The Calculus of Functional Differences and Integrals

Bruce MacLennan

Department of Computer Science
The University of Tennessee
Knoxville, TN 37996-1301

CS-89-80 April 1989

Abstract

Functions can often be defined by recursive equations such as these:

$$\begin{aligned} f(x) &= b(x), \quad \text{if } p(x) \\ f[h(x)] &= g_x[f(x)], \quad \text{otherwise.} \end{aligned}$$

The second equation says that if the input to f "varies" by h, then its output "varies" by g_x. We call this a functional difference equation; it is analogous to a numerical difference equation, such as:

$$f(h+x) = g_x + f(x),$$

which says that the output of f varies by g_x when its input is varied by h. Since in our theory we vary the input by an arbitrary function, rather than a numerical displacement, we can take differences of functions defined on nonnumeric domains, such as lists, trees and sets.

The ultimate goal of this research is to improve our ability to reason about functional programs. Specifically, the theory of functional differences provides a means of going from local descriptions of behavior (difference equations) to global descriptions (the solution to the difference equation), and vice versa.

In this paper we define several different kinds of functional differences and present fundamental results concerning them (existence, uniqueness, etc.). We also present a complementary operation of "functional integration" that can be used to solve functional difference equations.

1 Introduction

The principal goal of this research is to improve our ability to reason about programs. This is hampered by the difficulty of relating the individual statements we write — which specify behavior locally — to the global behavior of the program. How can local descriptions of behavior be related to global descriptions of behavior? In the physical sciences, the local behavior of a system is usually described by *differential equations.* For example,

$$f'(x) = kf(x) \tag{1}$$

The solution of the differential equations gives the global behavior of the system, subject to certain *boundary conditions.* For example,

$$f(x) = ce^{kx}$$

is a solution to Eq. 1. The constant c is arbitrary, i.e., its value is determined by the boundary conditions. In the physical sciences local behavior is often described in terms of the derivative. For example, if g is the derivative of f, then:

$$f(x + dx) = g(x)dx + f(x)$$

This derivative $g(x)$ shows how the value of f varies when its input in varied by an infinitesmal amount dx.

In applying these ideas to programs we face two difficulties. First, state changes in (digital) computers are discrete rather than continuous. This suggests that *finite differences* are better than (infinitesmal) derivatives as the basis for an analogy. Consider:

$$f(h + x) = g_x + f(x)$$

This equation says that g is the (first forward) difference of f with respect to h. The quantity g_x describes how much the value of f (at x) varies when its input is varied by h. It's often written $g_x = \Delta_h f(x)$.

The second problem is that in programming we are often concerned with *nonnumerical functions.* Thus we must be able to reason about *nonnumerical differences.* To solve this problem we generalize numerical displacements by allowing the "changes" to be arbitrary functions:

$$f[h(x)] = g_x[f(x)] \tag{2}$$

In this equation g_x describes how the value of f (at x) varies when its input is varied by h.

We mention briefly some related work. Robert Paige (Paige, 1981; Paige & Koenig, 1982) has described a related technique called "formal differentiation" or "finite differencing" of computable expressions. Although this work is superficially similar to ours, it is in fact quite different in both motivation and approach. Paige's goal is program optimization by a generalization of the reduction of strength technique. In contrast, we are interested in improving program comprehension and derivation by relating local and global behavior. This divergence in goals is reflected in our difference operators. Paige's differential is semantics-preserving, whereas ours reduces a global change to a local change. Also, we define the inverse operation of functional integration, which accumulates local changes into global changes; Paige has no need for the inverse. A further difference is that Paige's operations apply to imperative programs, whereas our work has been done in the context of functional programming.[1] Finally, Paige's notion of differentiability is relative to a particular set of rules (Paige, 1981, Definition 2), whereas ours is quite general (see Theorem 3 below).

More directly relevant is the research of Backus and his colleagues (Backus, 1978, 1981). They have proved a number of "expansion theorems" that are similar to our Theorems 21 and 35. However, since their theoretical framework is quite different from ours, we've not yet been able to establish precise relationships between the two approaches.

We have previously investigated (1986, 1987) a theory of *fixed* functional differences, that is, a theory in which g_x in Eq. 2 does not depend on x. The theory of fixed differences is subsumed in the present theory (see Section 3.4). It is interesting that the general theory of differences is in many ways *simpler* than the more restricted theory.

2 Differences

2.1 Definitions

In this section we define functional difference operators that correspond to a number of patterns of recursive function definition. First, however, we need some preliminary notation and definitions. We will be considering the difference of $f : S \longrightarrow T$.[2] The difference equation may hold on the entire set S, or it may hold only on a subset S', in which case the boundary conditions specify f on the remainder $S_0 = S - S'$. To allow both of these cases to be expressed by a single equation we introduce the following notation:

Notation (Standard Injections): Suppose $S = S_0 + S'$. We use two injections, $i : S_0 \longrightarrow S$ and $j : S' \longrightarrow S$. If $S \neq S'$, then we take $i : S_0 \longrightarrow S$ and $j : S' \longrightarrow S$ to be the trivial injections $i(x) = x$ and $j(x) = x$. If $S = S'$ then we take $j = \mathbf{I}$, the identity on S, and i to be the degenerate function $i : \emptyset \longrightarrow S$.

[1] Although it can be applied to imperative programs by operating on the state transition functions.

[2] As is customary in algebra, we interpret $f : S \longrightarrow T$ to mean that f is a *total* function from S to T. Partial functions must be handled by circumlocutions (typically by completing the function in some way).

Notation (Embedding Injections): The notation $i : S \xrightarrow{\mathrm{I}} T$ means that $i : S \longrightarrow T$ is the injection $i(x) = x$. Thus, the definition of the standard injections can be expressed $i : S_0 \xrightarrow{\mathrm{I}} S$ and $j : S' \xrightarrow{\mathrm{I}} S$.

The easiest to understand functional differences are the *curried* differences, because they correspond most closely with numerical differences. For this reason we define the curried differences first. However, the *uncurried* differences seem more amenable to algebraic manipulation, so we will use these throughout most of this paper. In any case, it is easy to convert between the curried and uncurried forms.

Notation (Implicit Composition): When confusion is unlikely we represent composition by juxtaposition. Thus, $fg = f \circ g$. When this convention is being used, we will often use subscripting for function application. When subscripting is inconvenient, we use Backus's colon operator for application. Thus, $f : x = f_x = f(x)$ and

$$fg : x = (fg)_x = (f \circ g)(x) = f[g(x)] = f_{g_x}$$

The well-known S combinator will simplify the equations for the curried differences.

Definition (S Combinator): If[3] $F : S \longrightarrow T \longrightarrow U$ and $g : S \longrightarrow T$ then $\mathsf{S}Fg : S \longrightarrow U$ is defined by the equation:

$$(\mathsf{S}Fg)_x = F_x : g_x, \text{ for all } x \in S$$

Definition (Curried Forward Differences): Suppose $f : S \longrightarrow T$ and $h : S \longrightarrow S'$. Any $G : S \longrightarrow T \longrightarrow T$ satisfying

$$(fjh)_x = G_x : f_x, x \in S \tag{3}$$

is called an *undisplaced curried forward difference* of f with respect to h. Alternately,

$$fjh = \mathsf{S}Gf$$

Any $G : S' \longrightarrow T \longrightarrow T$ satisfying

$$(fjh)_x = (Gh)_x : f_x, x \in S \tag{4}$$

is called a *displaced curried forward difference* of f with respect to h. Alternately,

$$fjh = \mathsf{S}(Gh)f$$

The undisplaced forward difference can be visualized as in Fig. 1. The displaced forward difference can be visualized as in Fig. 2.

Definition (Curried Backward Differences): Suppose $f : S \longrightarrow T$ and $r : S' \longrightarrow S$. Any function $G : S' \longrightarrow T \longrightarrow T$ satisfying

$$(fj)_x = G_x : (fr)_x, x \in S' \tag{5}$$

[3]We take the arrow to be right associative, thus $S \longrightarrow T \longrightarrow U = S \longrightarrow (T \longrightarrow U)$.

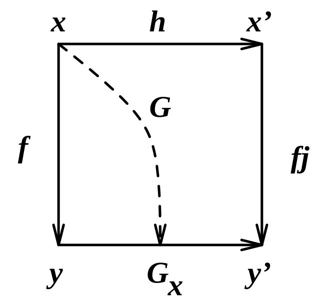

Figure 1: Undisplaced Forward Difference

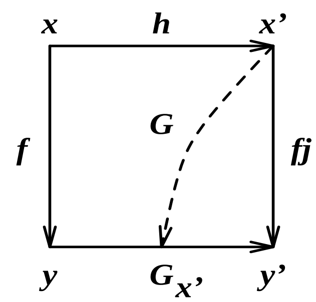

Figure 2: Displaced Forward Difference

is called an *undisplaced curried backward difference* of f with respect to r. Alternately,

$$fj = \mathsf{S}G(fr) \tag{6}$$

Any function $G : S \longrightarrow T \longrightarrow T$ satisfying

$$(fj)_x = (Gr)_x : (fr)_x, x \in S' \tag{7}$$

is called a *displaced curried backward difference* of f with respect to r. Alternately,

$$fj = \mathsf{S}(Gr)(fr) = (\mathsf{S}Gf)r$$

The undisplaced backward difference can be visualized as in Fig. 3. The displaced backward difference can be visualized as in Fig. 4.

Note: The foregoing are just the simplest, *first order, ordinary* difference equations. Higher-order and partial equations have not been systematically investigated. Example of partial difference equations are:

$$\begin{aligned} fjh : x : y &= G : x : y : f_x : f_y \\ f_x &= G_x : (fr)_x : (fs)_x \end{aligned}$$

Example of such programs are tree-walks.

Clearly the corresponding displaced and undisplaced differences are closely related. The relationship is specified in the following theorem:

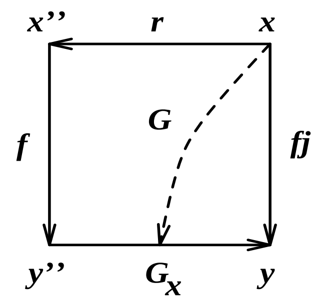

Figure 3: Undisplaced Backward Difference

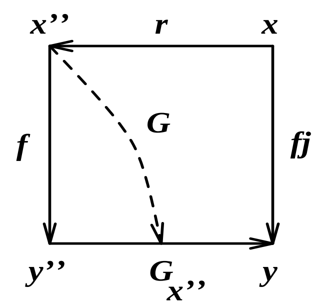

Figure 4: Displaced Backward Difference

Theorem 1 : *Gh is an* undisplaced *curried forward difference of f with respect to h if and only if G is a* displaced *curried forward difference of f with respect to h. Gr is an* undisplaced *curried backward difference of f with respect to r if and only if G is a* displaced *curried backward difference of f with respect to r.*

Proof: By inspection of the difference Eqs. 3, 4, 5 and 2. $\square$

Theorem 1 seems to imply that we dispense with either the undisplaced or the displaced differences, but we will see below (Section 2.2) that there are circumstances in which it is useful to have both.

The higher-order curried differences are difficult to manipulate algebraically. Therefore we introduce corresponding uncurried differences. Expressing the difference equations requires the use of *functional construction.*

Definition (Functional Construction): If $f : R \longrightarrow S$ and $g : R \longrightarrow T$, then $(f;g)$ is the function $(f;g) : R \longrightarrow S \times T$ defined as follows:

$$(f;g)_x = (f_x, g_x)$$

That is, $(f;g)$ returns the ordered pair obtained by applying both f and g to its argument.

Definition (Uncurried Forward Differences): Suppose $f : S \longrightarrow T$ and $h : S \longrightarrow S'$. Any $g : S \times T \longrightarrow T$ satisfying

$$fjh = g(\mathbf{I}; f) \tag{8}$$

is called an *undisplaced (uncurried) forward difference* of f with respect to h. Any $g : S' \times T \longrightarrow T$ satisfying

$$fjh = g(h; f) \tag{9}$$

is called a *displaced (uncurried) forward difference* of f with respect to h.

