes !X \overset{h}{\multimap} B$ and $!A \otimes !X \overset{f}{\multimap} X$, employing the terminology of intuitionistic linear logic. Promoting the latter, we have $pf : !A \otimes !X \multimap !X$. Hence $h \otimes pf$ preceded by canonical morphism from $!A \otimes !X$ to $!A \otimes !X \otimes !A \otimes !X$ yields a linear map from $!A \otimes !X$ to $B \otimes !X$. We define $\sigma_h(f)$ as the diagonal sum of this linear map with respect to $!X$. In the formal power series notation, this amounts to the following definition.

3.7 Definition (of $\sigma_h(f)$)
Let $\mathbf{Set}^{A+X} \xrightarrow{f} \mathbf{Set}^X$ and $\mathbf{Set}^{A+X} \xrightarrow{h} \mathbf{Set}^B$ be normal functors.

The normal functor $\sigma_h(f)$ from $\mathbf{Set}^A$ to $\mathbf{Set}^B$ is defined by $\sum [x^\gamma] h(a,x) f(a,x)^\gamma$ where the summation is over all $\gamma \in \exp X$.

Unfortunately, this $\sigma_h(f)$ does not satisfy the axioms of traced monoidal categories. So we "normalize" it as in the following definition. We verify that, with this $\tau_h(f)$, the category $\mathbf{CAAcc}_{\mathbf{NF}}$ turns out to be a traced cartesian category.

3.8 Definition (of $\tau_h(f)$)
Let $\mathbf{Set}^{A+X} \xrightarrow{h} \mathbf{Set}^B$ and $\mathbf{Set}^{A+X} \xrightarrow{f} \mathbf{Set}^X$ be normal functors.

The formal power series $\tau_h(f)(a)$ is defined by $\sigma_h(f)(a)/\sigma_1(f)(a)$.

Let $R = \mathbb{Z}[f]$ be the ring of all polynomials over integers where the indeterminates are all coefficients of f. If g is of the form $z \cdot f(a,x)$ with $z \in \mathbf{Set}^X$, the denominator

$$\sigma_1(g)(a,z) \;=\; \sum_{\gamma \in \exp X} z^\gamma [x^\gamma] f(a,x)^\gamma$$

of $\tau_h(g)(a,z)$ is of the shape $1 + P(a,z)$ where $P(a,z)$ is a formal power series in the ring $R[[a,z]]$ with no constant term. Noticing the formal power series of this form is invertible for multiplication, the expression $\tau_h(g)(a,z)$ makes sense as an element of the ring $R[[a,z]]$ (supposed the coefficients of $h(a,x)$ are finite). We postpone the verification that $\tau_h(g)(a)$ is meaningful for all g. This will be proved by observing that only the polynomials of non-negative coefficients in R are involved. For the moment, we deal with only the case where g is of the form $zf(a,x)$.

Notation: We write $\tau x.\, f(a,x) = \tau_h(f)(a,x)$ in case that $\mathbf{Set}^{A+X} \xrightarrow{h} \mathbf{Set}^A$ is the projection. This operator τ binds the variable succeeding, so the α-convertible expressions are identified.

3.9 Lemma
Let $\mathbf{Set}^{A+X} \xrightarrow{h} \mathbf{Set}^B$ *and* $\mathbf{Set}^{A+X} \xrightarrow{f} \mathbf{Set}^X$ *be normal functors.*

The equality $\tau_h(f)(a,x) = h(a,\, \tau x.\, f(a,x))$ *holds.*

Employing the new notation, the axiom of traced monoidal category translates into the following equations: First of all, tightening and superposing are direct consequences of Lemma 3.10. The rest turns out to be

$$
\begin{array}{ll}
\text{(vanishing)} & \tau(x,y).\,(f(x,y),F(x,y)) \\
& \quad = (\tau x.\,f'(x),\ \tau y.\,F(\tau x.\,f'(x),\,y)) \\
& \qquad\qquad\qquad \text{where } f'(x) \triangleq f(x,\ \tau y.\,F(x,y)) \\
\text{(sliding)} & g(\tau y.\,f(g(y))) \ = \ \tau x.\,g(f(x)) \\
\text{(yanking)} & \tau y.\,x \ = \ x\,.
\end{array}
$$

3.10 Theorem
Let $f : \mathbf{Set}^{A+X} \to \mathbf{Set}^{X}$ be a normal functor.

$\tau x.\,f(a,x)$ coincides the initial algebra $\mu x.\,f(a,x)$. In particular, $\tau x.\,f(a,x)$ is a normal functor.