Definition (Uncurried Backward Differences): Suppose $f : S \longrightarrow T$ and $r : S' \longrightarrow S$. Any function $g : S' \times T \longrightarrow T$ satisfying

$$fj = g(\mathbf{I}; fr) \tag{10}$$

is called an *undisplaced (uncurried) backward difference* of f with respect to r. Any function $g : S \times T \longrightarrow T$ satisfying

$$fj = g(\mathbf{I}; f)r \tag{11}$$

is called a *displaced (uncurried) backward difference* of f with respect to r.

Corresponding to Theorem 1 we have the following Theorem 2 for uncurried differences. First however we need additional notation.

Definition (Projections): The following functions select elements from pairs:

$$\begin{array}{ll} \lambda : \ (x, y) = x & \text{(left)} \\ \rho : \ (x, y) = y & \text{(right)} \end{array}$$

Definition (Functional Direct Product): Suppose $f : Q \longrightarrow R$ and $g : S \longrightarrow T$. Then the functional direct product $f \times g$ is the function $f \times g : Q \times S \longrightarrow R \times T$ defined as follows:

$$f \times g = (f\lambda; g\rho)$$

Hence, $(f \times g)_{(x,y)} = (f_x, g_y)$.

We note without proof that:

$$\begin{aligned} \lambda(f; g) &= f, \\ \rho(f; g) &= g, \\ (f; g)h &= (fh; gh), \\ \lambda(f \times g) &= f\lambda, \\ \rho(f \times g) &= g\rho, \\ (f \times g)(p; q) &= (fp; gq), \text{ and} \\ (f \times g)(h \times k) &= (fh \times gk). \end{aligned}$$

Theorem 2 : *First, $g(h \times \mathbf{I})$ is an undisplaced forward difference of f with respect to h if and only if g is a displaced forward difference of f with respect to h. Second, $g(r \times \mathbf{I})$ is an undisplaced backward difference of f with respect to r if and only if g is a displaced backward difference of f with respect to r.*

Proof: Observe:

$$\begin{aligned} fjh &= g(h;f) = g(h\times\mathbf{I})(\mathbf{I};f) \\ fj &= g(r;fr) = g(r\times\mathbf{I})(\mathbf{I};fr) \end{aligned}$$

□

2.2 Existence and Computability

In this section we establish the conditions for the existence and computability of functional differences.

Lemma 1 : *The function $g = fj\lambda$ is a solution to the displaced forward difference equation $fjh = g(h;f)$. The function $g = fj\lambda$ is a solution to the undisplaced backward difference equation $fj = g(\mathbf{I};fr)$. The function $g = fjh\lambda$ is a solution to the undisplaced forward difference equation $fjh = g(\mathbf{I};f)$.*

Proof: Observe: $(fj\lambda)(h;f) = fj\lambda(h;f) = fjh$. The other results are similar. □

Theorem 3 (Existence of Differences): *The displaced forward, undisplaced backward and undisplaced forward differences always exist.*

Lemma 2 : *If $qr = \mathbf{I}$ then $g = fjq\lambda$ is a solution to the displaced backward difference equation $fj = g(\mathbf{I};f)r$.*

Proof: Observe: $(fjq\lambda)(\mathbf{I};f)r = fjq\lambda(\mathbf{I};f)r = fjqr = fj$. □

Theorem 4 (Existence of Displaced Backward Difference): *If r is injective, then the displaced backward difference of f with respect to r exists.*[4]

Note that there is not the apparent symmetry among the four differences; the displaced backward difference loses information in a way that the displaced forward difference does not.

Theorem 5 (Computability of Differences): *In the preceding two theorems, the given differences are computable provided f, h and q are computable.*

Proof: This follows directly from the form of the solutions $fj\lambda$, $fjh\lambda$ and $fjq\lambda$. □

The preceding theorems do not give interesting formulas for the differences, since they are defined in terms of f and the displacement function (h or r). However these theorems do establish the conditions under which the differences exist and under which they are computable. With this information in hand we can attempt to find formulas for the differences that do not involve f (see Section 2.4).

[4]The converse does not hold. For a counterexample, consider the displaced backward difference of 'length' with respect to 'rest' (`cdr`).

2.3 Uniqueness and General Forms

In general, functional differences are not unique. Consider the undisplaced forward difference equation $fjh = g(\mathbf{I}; f)$. The value of g does not matter on any pair (x, y) not in the image of $(\mathbf{I}; f)$, that is, any pair (x, y) for which $y \neq f_x$. Thus there may be many gs that satisfy the equation, but that vary in their values on the noncritical elements of $S \times T$. More carefully, since $(\mathbf{I}; f)$ is injective, it has one or more left inverses. It is easy to see that for each left inverse ψ of this function, $\psi(\mathbf{I}; f) = \mathbf{I}$, the function $g = fjh\psi$ is a solution of the difference equation:

$$g(\mathbf{I}; f) = fjh\psi(\mathbf{I}; f) = fjh$$

On the other hand, if $(\mathbf{I}; f)$ is also surjective, then it has a two sided inverse and we can solve the difference equation for a unique g:

$$g = g(\mathbf{I}; f)(\mathbf{I}; f)^{-1} = fjh(\mathbf{I}; f)^{-1}$$

Unfortunately, this result is not useful, since it's easy to show that $(\mathbf{I}; f)$ is surjective if and only if $T = \mathbf{1}$. Analogous properties can be seen to hold for the other differences. Since we cannot have unique differences (under nondegenerate circumstances), we must try to characterize the differences as members of a family. Although we do not have a unique *function,* perhaps we can have a unique *form.* In anticipation of this, we introduce notations to represent these forms:

Definition (Uncurried Differences): The general forms for the uncurried differences are represented by δ, $\bar{\delta}$, ∂ or $\bar{\partial}$ according to the following difference equations:

$$\begin{array}{ll} fjh = \delta_h f(\mathbf{I}; f) & \text{(undisplaced forward)} \\ fjh = \bar{\delta}_h f(h; f) & \text{(displaced forward)} \\ fj = \partial_r f(\mathbf{I}; fr) & \text{(undisplaced backward)} \\ fj = \bar{\partial}_r f(\mathbf{I}; f)r & \text{(displaced backward)} \end{array} \tag{12}$$

These symbols ambiguously denote *any* solution of the difference equations in which they appear.[5] A formula such as '$\delta_h f$' is read, "the undisplaced forward difference with respect to h of f."

Definition (Curried Differences): The general forms for the curried differences are represented by Δ, $\bar{\Delta}$, ∇ or $\bar{\nabla}$ according to the following equations:

$$\begin{array}{ll} fjh = \mathsf{S}(\Delta_h f)f & \text{(undisplaced forward)} \\ fjh = \mathsf{S}(\bar{\Delta}_h f \circ h)f & \text{(displaced forward)} \\ fj = \mathsf{S}(\nabla_r f)(fr) & \text{(undisplaced backward)} \\ fj = [\mathsf{S}(\bar{\nabla}_r f)f]r & \text{(displaced backward)} \end{array}$$

Definition (Functional Alternation): Suppose $f : R \longrightarrow T$ and $g : S \longrightarrow T$. Then the functional alternation $f \mid g$ is the function $(f \mid g) : R + S \longrightarrow T$ defined by

[5] This ambiguity is resolved later.

the equations:

$$\begin{array}{l}(f \mid g)_x = f_x, \ \text{ if } \ x \in R \\ (f \mid g)_x = g_x, \ \text{ if } \ x \in S\end{array}$$

We note without proof that $f(g \mid h) = fg \mid fh$.

Theorem 6 (Properties of Direct Sum Injections): *Suppose $i : R \xrightarrow{\mathrm{I}} R+S$ and $j : S \xrightarrow{\mathrm{I}} R+S$. Further suppose $f : R \longrightarrow T$, $g : S \longrightarrow T$ and $h : R+S \longrightarrow T$. Then:*

$$\begin{array}{l} i \mid j = \mathbf{I} : S+T \xrightarrow{\mathrm{I}} S+T \\ (f \mid g)i = f \\ (f \mid g)j = g \\ h = hi \mid hj \end{array} \tag{13}$$

Corollary 1 : *Suppose $f : R \longrightarrow T$, $g : S \longrightarrow T$, $h : R \longrightarrow U$ and $k : S \longrightarrow U$ Then:*

$$(f \mid g \;\; ; \;\; h \mid k) = (f;h) \mid (g;k)$$

Proof:

$$\begin{array}{rcl}(f \mid g;h \mid k) & = & (f \mid g;h \mid k)i \mid (f \mid g;h \mid k)j \\ & = & [(f \mid g)i;(h \mid k)i] \mid [(f \mid g)j;(h \mid k)j] \\ & = & (f;h) \mid (g;k)\end{array}$$

□

Theorem 7 : *Suppose $i : R \xrightarrow{\mathrm{I}} R+S$, $f : R+S \longrightarrow T$, and $g : R+S \longrightarrow T$. Then $fi = gi$ if and only if there is a function $c : S \longrightarrow T$ such that*

$$f = gi \mid c$$

Proof: For the "only if part" let $c = fj$ and observe that by the preceding theorem and $fi = gi$ we have:

$$f = fi \mid fj = gi \mid fj = gi \mid c$$

For the "if part" observe (Eqs. 13) $fi = (gi \mid c)i = gi$. □

Note how $gi \mid c$ can be taken as a general form for functions f in the equivalence class $fi = gi$. (We will shortly introduce the notation $f \doteq g$ for the corresponding equivalence relation.) More accurately, a general form is a family of functions indexed by c, $G_c = gi \mid c$.

Theorem 8 (General Form for Undisplaced Forward Difference): *If g is any solution to $fjh = g(\mathbf{I};f)$, then*

$$\delta_h f = g\iota \mid c \tag{14}$$

and

$$g = (\delta_h f)\iota \mid c \tag{15}$$

where $\iota : \mathrm{Im}(\mathbf{I}; f) \xrightarrow{\mathrm{I}} S \times T$ *and* c *is an arbitrary "remainder" function*

$$c : \overline{\mathrm{Im}}(\mathbf{I}; f) \longrightarrow T.$$

Here $\mathrm{Im}(\mathbf{I}; f)$ *is the image of* $(\mathbf{I}; f)$ *and* $\overline{\mathrm{Im}}(\mathbf{I}; f)$ *is the image's complement:*

$$\overline{\mathrm{Im}}(\mathbf{I}; f) = S \times T - \mathrm{Im}(\mathbf{I}; f)$$

Proof: We are given $fjh = g(\mathbf{I}; f)$. First we must show that any other solution $\delta_h f = g'$ can be written in the form $g' = g\iota \mid c$. Thus we assume $fjh = g'(\mathbf{I}; f)$. Since $(\mathbf{I}; f)$ is injective, let $\iota\phi = (\mathbf{I}; f)$; the bijection $\phi : S \longrightarrow \mathrm{Im}(\mathbf{I}; f)$ is uniquely determined by $(\mathbf{I}; f)$. Then,

$$\begin{aligned} g'\iota &= g'\iota\phi\phi^{-1} = g'(\mathbf{I}; f)\phi^{-1} = fj \\ h\phi^{-1} &= g(\mathbf{I}; f)\phi^{-1} = g\iota\phi\phi^{-1} = g\iota \end{aligned}$$

Hence, by the preceding theorem, $g' = g\iota \mid c$ and $g = g'\iota \mid c'$.

Next we must show that any function $g' = g\iota \mid c$ is a solution to $fjh = g'(\mathbf{I}; f)$. We know by Thm. 7 that $g\iota = g'\iota$, so simply derive:

$$g'(\mathbf{I}; f) = g'\iota\phi = g\iota\phi = g(\mathbf{I}; f) = fjh$$

□

The other differences have similar general forms.

Corollary 2 : *Any undisplaced forward difference can be written in the form*

$$\delta_h f = fjh\lambda\iota \mid c$$

and any function that can be written in this form is an undisplaced forward difference.

We will not be able to get a *useful* general form for the curried differences analogous to that for the uncurried. The problem is that the two arguments of a curried difference cannot be treated as components of a single domain — another example of the less tractable nature of curried functions. Although we will continue to pursue the curried differences for their motivational and intuitive value, we will generally have to use the uncurried differences when algebraic manipulation is the goal.

The following definition and notation simplify dealing with general forms.

Definition (Kernel): The *kernel* of a functional difference is that subset of its domain over which its value matters. The kernels of the differences are listed in Table 1.

Notation (Equal Over a Kernel): We use the notation $f \doteq_K g$ to mean that f and g are equal over the kernel K; that is, that $f\iota = g\iota$ for $\iota : K \xrightarrow{\mathrm{I}} S$. We

Table 1: Kernels of Differences

Difference	Kernel
undisplaced forward	$\mathrm{Im}(\mathbf{I}; f)$
displaced forward	$\mathrm{Im}(h; f)$
undisplaced backward	$\mathrm{Im}(\mathbf{I}; fr)$
displaced backward	$\mathrm{Im}(r; fr)$

write $f \doteq g$ when the relevant kernel is clear from context. For example, $\delta_h f \doteq g$ means that g is a particular solution, and hence $\delta_h f = g\iota \mid c$ and $g = (\delta_h f)\iota \mid c$. By extension, we use $\Delta_h f \doteq G$ to mean that the corresponding uncurried differences agree on their kernel (i.e., $\delta_h f \doteq g$ where g is uncurried G). Thus, although we do not have a useful formula for the general form of a curried difference, we can nevertheless talk about equality of their general forms.

Theorem 9 : *If h is bijective, then $\delta_h f \doteq \bar{\partial}_{h^{-1}} f$.*

Proof: We prove that any solution of the displaced forward difference equation is also a solution to the displaced backward equation, and vice versa. Therefore, suppose that $fjh = g(\mathbf{I}; f)$. Then

$$fj = fjhh^{-1} = (fjh)h^{-1} = g(\mathbf{I}; f)h^{-1}$$

Conversely, suppose that $fj = g(\mathbf{I}; f)h^{-1}$. Then

$$fjh = g(\mathbf{I}; f)h^{-1}h = g(\mathbf{I}; f)$$

The kernels of the differences are the same since

$$\mathrm{Im}(h^{-1}; fh^{-1}) = \mathrm{Im}(\mathbf{I}; f)h^{-1} = \mathrm{Im}(\mathbf{I}; f)$$

The last equality holds because h^{-1} is surjective. □

Theorem 10 : *If h is bijective, then $\bar{\delta}_h f \doteq \partial_{h^{-1}} f$.*

Proof: We show that any solution to the forward displaced difference equation is a solution to the undisplaced backward equation, and vice versa. First, suppose $fjh = g(h; f)$. Then

$$fj = fjhh^{-1} = g(h; f)h^{-1} = g(hh^{-1}; fh^{-1}) = g(\mathbf{I}; fh^{-1})$$

Conversely, assume $fj = g(\mathbf{I}; fh^{-1})$. Then,

$$fjh = g(\mathbf{I}; fh^{-1})h = g(h; fh^{-1}h) = g(h; f)$$

The kernels are the same since

$$\mathrm{Im}(\mathbf{I}; fh^{-1}) = \mathrm{Im}(\mathbf{I}; fh^{-1})h = \mathrm{Im}(h; f)$$

The first equality holds because h is surjective. □

2.4 Basic Properties of Forward Differences

Definition (Constant Operation): The operation $\kappa : S \longrightarrow T \longrightarrow S$ returns a constant function:

$$\kappa_x : y = x \text{ for all } y.$$

Theorem 11 (Difference With Respect to Identity): *Suppose* $\mathbf{I} : S \xrightarrow{\mathrm{I}} S$. *Then:*

$$\Delta_{\mathbf{I}} f \doteq \kappa_{\mathbf{I}}, \quad \delta_{\mathbf{I}} f \doteq \rho$$

Proof: Note that $(f\mathbf{I})_x = (\kappa_{\mathbf{I}})_x : f_x$ and $f\mathbf{I} = \rho(\mathbf{I}; f)$. □

Theorem 12 (Curried Difference of Construction): *If* $f : R \longrightarrow S$, $g : R \longrightarrow T$, *and* $h : R \longrightarrow R'$ *(where* $R' \subseteq R$*), then*

$$[\Delta_h(f;g)]_x \doteq (\Delta_h f)_x \times (\Delta_h g)_x$$

Proof:

$$\begin{aligned} [(f;g)jh]_x &= ([fjh]_x, [gjh]_x) \\ &= ([\Delta_h f]_x : f_x, [\Delta_h f]_x : g_x) \\ &= ([\Delta_h f]_x \times [\Delta_h f]_x) : (f;g)_x \end{aligned}$$

Hence $[\Delta_h f]_x \times [\Delta_h f]_x$ satisfies the difference equation defining $[\Delta_h(f;g)]_x$. □

Definition (Mu Function): The function $\mu : R \times (S \times T) \longrightarrow (R \times S) \times (R \times T)$ is defined:

$$\mu = (\mathbf{I} \times \lambda); (\mathbf{I} \times \rho)$$

It has the property $\mu : (x, (y, z)) = ((x, y), (x, z))$.

Theorem 13 :

$$\mu[f; (g; h)] = [(f; g); (f; h)]$$

Proof: Derive:

$$\begin{aligned} \mu[f;(g;h)] &= [(\mathbf{I} \times \lambda); (\mathbf{I} \times \rho)][f;(g;h)] \\ &= (\mathbf{I} \times \lambda)[f;(g;h)]; (\mathbf{I} \times \rho)[f;(g;h)] \\ &= [f; \lambda(g;h)]; [f; \rho(g;h)] \\ &= (f;g); (f;h) \end{aligned}$$

□

Theorem 14 (Difference of Construction): *If* $f : R \longrightarrow S$, $g : R \longrightarrow T$, *and* $h : R \longrightarrow R'$ *(where* $R' \subseteq R$*), then*

$$\delta_h(f;g) \doteq (\delta_h f \times \delta_h g)\mu$$

Proof: Let $f' \doteq \delta_h f$ and $g' \doteq \delta_h g$. Observe:

$$\begin{aligned}
(f;g)jh &= (fjh;gjh) \\
&= [f'(\mathbf{I};f);g'(\mathbf{I};g)] \\
&= (f' \times g')[(\mathbf{I};f);(\mathbf{I};g)] \\
&= (f' \times g')\mu[\mathbf{I};(f;g)]
\end{aligned}$$

Hence $(f' \times g')\mu$ is a difference of $(f;g)$ with respect to h. $\square$

Definition (Nu Function): The function $\nu : (S \times T) \times (U \times V) \longrightarrow (S \times U) \times (T \times V)$ is defined:

$$\nu = (\lambda^2;\lambda\rho);(\rho\lambda;\rho^2)$$

It is in effect a 2×2 transpose operation, $\nu :\ ((u,v),(x,y)) = ((u,x),(v,y))$.

Theorem 15 :

$$\nu[(f \times g);(h \times k)] = (f;h) \times (g;k)$$

Proof: Let $\phi = [(f \times g);(h \times k)]$. Then,

$$\begin{aligned}
\nu\phi &= [(\lambda^2;\lambda\rho);(\rho\lambda;\rho^2)]\phi \\
&= (\lambda^2\phi;\lambda\rho\phi);(\rho\lambda\phi;\rho^2\phi) \\
&= [\lambda(f \times g);\lambda(h \times k)];[\rho(f \times g);\rho(h \times k)] \\
&= (f\lambda;h\lambda);(g\rho;k\rho) \\
&= (f;h)\lambda;(g;k)rt \\
&= (f;h) \times (g;k)
\end{aligned}$$

$\square$

Theorem 16 (Curried Differences of Products): *If $f : R \longrightarrow T$, $g : S \longrightarrow U$, $h : R \longrightarrow R'$ and $k : S \longrightarrow S'$, then:*

$$[\Delta_{h \times k}(f \times g)]_{(x,y)} \doteq (\Delta_h f)_x \times (\Delta_k g)_y \tag{16}$$

Proof: Suppose $j_1 : R' \xrightarrow{\mathrm{I}} R$ and $j_2 : S' \xrightarrow{\mathrm{I}} S$.

$$\begin{aligned}
[(f \times g)(j_1 \times j_2)(h \times k)]_{(x,y)} &= (fj_1h \times gj_2k)_{(x,y)} \\
&= ([fj_1h]_x, [gj_2k]_y) \\
&= ([\Delta_h f]_x :\ f_x, [\Delta_k f]_y :\ g_y) \\
&= ([\Delta_h f]_x \times [\Delta_k f]_y) :\ (f \times g)_{(x,y)}
\end{aligned}$$

Hence $[\Delta_h f]_x \times [\Delta_k f]_y$ satisfies the difference equation defining $[\Delta_{h \times k}(f \times g)]_{(x,y)}$. $\square$

Equation 16 can be understood from Fig. 5.

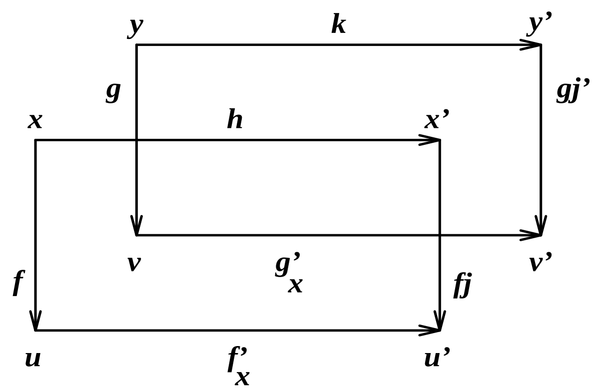

Figure 5: Curried Difference of Product

Theorem 17 (Differences of Products): *If $f : R \longrightarrow T$, $g : S \longrightarrow U$, $h : R \longrightarrow R'$ and $k : S \longrightarrow S'$, then:*

$$\delta_{h \times k}(f \times g) \doteq (\delta_h f \times \delta_k g)\nu$$

Proof: Let $f' \doteq \delta_h f$ and $g' \doteq \delta_k g$. Suppose $j_1 : R' \xrightarrow{\mathrm{I}} R$ and $j_2 : S' \xrightarrow{\mathrm{I}} S$. Observe:

$$\begin{aligned}
(f \times g)(j_1 \times j_2)(h \times k) &= fj_1h \times gj_2k \\
&= f'(\mathbf{I}; f) \times g'(\mathbf{I}; g) \\
&= (f' \times g')[(\mathbf{I}; f) \times (\mathbf{I}; g)] \\
&= (f' \times g')\nu[(\mathbf{I} \times \mathbf{I}); (f \times g)] \\
&= (f' \times g')\nu[\mathbf{I}; (f \times g)]
\end{aligned}$$

Hence $(f' \times g')\nu$ is a difference of $f \times g$ with respect to $h \times k$. $\Box$

Theorem 18 (Curried Difference with Respect to a Composition): *Suppose $S = S_0 + S'$ and $S' = S_1 + S''$. Further suppose $f : S \longrightarrow T$, $k : S \longrightarrow S'$, $h : S' \longrightarrow S''$ and $i : S' \xrightarrow{\mathrm{I}} S$. Then:*

$$(\Delta_{hk} f)_x \doteq [(\Delta_h fi)k]_x(\Delta_k f)_x$$

Proof: (See Fig. 6.) Let $G \doteq \Delta_k f$ and $G' \doteq \Delta_h fi$. Hence $(fik)_x = G_x : f_x$ and $(fijh)_{x'} = G'_{x'} : (fi)_{x'}$. Let $G''_x = (G'k)_x G_x$; our goal is to show G'' satisfies the difference equation $(fijhk)_x = G''_x : f_x$.

$$\begin{aligned}
G''_x : f_x &= (G'k)_x G_x : f_x \\
&= (G'k)_x : (fik)_x \\
&= G'_{k_x} : (fi)_{k_x} \\
&= (fijh)_{k_x} \\
&= (fijhk)_x
\end{aligned}$$

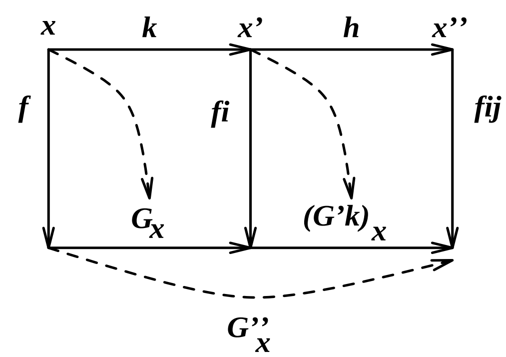

Figure 6: Curried Difference with Respect to a Composition

Hence the difference equation is satisfied. □

The formula for the curried difference with respect to a composition can be understood from Fig. 6.