As a consequence of this theorem, we see that $\tau_h(g)$ is well-defined for all normal functors h and g.

3.11 Theorem
The category $\mathbf{CAAcc}_{\mathbf{NF}}$ is a traced cartesian category where the trace is given by $\tau_h(f)$.

Lagrange-Good Inversion

As explained in the second section, the Lagrange-Good inversion is the formula to compute the fixed point $x = a(z)$ of the operation of the form $zg(x)$. By Thm. 3.11, the fixpoint $a(z) = f^{\dagger}(z)$ of $f(z,x)$ is given by $\tau x.\,f(z,x)$. By computing $h(f^{\dagger}(z)) = \tau_h(f)(z)$ for $f(z,x)$ defined by $zg(x)$, we have the formula

$$
h(f^{\dagger}(z)) \ = \ \frac{\sum_{\gamma} z^{\gamma}[x^{\gamma}]h(x)g(x)^{\gamma}}{\sum_{\gamma} z^{\gamma}[x^{\gamma}]g(x)^{\gamma}}.
$$

This formula can be regarded as an alternative form of the Lagrange-Good inversion.

Applying Jacobi's residue formula, we can verify this formula is equal to the standard Lagrange-Good inversion formula. We recall Jacobi's residue formula. Let $F_1, F_2, \ldots, F_n$ be formal Laurent series in n variables of the shape $F_i(x_1, x_2, \ldots, x_n) = a_i {x_1}^{b_{i1}} {x_2}^{b_{i2}} \cdots {x_n}^{b_{in}} +$ (higher degree terms). Then, for an arbitrary Laurent series $h(x_1, x_2, \ldots, x_n)$, the formula

$$
\det(b_{ij})\,\mathrm{Res}\,h(x) \ = \ \mathrm{Res}\left(h(F(x))\frac{\partial(F_1,F_2,\ldots,F_n)}{\partial(x_1,x_2,\ldots,x_n)}\right)
$$

holds, where the residue $\mathrm{Res}(f(x_1,x_2,\ldots,x_n))$ is defined as the coefficient of $(x_1 x_2 \cdots x^n)^{-1}$ in Laurent series f. Following [9], the equality $\sum_{\gamma} z^{\gamma}[x^{\gamma}]g(x)^{\gamma} =$

$\det(E - M(z, f^\dagger(z)))^{-1}$ is derived from Jacobi's residue formula. See the full paper for the detail. Therefore the standard Lagrange-Good inversion formula

$$\frac{h(f^\dagger(z))}{\det(E - M(z, f^\dagger(z)))} = \sum_\gamma z^\gamma [x^\gamma] h(x) g(x)^\gamma.$$

is derived.

References

[1] J. Adámek and J. Rosický, *Locally Presentable and Accessible Categories*, London Math. Soc. Lecture Note Ser. 189, (Cambridge University Press, 1994).

[2] H. P. Barendregt, *The Lambda Calculus, Its Syntax and Semantics*, Revised Edition, Studies in Logic and the Foundations of Mathematics, Volume 103, (North-Holland, 1984).

[3] N. Benton, G. Bierman, V. de Paiva and M. Hyland, A term calculus for intuitionistic linear logic, in: *Typed Lambda Calculi and Applications*, M. Bezem, J. F. Groote, eds., 1993, Utrecht, The Netherlands, Lecture Notes in Computer Science 664 (Springer, 1993) pp. 75–90.

[4] G. M. Bierman, What is a categorical model of intuitionistic linear logic?, in: *Typed Lambda Calculi and Applications*, M. Dezani-Ciancaglini, G. Plotkin, eds., Edinburgh, U.K., 1995, Lecture Notes in Computer Science 902, (Springer, 1995) pp. 78–93.