Theorem 19 (Difference with Respect to a Composition): *Suppose* $S = S_0 + S'$ *and* $S' = S_1 + S''$. *Further suppose* $f : S \longrightarrow T$, $k : S \longrightarrow S'$, $h : S' \longrightarrow S''$ *and* $i : S' \xrightarrow{\mathrm{I}} S$. *Then:*

$$\delta_{hk} f \doteq \delta_h fi(k\lambda; \delta_k f)$$

Proof: We show that the function $G = g'(k\lambda; g)$, where $g \doteq \delta_k f$ and $g' \doteq \delta_h fi$, satisfies the difference equation $fihk = G(\mathbf{I}; f)$. Note that $fik = g(\mathbf{I}; f)$ and $fijh = g'(\mathbf{I}; fi)$.

$$\begin{aligned} G(\mathbf{I}; f) &= g'(k\lambda; g)(\mathbf{I}; f) \\ &= g'[k\lambda(\mathbf{I}; f); g(\mathbf{I}; f)] \\ &= g'(k; fik) \\ &= g'(\mathbf{I}; fi)k \\ &= fijhk \end{aligned}$$

Thus G satisfies the equation. □

Corollary 3 :

$$fijhk = \delta_h fi(k\lambda; \delta_k f)(\mathbf{I}; f)$$

The following theorem allows the extrapolation from the function's value at a point x_0 to its value an arbitrary point $x_n = h_n \cdots h_1 : x_0$. First, some notation and a lemma.

Notation (Product Notation for Compositions): The product notation is used for n-fold compositions:

$$\prod_{i=1}^{n} F_i = F_1 \circ F_2 \circ \cdots \circ F_n$$

Notation (Factorial Composition): Let $r =< r_1, r_2, \ldots >$ be a finite or infinite sequence of functions $r_k : S_k \longrightarrow S_{k-1}$ (for any sets S_k). Then define the *ascending (forward) factorial composition:*

$$r \uparrow_m^n = r_m r_{m+1} r_{m+2} \cdots r_{n-1} r_n = \prod_{k=m}^{n} r_k, \quad m \leq n$$

Let $r =< r_1, r_2, \ldots >$ be a finite or infinite sequence of functions $r_k : S_{k-1} \longrightarrow S_k$ (for any sets S_k). Then define the *descending (backward) factorial composition:*

$$r \downarrow_m^n = r_n r_{n-1} r_{n-2} \cdots r_{m+1} r_m = \prod_{k=n}^{m} r_k, \quad m \leq n$$

By convention $r \uparrow_{m+1}^m = r \downarrow_{m+1}^m = \mathbf{I}$.

Theorem 20 (Curried General Composition): *Suppose $S_0 = S$, $S_k \subseteq S_{k-1}$, $h_k : S_{k-1} \longrightarrow S_k$ for $k > 0$, and $i_k : S_k \xrightarrow{\mathrm{I}} S$ for $k \geq 0$. Then, for $n \geq 0$,*

$$(\Delta_{h_n \cdots h_1} f)_x \doteq \prod_{k=n}^{1} [(\Delta_{h_k} f i_{k-1}) h \downarrow_1^{k-1}]_x$$

Proof: For $n = 0$ the left-hand side of the equation is $(\Delta_{\mathbf{I}} f)_x = \mathbf{I}$; the right-hand side is also $\mathbf{I}$ since the product is empty. Now suppose $n > 0$; applying Th. 18 observe:

$$\begin{aligned}
(\Delta_{h_n \cdots h_1} f)_x &\doteq [\Delta_{h_n (h_{n-1} \cdots h_1)} f]_x \\
&\doteq [\Delta_{h_n h \downarrow_1^{n-1}} f)_x \\
&\doteq [(\Delta_{h_n} f i_{n-1}) h \downarrow_1^{n-1}]_x (\Delta_{h_{n-1} \cdots h_1} f)_x
\end{aligned}$$

Apply the inductive hypothesis to the difference with respect to $h_{n-1} \cdots h_1$ to get:

$$\begin{aligned}
&\doteq [(\Delta_{h_n} f i_{n-1}) h \downarrow_1^{n-1}]_x \prod_{k=n-1}^{1} [(\Delta_{h_k} f i_{k-1}) h \downarrow_1^{k-1}]_x \\
&\doteq \prod_{k=n}^{1} [(\Delta_{h_k} f i_{k-1}) h \downarrow_1^{k-1}]_x
\end{aligned}$$

□

Corollary 4 : *Suppose $S_n = S$, $S_k \subseteq S_{k+1}$, $h_k : S_k \longrightarrow S_{k-1}$, and $i_k : S_k \xrightarrow{\mathrm{I}} S$, all for $k \geq 0$. Then, for $n \geq 0$,*

$$(\Delta_{h_1 \cdots h_n} f)_x \doteq \prod_{k=1}^{n} [(\Delta_{h_k} f i_k) h \uparrow_{k+1}^n]_x$$

Note: In contrast to the theorem, the product and factorial here are ascending rather than descending.

Proof: This follows from the theorem by reindexing. $\Box$

Corollary 5 (Curried General Expansion): *Suppose* $S_0 = S$, $S_k \subseteq S_{k-1}$, $h_k : S_{k-1} \longrightarrow S_k$ *for* $k > 0$, *and* $i_k : S_k \xrightarrow{\mathrm{I}} S$, *for* $k \geq 0$. *Then, for* $n \geq 0$,

$$(fi_n h_n \cdots h_1)_x = \prod_{k=n}^{1} [(\Delta_{h_k} f i_{k-1}) h \downarrow_1^{k-1}]_x : f_x$$

Corollary 6 (Curried Exponential Composition): *Suppose* $S' \subseteq S$, $h : S \longrightarrow S'$ *and* $j : S' \xrightarrow{\mathrm{I}} S$*; then for* $n \geq 0$*:*

$$(\Delta_{(jh)^n} f)_x \doteq \prod_{k=n}^{1} [(\Delta_h f)(jh)^{k-1}]_x$$

Proof: Simply let $h_k = jh$ and observe $h \downarrow_1^{k-1} = (jh)^{k-1}$. Further note $\Delta_{jh} f \doteq \Delta_h f$ since

$$\begin{aligned}(\Delta_{jh} f) : f_x &= f\mathbf{I}(jh) : x \\ &= fjh : x \\ &= (\Delta_h f)_x : f_x\end{aligned}$$

$\Box$

The following theorem could be proved as a corollary to the curried general expansion theorem. However, to avoid use of the general forms for the curried differences we will prove it directly.

Lemma 3 : *Suppose* $S = S_0$ *and* $S_k \subseteq S_{k-1}$ *for* $1 \leq k \leq n$. *Further suppose* $h_k : S_{k-1} \longrightarrow S_k$ *for* $k > 0$, *and* $i_k : S_k \xrightarrow{\mathrm{I}} S$ *for* $k \geq 0$. *Then, for* $n \geq 0$*:*

$$(\mathbf{I}; fi_n) h_n \cdots h_1 = \left[\prod_{k=n}^{1} (h_k \lambda; \delta_{h_k} f i_{k-1}) \right] (\mathbf{I}; f) \tag{17}$$

Proof: This follows from Corollary 3 by induction. When $n = 0$ the left and right-hand sides of Eq. 17 reduce to $(\mathbf{I}; f)$, since the products are empty. Now suppose $n > 0$ and observe:

$$(\mathbf{I}; fi_n) h_n \cdots h_1 = [(\mathbf{I}; fi_n) h_n \cdots h_2] h_1$$

Applying the inductive hypothesis and letting $g_k = \delta_{h_k} f i_{k-1}$:

$$\begin{aligned} &= \left[\prod_{k=n}^{2} (h_k \lambda; g_k) \right] (\mathbf{I}; fi_1) h_1 \\ &= \left[\prod_{k=n}^{2} (h_k \lambda; g_k) \right] (h_1; fi_1 h_1) \\ &= \left[\prod_{k=n}^{2} (h_k \lambda; g_k) \right] [h_1; g_1(\mathbf{I}; f)] \end{aligned}$$

The last step follows from the difference equation $fi_1h_1 = g_1(\mathbf{I}; f)$. The remaining steps are algebraic simplification.

$$
\begin{aligned}
&= \left[\prod_{k=n}^{2}(h_k\lambda; g_k)\right][h_1\lambda(\mathbf{I}; f); g_1(\mathbf{I}; f)] \\
&= \left[\prod_{k=n}^{2}(h_k\lambda; g_k)\right](h_1\lambda; g_1)(\mathbf{I}; f) \\
&= \left[\prod_{k=n}^{1}(h_k\lambda; g_k)\right](\mathbf{I}; f)
\end{aligned}
$$

□

Theorem 21 (General Expansion): *Suppose $S = S_0$ and $S_k \subseteq S_{k-1}$ for $1 \leq k \leq n$. Further suppose $h_k : S_{k-1} \longrightarrow S_k$ for $k > 0$, and $i_k : S_k \xrightarrow{\mathrm{I}} S$ for $k \geq 0$. Then, for $n \geq 0$:*

$$fi_nh_n \cdots h_1 = \rho\left[\prod_{k=n}^{1}(h_k\lambda; \delta_{h_k} fi_{k-1})\right](\mathbf{I}; f)$$

Proof: Simply compose rt with both sides of Eq. 17. □

Corollary 7 : *Suppose $S = S_0$ and $S_k \subseteq S_{k-1}$ for $1 \leq k \leq n$. Further suppose $h_k : S_{k-1} \longrightarrow S_k$ for $k > 0$, and $i_k : S_k \xrightarrow{\mathrm{I}} S$ for $k \geq 0$. Then, for $n \geq 0$:*

$$\delta_{h_n \cdots h_1} f \doteq \rho \prod_{k=n}^{1}(h_k\lambda; \delta_{h_k} fi_{k-1})$$

Corollary 8 (Exponential Expansion): *Suppose $S' \subseteq S$, $f : S \longrightarrow T$, $h : S \longrightarrow S'$ and $j : S' \xrightarrow{\mathrm{I}} S$; then, for $n \geq 0$:*

$$f(jh)^n = \rho(jh\lambda; \delta_h f)^n(\mathbf{I}; f)$$

Proof: Follows from the theorem by letting $S_k = S$, $h_k = jh$, $i_k = \mathbf{I}$ and by noting $\delta_{jh} f \doteq \delta_h f$ [since $\delta_{jh} f(\mathbf{I}; f) = fjh = \delta_h f(\mathbf{I}; f)$]. □

Corollary 9 (Exponential Composition): *Suppose $S' \subseteq S$, $f : S \longrightarrow T$, $h : S' \longrightarrow S$ and $j : S' \xrightarrow{\mathrm{I}} S$; then, for $n \geq 0$:*

$$\delta_{(jh)^n} f \doteq \rho(jh\lambda; \delta_h f)^n$$

Theorem 22 (Curried Difference of Composition): *Suppose $f : S \longrightarrow T$, $g : R \longrightarrow S$ and $h : R \longrightarrow R'$; then:*

$$(\Delta_h fg)_x \doteq [\Delta_{(\Delta_h g)_x} f]g : x$$

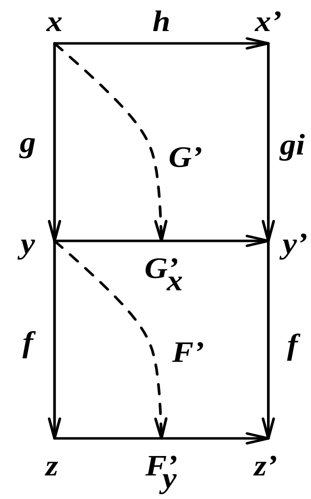

Figure 7: Curried Difference of a Composition

Proof: Let $G' \doteq \Delta_h g$ and $F' \doteq \Delta_{G'_x} f$; hence $gih : x = G'_x : g_x$ and $fG'_x : y = F'_y : f_y$. Our goal is to show $(\Delta_h fg)_x \doteq (F'g)_x$. Simply derive:

$$\begin{aligned}
(fg)ih : x &= f : (gih : x) \\
&= f : (G'_x : g_x) \\
&= F'_{g_x} : f_{g_x} \\
&= (F'g)_x : (fg)_x
\end{aligned}$$

Hence $(fg)'_x = (F'g)_x$. $\square$

The formula for the curried difference of a composition can be understood from Fig. 7.

Stating the uncurried difference of an arbitrary composition requires another definition.

Definition (Pairing Operation): The operation $\pi : S \longrightarrow T \longrightarrow S \times T$ is a curried pair-forming operation: $(\pi_x)_y = \pi_x : y = (x, y)$. Note that $f\pi_x$ is the function f with its first argument bound to x, since $(f\pi_x)_y = f : (x, y)$. It's an alternative to currying. We note without proof that $(\kappa_x; \phi) = \pi_x \phi$.

Theorem 23 (Difference of Composition): *If $f : S \longrightarrow T$, $g : R \longrightarrow S$ and $h : R \longrightarrow R'$ then:*

$$(\delta_h fg)\pi_x \doteq \delta_{\gamma_x} f(g \times \mathbf{I})\pi_x \quad \textit{where} \quad \gamma_x \doteq (\delta_h g)\pi_x$$

for all $x \in R$.

Proof: This may be proved as a corollary of Thm. 22. However it's more instructive to see the following direct proof.

Let $g' \doteq \delta_h g$; then $gjh = g'(\mathbf{I}; g)$ and $\gamma_x = g'\pi_x$. Also let $f'_x \doteq \delta_{\gamma_x} f$; then $f\gamma_x = f'_x(\mathbf{I}; f)$. We need to show that $d_x = f'_x(g \times \mathbf{I})$ solves

$$(fgjh)_x = [d_x(\mathbf{I}; fg)]_x$$

The fact that the difference of f is with respect to a variable function γ_x complicates the proof. The first-order value x interferes with the use of higher-order (algebraic) reasoning. Therefore, the proof can be simplified by working entirely at the function level. To accomplish this we make use of the fact that there is an isomorphism between domain values x and constant functions κ_x, and an isomorphism between range values f_x and compositions with constant functions $f\kappa_x$. Therefore it will suffice to prove the following functional equation (for all x):

$$fgjh\kappa_x = d_x(\mathbf{I}; fg)\kappa_x$$

The necessary algebraic manipulations, which use the difference equations for g' and f'_x, are easy:

$$\begin{aligned}
fgjh\kappa_x &= fg'(\mathbf{I}; g)\kappa_x \\
&= fg'(\kappa_x; g\kappa_x) \\
&= fg'\pi_x g\kappa_x \\
&= f\gamma_x g\kappa_x \\
&= f'_x(\mathbf{I}; f)g\kappa_x \\
&= f'_x(g; fg)\kappa_x \\
&= f'_x(g \times \mathbf{I})(\mathbf{I}; fg)\kappa_x \\
&= d_x(\mathbf{I}; fg)\kappa_x
\end{aligned}$$

Thus d_x solves the difference equation at x. $\Box$

Next we turn to the forward difference of sums of functions. The presentation requires several definitions and preliminary results.

Definition (Functional Direct Sum): Suppose $f : R \longrightarrow T$ and $g : S \longrightarrow U$; then $(f + g) : R + S \longrightarrow T + U$ is the function:

$$f + g = if \mid jg$$

where $i : T \xrightarrow{\mathrm{I}} T + U$ and $j : U \xrightarrow{\mathrm{I}} T + U$.

Theorem 24 : *Suppose $f : R \longrightarrow T$ and $g : S \longrightarrow U$. Let:*

$$i : R \xrightarrow{\mathrm{I}} R + S, \quad j : S \xrightarrow{\mathrm{I}} R + S, \quad i' : T \xrightarrow{\mathrm{I}} T + U, \quad j' : U \xrightarrow{\mathrm{I}} T + U$$

Then: $(f + g)i = i'f$ and $(f + g)j = j'g$. Also, $(f \mid g)(h + k) = fh \mid dk$ and $(f + g)(h + k) = fh + dk$.

Definition (Tau Function): Let $i : R \xrightarrow{\mathrm{I}} R + S$, $j : S \xrightarrow{\mathrm{I}} R + S$, $i' : T \xrightarrow{\mathrm{I}} T + U$ and $j' : U \xrightarrow{\mathrm{I}} T + U$. Since $(i \times i') \mid (j \times j')$ is an injection, it has left inverses. Let

$$\tau : (R + S) \times (T + U) \longrightarrow R \times T + S \times U$$

be any left inverse:[6]

$$\tau(i \times i' \mid j \times j') = \mathbf{I}$$

The τ function converts a product of sums into a sum of products by matching up the corresponding variants from the components of its input pair. Its purpose is shown by the following lemma.

Lemma 4 : *Suppose* $f : X \longrightarrow R$, $g : Y \longrightarrow S$, $h : X \longrightarrow T$ *and* $k : Y \longrightarrow U$. *Then,*

$$\tau(f + g; h + k) = (f; h) + (g; k)$$

Proof: We apply the definition of +, Cor. 1 and algebraic simplification:

$$\begin{aligned}
(f + g; h + k) &= (if \mid jg; i'h \mid j'k) \\
&= (if; i'h) \mid (jg; j'k) \\
&= (i \times i')(f; h) \mid (j \times j')(g; k) \\
&= (i \times i' \mid j \times j')[(f; h) + (g; k)]
\end{aligned}$$

Composing with τ on both sides gives the result. □

Theorem 25 (Difference of Direct Sum): *Suppose* $f : R \longrightarrow T$, $g : S \longrightarrow U$, $h : R \longrightarrow R'$ *and* $k : S \longrightarrow S'$, *where* $R' \subseteq R$ *and* $S' \subseteq S$. *Then:*

$$\delta_{h+k} f + g \doteq (\delta_h f + \delta_k g)\tau$$

Proof: Let $f' \doteq \delta_h f$ and $g' \doteq \delta_k g$; we show that $(f' + g')\tau$ satisfies the difference equation.

$$\begin{aligned}
(f' + g')\tau(\mathbf{I}; f + g) &= (f' + g')\tau(\mathbf{I} + \mathbf{I}; f + g) \\
&= (f' + g')[(\mathbf{I}; f) + (\mathbf{I}; g)] \\
&= f'(\mathbf{I}; f) + g'(\mathbf{I}; g) \\
&= f j_1 h + g j_2 k \\
&= (f + g)(j_1 + j_2)(h + k)
\end{aligned}$$

□

Definition (Upsilon Bijection): Define υ to be the inverse of the bijection $i \times \mathbf{I} \mid j \times \mathbf{I}$:

$$\upsilon = (i \times \mathbf{I} \mid j \times \mathbf{I})^{-1}$$

Clearly, υ could be defined constructively.

[6] This is obviously a nonconstructive definition; a constructive definition is routine.

Lemma 5 :

$$v(f+g; h \mid k) = (f;h) + (g;k)$$

Proof: Apply the definition of + and Cor. 1:

$$\begin{aligned}(f+g;h \mid k) &= (if \mid jg; h \mid k) \\ &= (if;h) \mid (jg;k) \\ &= (i \times \mathbf{I})(f;h) \mid (j \times \mathbf{I})(g;k) \\ &= (i \times \mathbf{I} \mid j \times \mathbf{I})[(f;h) + (g;k)]\end{aligned}$$

Now compose with v on both sides. □

Theorem 26 (Difference of an Alternation):

$$\delta_{h+k} f \mid g \doteq (\delta_h f \mid \delta_k g) v$$

where $f : S \longrightarrow U$, $g : T \longrightarrow U$, $h : S \longrightarrow S'$ *and* $k : T \longrightarrow T'$.

Proof: Let f' and g' be the differences of f and g; substitute into the difference equation:

$$\begin{aligned}(f \mid g)(j_1 + j_2)(h+k) &= f j_1 h \mid g j_2 k \\ &= f'(\mathbf{I}; f) \mid g'(\mathbf{I}; g) \\ &= (f' \mid g')[(\mathbf{I}; f) + (\mathbf{I}; g)] \\ &= (f' \mid g') v(\mathbf{I} + \mathbf{I}; f \mid g) \\ &= (f' \mid g') v(\mathbf{I}; f \mid g)\end{aligned}$$

□

2.5 Basic Properties of Backward Differences

Since many of these proofs are very similar to those for forward differences, they are omitted.

Theorem 27 (Difference With Respect to Embedding): *Suppose* $S = S_0 + S'$, $\mathbf{I} : S \xrightarrow{\mathrm{I}} S$, $i : S_0 \xrightarrow{\mathrm{I}} S$ *and* $j : S' \xrightarrow{\mathrm{I}} S$. *Then:*

$$\begin{aligned}\nabla_{\mathbf{I}} f &\doteq \kappa_{\mathbf{I}}, \quad \nabla_i f \doteq \kappa_{\mathbf{I}}, \quad \nabla_j f \doteq \kappa_{\mathbf{I}} \\ \partial_{\mathbf{I}} f &\doteq \rho, \quad \partial_i f \doteq \rho, \quad \partial_j f \doteq \rho\end{aligned}$$

Note that the differences have different kernels, so we can't conclude, for example, that $\nabla_{\mathbf{I}} f \doteq \nabla_i f$.

Theorem 28 (Differences of Constructions): *If $f : R \longrightarrow S$, $g : R \longrightarrow T$, and $r : R' \longrightarrow R$, then*

$$\partial_r(f;g) \doteq (\partial_r f \times \partial_r g)\mu$$

Theorem 29 (Differences of Products): *If $f : R \longrightarrow T$, $g : S \longrightarrow U$, $r : R' \longrightarrow R$ and $s : S' \longrightarrow S$, then:*

$$\partial_{r \times s}(f \times g) \doteq (\partial_r f \times \partial_s g)\nu$$

Lemma 6 : *Suppose $f : R \longrightarrow T \longrightarrow U$, $g : S \longrightarrow T \longrightarrow U$, $h : R \longrightarrow T$ and $k : S \longrightarrow T$; then:*

$$\mathsf{S}fh \mid \mathsf{S}gk = \mathsf{S}(f \mid g)(h \mid k)$$

Note that $(f \mid g) : R + S \longrightarrow T \longrightarrow U$ and $(h \mid k) : R + S \longrightarrow T$.

Proof: Recall that in general $F = Fi \mid Fj$ (Eqs. 13), so we will be done if we can show $[\mathsf{S}(f \mid g)(h \mid k)]i = \mathsf{S}fh$ and $[\mathsf{S}(f \mid g)(h \mid k)]j = \mathsf{S}gk$. To do this we make use of a simple property of S, namely $(\mathsf{S}ab)c = \mathsf{S}(ac)(bc)$, and derive:

$$[\mathsf{S}(f \mid g)(h \mid k)]i = \mathsf{S}[(f \mid g)i][(h \mid k)j] = \mathsf{S}fh$$

The proof for j is analogous. □

Theorem 30 (Curried Difference with Respect to an Alternation): *The curried difference with respect to an alternation is the alternation of the differences:*

$$\nabla_{r|s} f \doteq \nabla_r f \mid \nabla_s f$$

where $S' = S'_1 + S'_2$, $r : S'_1 \longrightarrow S$ and $s : S'_2 \longrightarrow S$.

Proof: Suppose $j_1 : S'_1 \xrightarrow{\mathrm{I}} S'_1 + S'_2$ and $j_2 : S'_2 \xrightarrow{\mathrm{I}} S'_1 + S'_2$. Note that $j = j_1 \mid j_2$ is the standard injection $j : S' \xrightarrow{\mathrm{I}} S$. Now derive:

$$\begin{aligned} fj &= f(j_1 \mid j_2) \\ &= fj_1 \mid fj_2 \\ &= \mathsf{S}(\nabla_r f)(fr) \mid \mathsf{S}(\nabla_s f)(fs) \\ &= \mathsf{S}(\nabla_r f \mid \nabla_s f)(fr \mid fs) \end{aligned}$$

The last line follows from Lemma 6. Factoring f from $fr \mid fs$ shows we have satisfied the backward difference equation:

$$fj = \mathsf{S}(\nabla_r f \mid \nabla_s f)[f(r \mid s)]$$

□

Note that for forward differences there is no theorem analogous to the preceding.