[5] C. C.-A. Cheng, J. H. McKay, J. Towber, S. S.-S. Wang, and D. L. Wright, Reversion of power series and the extended Raney coefficients, *Transactions American Math. Soc.* **349** (1997) 1769–1782.

[6] N. G. de Bruijn, The Lagrange-Good inversion formula and its applications to integral equations, *J. Math. Analysis Applications* **92** (1983) 397–409.

[7] R. Ehrenborg and M. Méndez, A bijective proof of infinite variated Good's inversion, *Advances Math.* **103** (1994) 221–259.

[8] S. Eilenberg and G. M. Kelly, Closed categories, in: *Proc. Conference on Categorical Algebra*, La Jolla, USA, 1965 (Springer, 1966) pp. 421–562.

[9] I. M. Gessel, A combinatorial proof of the multivariate Lagrange inversion formula, *J. Combinatorial Th., Ser. A* **45** (1987) 178–195.

[10] J.-Y. Girard, Normal functors, power series and λ-calculus, *Ann. Pure Applied Logic* **37** (1988) 129–177.

[11] I. J. Good, Generalization to several variables of Lagrange's expansion, with applications to stochastic processes, *Proc. Cambridge Philosophical Soc.* **56** (1960) 367–380.

[12] C. A. Gunter, *Semantics of Programming Languages. Structures and Techniques*, Foundations of Computing Series, (MIT Press, 1992).

[13] M. Hasegawa, Recursion from cyclic sharing: Traced monoidal categories and models of cyclic lambda calculi, in: *Proc. 3rd International Conference on Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science 1210 (Springer, 1997) pp. 196–213.

[14] M. Hasegawa, *Models of Sharing Graphs: A Categorical Semantics of* let *and* letrec, PhD Thesis, The University of Edinburgh, 1997.

[15] R. Hasegawa, The generating function of lambda terms, to appear; a preliminary version appeared in *Combinatorics, Complexity & Logic*, Proc. DMTCS'96, D. S. Bridges, C. S. Claude, J. Gibbons, S. Reeves, I. H. Witten, eds., Auckland, New Zealand, 1997, (Springer, 1997) pp. 253–263.

[16] R. Hasegawa, Two applications of analytic functors, to appear, 1998.

[17] I-C. Huang, Reversion of power series by residues, *Communications Algebra* **26** (1998) 803–812.

[18] A. Joyal, Une théorie combinatoire des séries formelles, *Advances Math.* **42** (1981) 1–82.

[19] A. Joyal, Foncteurs analytiques et espèces de structures, *Combinatoire Enumérative*, Proceedings, Montreal, Québec, Canada, 1985, G. Labelle, P. Leroux, eds., Lecture Notes in Mathematics 1234, (Springer, 1986) pp. 126–159.

[20] A. Joyal and R. Street, Braided tensor categories, *Advances Math.* **102** (1993) 20–78.

[21] A. Joyal, R. Street, and D. Verity, Traced monoidal categories, *Math. Proc. Cambridge Philosophical Soc.* **119** (1996) 447–468.

[22] G. Labelle, On asymmetric structures, *Discrete Math.* **99** (1992) 141–164.

[23] S. Mac Lane, *Categories for the Working Mathematician*, (Springer, 1971).

[24] C.-H. L. Ong, Correspondence between operational and denotational semantics: the full abstraction problem for PCF, in: *Handbook of Logic in Computer Science, Vol 4*, (Oxford University Press, 1995) pp. 269–356.

[25] G. D. Plotkin, LCF considered as a programming language. *Theoretical Computer Sci.* **5** (1977/78) 223–255.

[26] I. Satake, *Linear Algebra* (in Japanese), Suugaku Sensho, Vol 1, (Shoukabou, 1958).

[27] D. S. Scott, Continuous Lattices, in: *Toposes, Algebraic Geometry and Logic*, Halifax, Canada, 1971, Lecture Notes in Mathematics 274, (Springer, 1972) pp. 97–136.