Corollary 10 (Difference with Respect to an Alternation):

$$\partial_{r|s} f \doteq (\partial_r f \mid \partial_s f)\upsilon$$

Proof: This can be proved as a corollary to the theorem, or directly as follows. Let $g \doteq \partial_r f$ and $g' \doteq \partial_s f$. Then:

$$\begin{aligned}
(g \mid g')\upsilon[\mathbf{I}; f(r \mid s)] &= (g \mid g')\upsilon(\mathbf{I}; fr \mid fs) \\
&= (g \mid g')\upsilon(\mathbf{I} + \mathbf{I}; fr \mid fs) \\
&= (g \mid g')[(\mathbf{I}; fr) + (\mathbf{I}; fs)] \\
&= g(\mathbf{I}; fr) \mid g'(\mathbf{I}; fs) \\
&= fj_1 \mid fj_2 \\
&= f(j_1 \mid j_2) \\
&= fj
\end{aligned}$$

□

Theorem 31 (Difference with Respect to a Composition): *Suppose $S = S_0 + S'$ and $S' = S_1 + S''$. Further suppose $f : S \longrightarrow T$, $r : S' \longrightarrow S$, $s : S'' \longrightarrow S'$ and $i : S' \xrightarrow{\mathrm{I}} S$. Then:*

$$\partial_{rs} f \doteq \partial_s f i[\lambda; \partial_r f(s \times \mathbf{I})]$$

Corollary 11 :

$$fij = \partial_s f i[\lambda; \partial_r f(s \times \mathbf{I})](\mathbf{I}; frs)$$

Theorem 32 (Curried Difference with Respect to a Composition): *Suppose $S = S_0 + S'$ and $S' = S_1 + S''$. Further suppose $f : S \longrightarrow T$, $r : S' \longrightarrow S$, $s : S'' \longrightarrow S'$ and $i : S' \xrightarrow{\mathrm{I}} S$. Then:*

$$(\nabla_{rs} f)_x \doteq (\nabla_s f i)_x[(\nabla_r f)s]_x \tag{18}$$

Theorem 33 (Curried General Composition): *Suppose $S_0 = S$, $S_k \subseteq S_{k-1}$ and r is the sequence of functions $r_k : S_k \longrightarrow S_{k-1}$ for $1 \leq k \leq n$. Further suppose $i_k : S_k \xrightarrow{\mathrm{I}} S$ for $0 \leq k \leq n$. Then, for $f : S \longrightarrow T$ and $n \geq 0$:*

$$(\nabla_{r_1 \cdots r_n} f)_x \doteq \prod_{k=n}^{1} [(\nabla_{r_k} f i_{k-1}) r \uparrow_{k+1}^{n}]_x \tag{19}$$

Proof: For $n = 0$ the left-hand side is $(\nabla_{\mathbf{I}} f)_x \doteq (\kappa_{\mathbf{I}})_x = \mathbf{I}$. The right-hand side is also $\mathbf{I}$ since the product is empty.

Now suppose $n > 0$. By the composition theorem (Eq. 18):

$$\begin{aligned}
(\nabla_{r\uparrow_1^n} f)_x &\doteq (\nabla_{r_n} f i_{n-1})_x[(\nabla_{r\uparrow_1^{n-1}} f) r_n]_x \\
&\doteq (\nabla_{r_n} f i_{n-1})_x(\nabla_{r\uparrow_1^{n-1}} f)_{r_n : x}
\end{aligned}$$

By the inductive hypothesis:

$$\begin{aligned}
&\doteq (\nabla_{r_n} f i_{n-1})_x \prod_{k=n-1}^{1} [(\nabla_{r_k} f i_{k-1}) r \uparrow_{k+1}^{n-1}]_{r_n : x} \\
&\doteq (\nabla_{r_n} f i_{n-1})_x \prod_{k=n-1}^{1} [(\nabla_{r_k} f i_{k-1}) r \uparrow_{k+1}^{n}]_x \\
&\doteq \prod_{k=n}^{1} [(\nabla_{r_k} f i_{k-1}) r \uparrow_{k+1}^{n}]_x
\end{aligned}$$

□

Corollary 12 : *Suppose* $S_n = S$*,* $S_{k-1} \subseteq S_k$*,* $i_k : S_k \xrightarrow{\mathrm{I}} S$ *for* $k \geq 0$ *and* $r_k : S_{k-1} \longrightarrow S_k$ *for* $k \geq 1$*. Then for* $n \geq 0$*:*

$$(\nabla_{r_n \cdots r_1} f)_x \doteq \prod_{k=1}^{n} [(\nabla_{r_k} f i_k) r \downarrow_1^{k-1}]_x \tag{20}$$

Corollary 13 (Curried General Expansion): *Suppose* $S_0 = S$*,* $S_k \subseteq S_{k-1}$*, and* r *is the sequence of functions* $r_k : S_k \longrightarrow S_{k-1}$ *for* $1 \leq k \leq n$*. Further suppose* $i_k : S_k \xrightarrow{\mathrm{I}} S$ *for* $0 \leq k \leq n$*. Then, for* $f : S \longrightarrow T$ *and* $n \geq 0$*:*

$$f i_n : x = \prod_{k=n}^{1} [(\nabla_{r_k} f i_{k-1}) r \uparrow_{k+1}^{n}]_x : (f r_1 \cdots r_n)_x \tag{21}$$

Note: In the following corollaries we would like to write $rsupnx$ to denote an n-step reduction of an object x in S'. This would presume that all the intermediate objects $r^m x$, for $m < n$, are also in S'. Unfortunately, exponentiation of $r : S' \longrightarrow S$ is not permitted, since its domain and range are not the same. We can accomplish the same effect however by using the injection $i : S_0 \xrightarrow{\mathrm{I}} S$ to complete r. Thus $(i \mid r)^n$ is permitted, since $(i \mid r) : S \longrightarrow S$. Note that $(i \mid r)_{x_0} = x_0$ for $x_0 \in S_0$.

Corollary 14 (Curried Exponential Composition): *Suppose* $S = S_0 + S'$*,* $r : S' \longrightarrow S$*, and* $i : S_0 \xrightarrow{\mathrm{I}} S$*. Then for all* $n \geq 0$*:*

$$(\nabla_{(i|r)^n} f)_x \doteq \prod_{k=1}^{n} [(\kappa_{\mathbf{I}} \mid \nabla_r f)(i \mid r)^{k-1}]_x \tag{22}$$

Proof: This follows from the Cor. 12 to the general expansion theorem. Rewrite Eq. 20 with primes as follows:

$$(\nabla_{r'_n \cdots r'_1} f)_x \doteq \prod_{k=1}^{n'} [(\nabla_{r'_k} f i'_k) r' \downarrow_1^{k-1}]_x \tag{23}$$

Now let:

$$\begin{aligned}
n' &= n \\
r'_k &= i \mid r, \ k > 0 \\
S'_k &= S, \ k \geq 0 \\
i'_k &= \mathbf{I}, \ k > 0
\end{aligned}$$

Substitute these into Eq. 22 and simplify:

$$\begin{aligned}
(\nabla_{(i|r)^n} f)_x &\doteq \prod_{k=1}^{n} [(\nabla_{r'_k} f i'_k) r' \downarrow_1^{k-1}]_x \\
&\doteq \prod_{k=1}^{n} [(\nabla_{i|r} f \mathbf{I}) r' \downarrow_1^{k-1}]_x \\
&\doteq \prod_{k=1}^{n} [(\nabla_{i|r} f)(i \mid r)^{k-1}]_x \\
&\doteq \prod_{k=1}^{n} [(\nabla_i f \mid \nabla_r f)(i \mid r)^{k-1}]_x \\
&\doteq \prod_{k=1}^{n} [(\kappa_{\mathbf{I}} \mid \nabla_r f)(i \mid r)^{k-1}]_x
\end{aligned}$$

□

Corollary 15 (Curried Exponential Expansion): *Suppose* $S = S_0 + S'$, $r : S' \longrightarrow S$, *and* $i : S_0 \xrightarrow{\mathrm{I}} S$. *Then for all* $n \geq 0$*:*

$$f_x = \left\{ \prod_{k=1}^{n} [(\kappa_{\mathbf{I}} \mid \nabla_r f)(i \mid r)^{k-1}]_x \right\} : [f(i \mid r)^n]_x$$

Corollary 16 : *If* $f : S \longrightarrow T$ *and* $r : S \longrightarrow S$ *then*

$$\begin{aligned}
(\nabla_{r^n} f)_x &\doteq \prod_{k=1}^{n} (\nabla_r f r^{k-1})_x \\
f_x &= \left\{ \prod_{k=1}^{n} (\nabla_r f r^{k-1})_x \right\} : (f r^n)_x
\end{aligned}$$

Proof: This follows from the preceding two corollaries when $S_0 = S - S' = \emptyset$. □

Lemma 7 : *Suppose that* $G_k : S \longrightarrow T \longrightarrow U$ *is the curried version of* $g_k : S \times T \longrightarrow U$*; that is,* $G_k : x : y = g_k : (x, y)$. *Then for* $n \geq 0$,

$$\left(x, \left[\prod_{k=1}^{n} G_k : x \right] : y \right) = \left[\prod_{k=1}^{n} (\lambda; g_k) \right] : (x, y)$$

Proof: The proof is inductive. When $n = 0$ the empty products are $\mathbf{I}$, so both sides of the equation are (x, y). For $n > 0$ derive as follows:

$$\begin{aligned} \left(x, \left[\prod_{k=1}^{n} G_k : x\right] : y\right) &= \left(x, \left[\prod_{k=1}^{n-1} G_k : x\right] [G_n : x] : y\right) \\ &= \left(x, \left[\prod_{k=1}^{n-1} G_k : x\right] : [g_n : (x, y)]\right) \end{aligned}$$

Now apply the inductive hypothesis to obtain:

$$\begin{aligned} &= \left[\prod_{k=1}^{n-1} (\lambda; g_k)\right] : (x, g_n : (x, y)) \\ &= \left[\prod_{k=1}^{n} (\lambda; g_k)\right] : (x, y) \end{aligned}$$

□

Theorem 34 : *Suppose that $G_k : S \longrightarrow T \longrightarrow U$ is the curried version of $g_k : S \times T \longrightarrow U$; that is, $G_k : x : y = g_k : (x, y)$. Then for $n \geq 0$,*

$$\left[\prod_{k=1}^{n} G_k : x\right] : y = \left[\rho \prod_{k=1}^{n} (\lambda; g_k)\right] : (x, y)$$

Theorem 35 (General Expansion): *Suppose $S = S_0$, $S_k \subseteq S_{k-1}$ and $r_k : S_k \longrightarrow S_{k-1}$ for $1 \leq k \leq n$. Further suppose $i_k : S_k \stackrel{\mathrm{I}}{\longrightarrow} S$ for $0 \leq k \leq n$. Then for $n \geq 0$:*

$$fi_n = \rho \left\{ \prod_{k=n}^{1} [\lambda; \partial_{r_k} fi_{k-1} (r \uparrow_{k+1}^{n} \times \mathbf{I})] \right\} (\mathbf{I}; fr_1 \cdots r_n)$$

Proof: First note that if G is the curried version of g, then $G\phi$ is the curried version of $g(\phi \times \mathbf{I})$. In particular, $(\nabla_{r_k} fi_{k-1})\phi$ is the curried version of $dub_{r_k} fi_{k-1}(\phi \times \mathbf{I})$. Next apply Thm. 34 and Cor. 13:

$$\begin{aligned} fi_n : x &= \prod_{k=n}^{1} [(\nabla_{r_k} fi_{k-1}) r \uparrow_{k+1}^{n}]_x : (fr_1 \cdots r_n)_x \\ &= \rho \left\{ \prod_{k=n}^{1} [\lambda; \partial_{r_k} fi_{k-1} (r \uparrow_{k+1}^{n} \times \mathbf{I})] \right\} : (x, fr_1 \cdots r_n : x) \\ &= \rho \left\{ \prod_{k=n}^{1} [\lambda; \partial_{r_k} fi_{k-1} (r \uparrow_{k+1}^{n} \times \mathbf{I})] \right\} (\mathbf{I}; fr_1 \cdots r_n) : x \end{aligned}$$

□

Corollary 17 (General Composition): *Suppose $S = S_0$, $S_k \subseteq S_{k-1}$, and $r_k : S_k \longrightarrow S_{k-1}$ for $1 \leq k \leq n$. Further suppose $i_k : S_k \xrightarrow{\mathrm{I}} S$ for $0 \leq k \leq n$. Then for $n \geq 0$:*

$$\partial_{r_1 \cdots r_n} f \doteq \rho \prod_{k=n}^{1} [\lambda; \partial_{r_k} f i_{k-1} (r \uparrow_{k+1}^{n} \times \mathbf{I})]$$

Corollary 18 (Exponential Composition): *Suppose $f : S \longrightarrow T$, $r : S' \longrightarrow S$, and $i : S_0 \xrightarrow{\mathrm{I}} S$. Then for all $n \geq 0$:*

$$\partial_{(i|r)^n} f \doteq \rho \prod_{k=1}^{n} \{\lambda; (\rho \mid \partial_r f) \upsilon [(i \mid r)^{k-1} \times \mathbf{I}]\}$$

Proof: This can be proved either by applying Thm. 34 to the formula in Cor. 14, or by a direct proof analogous to that of Cor. 14. Also note that

$$\partial_{i|r} f \doteq (\partial_i f \mid \partial_r f) \upsilon \doteq (\rho \mid \partial_r f) \upsilon$$

□

Corollary 19 (Exponential Expansion): *Suppose $f : S \longrightarrow T$, $r : S' \longrightarrow S$, and $i : S_0 \xrightarrow{\mathrm{I}} S$. Then for all $n \geq 0$:*

$$f = \rho \left(\prod_{k=1}^{n} \{\lambda; (\rho \mid \partial_r f) \upsilon [(i \mid r)^{k-1} \times \mathbf{I}]\} \right) [\mathbf{I}; f (i \mid r)^n]$$

Corollary 20 : *If $f : S \longrightarrow T$ and $r : S \longrightarrow S$ then*

$$\begin{aligned} \partial_{r^n} f &\doteq \rho \prod_{k=1}^{n} [\lambda; \partial_r f \upsilon (r^{k-1} \times \mathbf{I})] \\ f &= \rho \left\{ \prod_{k=1}^{n} [\lambda; \partial_r f \upsilon (r^{k-1} \times \mathbf{I})] \right\} (\mathbf{I}; f r^n) \end{aligned}$$

Theorem 36 (Difference of Composition): *If $f : S \longrightarrow T$, $g : R \longrightarrow S$ and $r : R' \longrightarrow R$ then:*

$$(\partial_r f g) \pi_x \doteq \delta_{\gamma_x} f (g r \times \mathbf{I}) \pi_x \quad \textit{where} \quad \gamma_x \doteq (\partial_r g) \pi_x$$

for all $x \in R'$.

Note that the backward difference of the composition is defined in terms of a *forward* difference and a backward difference.

Theorem 37 (Difference of Direct Sum): *Suppose $f : R \longrightarrow T$, $g : S \longrightarrow U$, $r : R' \longrightarrow R$ and $s : S' \longrightarrow S$, where $R' \subseteq R$ and $S' \subseteq S$. Then:*

$$\partial_{r+s} f + g \doteq (\partial_r f + \partial_s g) \tau$$

Theorem 38 (Difference of an Alternation):

$$\partial_{r+s} f \mid g \doteq (\partial_r f \mid \partial_s g) \upsilon$$

where $f : S \longrightarrow U$, $g : T \longrightarrow U$, $r : S' \longrightarrow S$ and $s : T' \longrightarrow T$.

3 Functional Integration

3.1 Properties Preparatory to Integration

Functional integration is the process of accumulating functional differences subject to a "boundary function." It is functional integration that permits the derivation of global behavior from local behavior. In this section we obtain results that facilitate the development of the theory of functional integration.

The following theorems define functions recursively in terms of their differences; they are preparatory to the definition of the functional integrals in Section 3.2.

Theorem 39 (Recursion Based on Backward Differences): *If $b = fi$ then*

$$f = b \mid \partial_r f(\mathbf{I}; fr) \tag{24}$$

Furthermore, if r is injective then

$$f = b \mid \bar{\partial}_r f(\mathbf{I}; f)r$$

Proof: By Theorem 3 the undisplaced difference exists and satisfies $fj = \partial_r f(\mathbf{I}; fr)$. Derive:

$$f = f(i \mid j) = fi \mid fj = b \mid \partial_r f(\mathbf{I}; fr)$$

The other result is proved analogously. □

Equation 24 is a standard recursive definition in disguise. Let $g = \partial_r f$ and observe:

$$\begin{aligned} f_x &= [b \mid g(\mathbf{I}; fr)]_x \\ &= \begin{cases} b_x, & \text{if } x \in S_0 \\ g(\mathbf{I}; fr)]_x, & \text{if } x \in S' \end{cases} \\ &= \begin{cases} b_x, & \text{if } x \in S_0 \\ g : [x, (fr)_x], & \text{if } x \in S' \end{cases} \end{aligned}$$

That is, in more conventional notation,

$$f(x) = \begin{cases} b(x), & \text{if } x \in S_0 \\ g\{x, f[r(x)]\}, & \text{if } x \in S' \end{cases}$$

Notice, however, the remarkable fact that *all* functions satisfy Eq. 24.

Theorem 40 (Recursion Based on Forward Differences): *If $hr = \mathbf{I}$ and $b = fi$, then*

$$f = b \mid \delta_h f(\mathbf{I}; f)r$$
$$f = b \mid \bar{\delta}_h f(\mathbf{I}; fr)$$

Proof: Derive:

$$f = fi \mid fj = b \mid fjhr = b \mid \delta_h f(\mathbf{I}; f)r$$

The other result is proved analogously. $\Box$

Definition (Discriminators): If $S = T + U$, then we define $\mathsf{1st?} : S \longrightarrow \mathbf{2}$ and $\mathsf{2nd?} : S \longrightarrow \mathbf{2}$ as follows:

$$\begin{aligned} \mathsf{1st?} &= \kappa_{\mathbf{true}} \mid \kappa_{\mathbf{false}} \\ \mathsf{2nd?} &= \kappa_{\mathbf{false}} \mid \kappa_{\mathbf{true}} \end{aligned}$$

Definition (Reduction): Suppose $S = S_0 + S'$. Then we say the $r : S' \longrightarrow S$ *reduces* S if for each $x \in S$ there is an $n \geq 0$ such that $\mathsf{1st?}[(i \mid r)^n : x]$. (This implies $S_0 \neq \emptyset$, i.e., the direct sum is nontrivial.)

Definition (Absolute Value Sign): Suppose $r : S' \longrightarrow S$ reduces $S = S_0 + S'$. Then, for any $x \in S$ define:

$$|x| = \min\{n \geq 0 \mid \mathsf{1st?}[(i \mid r)^n : x]\}$$

Definition (Extension): We say that $f : S \longrightarrow T$ *extends* $b : S_0 \longrightarrow T$ provided $S_0 \subset S$ and $b = fi$, where $i : S_0 \xrightarrow{\mathrm{I}} S$.

Definition (Phi Functions): Suppose we have $S = S_0 + S'$, $b : S_0 \longrightarrow T$, $G : S' \longrightarrow T \longrightarrow T$, $r : S' \longrightarrow S$, and that r reduces S. Then define $\Phi(b, G, r) : S \longrightarrow T$ by:

$$[\Phi(b, G, r)]_x = \left\{ \prod_{k=1}^{|x|} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k-1}]_x \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}]_x$$

where $\omega \in S_0$ is arbitrary, but fixed. If $g : S' \times T \longrightarrow T$ then we define analogously:

$$\phi(b, g, r) : x = \rho \left(\prod_{k=1}^{|x|} \{\lambda; (\rho \mid g)\upsilon[(i \mid r)^{k-1} \times \mathbf{I}]\} \right) [\mathbf{I}; b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}] : x$$

Note: The formulas for the phi functions are complicated by our requirement that all functions be total. If, as is common in computer science, we drop this requirement, then:

$$\begin{aligned} [\Phi(b, G, r)]_x &= \left[\prod_{k=1}^{|x|} (Gr^{k-1})_x \right] : (br^{|x|})_x \\ \phi(b, g, r) : x &= \rho \left\{ \prod_{k=1}^{|x|} [\lambda; g(r^{k-1} \times \mathbf{I})] \right\} (\mathbf{I}; br^{|x|}) : x \end{aligned}$$

Lemma 8 : *Suppose we have $S = S_0 + S'$, $b : S_0 \longrightarrow T$, $G : S' \longrightarrow T \longrightarrow T$, $r : S' \longrightarrow S$, and that r reduces S. Then $\Phi(b, G, r)$ satisfies the curried backward difference equation:*

$$\Phi(b, G, r)j = \mathsf{S}G[\Phi(b, G, r)r]$$

Proof: Let $f = \Phi(b, G, r)$; our goal is to show $(fj)_x = G_x : (fr)_x$ for all $x \in S'$. Note that $(\kappa_{\mathbf{I}} \mid G)j = G$. Therefore, the right-hand side of the difference equation is:

$$\begin{aligned} G_x : (fr)_x &= [(\kappa_{\mathbf{I}} \mid G)j]_x : f_{r:x} \\ &= [(\kappa_{\mathbf{I}} \mid G)j]_x : \left(\left\{ \prod_{k=1}^{n'} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k-1}]_{r:x} \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{n'}]_{r:x} \right) \end{aligned}$$

where $n' = |r : x|$. Now note that $(i \mid r)j = r$ and simplify the product:

$$\begin{aligned} &= [(\kappa_{\mathbf{I}} \mid G)j]_x \left\{ \prod_{k=1}^{n'} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k-1} r]_x \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{n'} r]_x \\ &= [(\kappa_{\mathbf{I}} \mid G)j]_x \left\{ \prod_{k=1}^{n'} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k} j]_x \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{n'+1} j]_x \\ &= [(\kappa_{\mathbf{I}} \mid G)]_{j:x} \left\{ \prod_{k=1}^{n'} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k}]_{j:x} \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{n'+1}]_{j:x} \\ &= \left\{ \prod_{k=0}^{n'} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k}]_{j:x} \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{n'+1}]_{j:x} \end{aligned}$$

Now note that $n' = |r : x| = |x| - 1$, so:

$$\begin{aligned} &= \left\{ \prod_{k=0}^{|x|-1} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k}]_{j:x} \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}]_{j:x} \\ &= \left\{ \prod_{k=1}^{|x|} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k-1}]_{j:x} \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}]_{j:x} \\ &= f_{j:x} \\ &= (fj)_x \end{aligned}$$

□

Theorem 41 (Unique Solution to Curried Backward Difference Equation): *Suppose we have $S = S_0 + S'$, $b : S_0 \longrightarrow T$, $G : S' \longrightarrow T \longrightarrow T$, $r : S' \longrightarrow S$, and that r reduces S. Then $f = \Phi(b, G, r)$ is the unique solution extending b to the curried backward difference equation $fj = \mathsf{S}G(fr)$.*

Proof: We have already shown that the Φ function is a solution; it remains to show that it is unique. Let f be any solution to the difference equation; by Cor. 15 we know that it satisfies:

$$f_x = \left\{ \prod_{k=1}^{n} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k-1}]_x \right\} : [f(i \mid r)^n]_x$$

for all n and x. Therefore let $n = |x|$ and compare the definition of the Φ function:

$$[\Phi(b, G, r)]_x = \left\{ \prod_{k=1}^{n} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k-1}]_x \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^n]_x \tag{25}$$

They are identical to the left of the application sign ':'. Therefore we will have shown $f = \Phi(b, G, r)$ if we can show:

$$[f(i \mid r)^n]_x = [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^n]_x \tag{26}$$

To accomplish this note that since $n = |x|$, $\mathsf{1st}?[(i \mid r)^n : x]$ is true. Hence for some $x_0 \in S_0$, $(i \mid r)^n : x = i : x_0$. Substitute this in the left-hand side of Eq. 26 to get

$$f(i \mid r)^n : x = fi : x_0 = b_{x_0}$$

Substitute it into the right-hand side of Eq. 26 to get

$$b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^n : x = b(\mathbf{I} \mid \kappa_\omega)i : x_0 = b_{x_0}$$

Thus any solution extending b must be the Φ function. □

Corollary 21 (Unique Solution to Backward Difference Equation): *Suppose we have $S = S_0 + S'$, $b : S_0 \longrightarrow T$, $g : S' \times T \longrightarrow T$, $r : S' \longrightarrow S$, and that r reduces S. Then $f = \phi(b, g, r)$ is the unique solution extending b to the (uncurried) backward difference equation $fj = g(\mathbf{I}; fr)$.*

Corollary 22 (Unique Solution for Displaced Curried Backward Difference Equation): *Suppose we have $S = S_0 + S'$, $f : S \longrightarrow T$, $G : S \longrightarrow T \longrightarrow T$, $r : S' \longrightarrow S$ and that r reduces S. Then $f = \Phi(b, Gr, r)$ is the unique solution extending b to the displaced curried backward difference equation $fj = \mathsf{S}(Gr)(fr)$.*

Corollary 23 (Unique Solution for Displaced Backward Difference Equation): *Suppose we have $S = S_0 + S'$, $b : S_0 \longrightarrow T$, $g : S \times T \longrightarrow T$, $r : S' \longrightarrow S$, and that r reduces S. Then $f = \phi[b, g(r \times \mathbf{I}), r]$ is the unique solution extending b to the displaced (uncurried) backward difference equation $fj = g(\mathbf{I}; f)r$.*

3.2 Definitions of Integrals and Fundamental Theorems

The term 'integral' needs some justification. It might seem that the inverse of a functional difference should be a functional *summation,* but this would be misleading, since it would suggest direct sums or alternations, and in fact the inverse of a functional difference is more like a product (i.e., repeated composition). An alternative would be to call a functional difference a functional *derivative,* but this could be confused with Fréchet and Gâteaux derivatives. Thus we seem to be forced into the functional difference/integral pair. As a consolation it should be noted that 'integral' accurately reflects the operation's generation of the "whole" function from its parts.[7]

Definition (Undisplaced Forward and Displaced Backward Integrals): Suppose $S = S_0 + S'$, $h : S \longrightarrow S'$, $g : S \times T \longrightarrow T$ and $b : S_0 \longrightarrow T$ are given. Then $f : S \longrightarrow T$ is an *(undisplaced uncurried) forward integral* of g with respect to h from b if and only if it satisfies the difference equation:

$$fjh = g(\mathbf{I}; f) \tag{27}$$

and $b = fi$. Also, if $r : S' \longrightarrow S$, then $f : S \longrightarrow T$ is a *displaced (uncurried) backward integral* of g with respect to r from b if and only if it satisfies the difference equation:

$$fj = g(\mathbf{I}; f)r \tag{28}$$

and $b = fi$. The corresponding curried integrals are defined in the obvious way.

Definition (Displaced Forward and Undisplaced Backward Integrals): Suppose $S = S_0 + S'$, $h : S \longrightarrow S'$, $g : S' \times T \longrightarrow T$ and $b : S_0 \longrightarrow T$ are given. Then $f : S \longrightarrow T$ is an *displaced (uncurried) forward integral* of g with respect to h from b if and only if it satisfies the difference equation:

$$fjh = g(h; f) \tag{29}$$

and $b = fi$. Also, if $r : S' \longrightarrow S$ then $f : S \longrightarrow T$ is a *(undisplaced uncurried) backward integral* of g with respect to r from b if and only if it satisfies the difference equation:

$$fj = g(\mathbf{I}; fr) \tag{30}$$

and $b = fi$. The corresponding curried integrals are defined in the obvious way.

Definition (Indefinite Integrals): The preceding definition describes the *definite* integrals. When we leave the boundary function b unspecified we have an *indefinite* integral.

Note: Indefinite integrals are not necessarily unique. For example both $\mathbf{I}$ and σ (the successor function) are backward integrals of $\sigma\rho$ with respect to σ^{-1} (the predecessor function), since:

$$\begin{aligned} \mathbf{I}j &= \sigma\sigma^{-1} = \sigma\rho(\mathbf{I}; \mathbf{I}\sigma^{-1}), \text{ and} \\ \sigma j &= \sigma\sigma\sigma^{-1} = \sigma\rho(\mathbf{I}; \sigma\sigma^{-1}). \end{aligned}$$

They have different boundary functions, however.

Theorem 42 (Relations Between Integrals):

1. *If $rh = \mathbf{I}$ and f is an undisplaced backward integral of g with respect to r, then f is also a displaced forward integral of g with respect to h.*

[7] Paige seems to have had similar terminological difficulties, as reflected in the titles of his 1981 and 1982 publications.

2. *If $hr = \mathbf{I}$ and f is a displaced forward integral of g with respect to h, then f is also an undisplaced backward integral of g with respect to r.*

3. *If $rh = \mathbf{I}$ and f is a displaced backward integral of g with respect to r, then f is also an undisplaced forward integral of g with respect to h.*

4. *If $hr = \mathbf{I}$ and f is an undisplaced forward integral of g with respect to h, then f is also a displaced backward integral of g with respect to r.*

Proof: Proposition (1) is proved as follows. Since f is an undisplaced backward integral of g with respect to r it satisfies the difference equation $fj = g(\mathbf{I}; fr)$. Compose h on the right of both sides of this equation:

$$\begin{aligned} fjh &= g(\mathbf{I}; fr)h \\ &= g(h; frh) \\ &= g(h; f) \end{aligned}$$

Thus f satisfies the displaced forward difference equation and is thus a displaced forward integral of g with respect to h. Propositions (2) – (4) are proved analogously. □

Corollary 24 (Integrals With Respect to Bijections):

1. *If h is bijective, then f is a displaced forward integral of g with respect to h if and only if f is an undisplaced backward integral of g with respect to h^{-1}.*

2. *If h is bijective, then f is an undisplaced forward integral of g with respect to h if and only if f is a displaced backward integral of g with respect to h^{-1}.*

Proof: This follows from the theorem since if h is bijective then $r = h^{-1}$ satisfies $rh = \mathbf{I} = hr$. □

Theorem 43 (Fundamental Theorem: Difference of Integral):

1. *If f is an undisplaced forward integral of g with respect to h, then $\delta_h f \doteq g$.*

2. *If f is an undisplaced backward integral of g with respect to r, then $dub_r f \doteq g$.*

3. *Analogous results holds for the displaced integrals and for the curried integrals of all kinds.*

Proof: Consider proposition (1). If f is a forward integral of g with respect to h then by definition it satisfies the difference equation $fjh = g(\mathbf{I}; f)$. But then by Eq. 14 the general form of the difference is $\delta_h f \doteq g$. The other propositions are proved in the same way. □

Theorem 44 (Fundamental Theorem: Integral of Difference):

Table 2: Notation for Integrals

Mode	Uncurried	Curried
undisplaced forward	$\int_b g\delta h$	$\int_b G\Delta h$
displaced forward	$\int_b g\bar{\delta} h$	$\int_b G\bar{\Delta} h$
undisplaced backward	$\int_b g\partial r$	$\int_b G\nabla r$
displaced backward	$\int_b g\bar{\partial} r$	$\int_b G\bar{\nabla} r$

1. *f is an undisplaced forward integral of $\delta_h f$ with respect to h from fi.*

2. *f is an undisplaced backward integral of $dub_r f$ with respect to r from fi.*

3. *Analogous results holds for the displaced integrals and for the curried integrals of all kinds.*

Proof: Consider proposition (a). By definition the forward difference satisfies the difference equation $fjh = \delta_h f(\mathbf{I}; f)$. But by the definition of the integral, this means that f is a forward integral of $\delta_h f$ with respect to h from fi. The other propositions are proved in the same way. □

Notation (Integral Signs): If the integrals exist and are unique, then they are written as in Table 2

Theorem 45 (Fundamental Theorem: Difference of Integral): *If the integral exists and is unique, then*

$$\delta_h \int_b g\delta h \doteq g \tag{31}$$

$$\partial_r \int_b g\partial r \doteq g \tag{32}$$

$$\bar{\delta}_h \int_b g\bar{\delta} h \doteq g \tag{33}$$

$$\bar{\partial}_r \int_b g\bar{\partial} r \doteq g \tag{34}$$

Analogous results hold for the curried integrals and differences.

Proof: The uniqueness of the integral allows Thm. 43 to be restated with the integral notation. □

Note: The fact that Eqs. 31 has '$\doteq$' rather than simply '$=$' reflects the fact that the integration operation makes no use of the behavior of g for objects not in its kernel. Hence, the integration operation "forgets" g's behavior on these objects, and so g cannot be completely recovered by differencing.

3.3 Existence and Computability of Integrals

The following theorems establish sufficient, but not necessary, conditions for the existence of functional integrals. We present these results for the uncurried integrals only; similar results apply to the curried integrals.

Definition (Recursive Discriminability): A direct sum $S = T + U$ is *recursively discriminable* if and only if the discriminators `1st`? and `2nd`? are computable.

In general it will be seen that reducibility of the domain is a sufficient condition for the *existence* of the integral, and its being recursively discriminable is sufficient for the *computability* of the integral. The integrals may also exist for nonreducible domains (e.g., domains of infinite sequences), but the proofs have not been completed.

Theorem 46 (Existence of Undisplaced Backward Integral): *Suppose we have $S = S_0 + S'$, $b : S_0 \longrightarrow T$, $g : S' \times T \longrightarrow T$ and $r : S' \longrightarrow S$, and that r reduces S. Then:*

$$\int_b g\partial r = \phi(b, g, r)$$

Proof: Cor. 21 shows that $f = \phi(b, g, r)$ is the unique solution extending b to $fj = g(\mathbf{I}; fr)$. $\square$

Corollary 25 (Computability of Undisplaced Backward Integral): *If b, g and r are computable, S is recursively discriminable, and r reduces S, then $\int_b g\partial r$ is computable.*

Proof: This follows from the equation

$$\int_b g\partial r = \phi(b, g, r)$$

Note that the recursive discriminability of the domain implies that $|x|$ is computable. $\square$

Corollary 26 (Fundamental Theorem: Undisplaced Backward Integral of Difference): *If r reduces S, then*

$$\int_{fi} \partial_r f \partial r = f$$

Proof: Since $fj = \partial_r f(\mathbf{I}; fr)$, by Cor. 21 we know $f = \phi(fi, \partial_r f, r)$. By the theorem this is $\int_{fi} \partial_r f \partial r$. $\square$

Theorem 47 (Existence of Displaced Forward Integral): *If h is bijective and h^{-1} reduces S, then*

$$\int_b g\bar{\delta} h = \int_b g\partial h^{-1}$$

Proof: By Cor. 24 every displaced forward integral is an undisplaced backward integral and vice versa. But by Thm. 46 the undisplaced backward integral is unique, therefore so is the displaced forward. $\Box$

Corollary 27 (Computability of Displaced Forward Integral): *Suppose h is bijective, S is recursively discriminable and h^{-1} reduces S. Then $\int_b g\bar{\delta}h$ is computable if b, g and h^{-1} are computable.*

Corollary 28 (Fundamental Theorem: Displaced Forward Integral of Difference): *If h is bijective, and h^{-1} reduces S then*

$$\int_{fi} \bar{\delta}_h f \bar{\delta} h = f$$

Proof: Under the hypothesis:

$$\begin{aligned}
\int_{fi} \bar{\delta}_h f \bar{\delta} h &= \int_{fi} \partial_{h^{-1}} f \bar{\delta} h, \text{ by Thm. 10} \\
&= \int_{fi} \partial_{h^{-1}} f \partial h^{-1}, \text{ by Thm. 47} \\
&= f, \text{ by Cor. 26}
\end{aligned}$$

$\Box$

Theorem 48 (Existence of Displaced Backward Integral): *Suppose we have $S = S_0 + S'$, $b : S_0 \longrightarrow T$, $g : S \times T \longrightarrow T$ and $r : S' \longrightarrow S$ and that r reduces S. Then*

$$\int_b g \bar{\partial} r = \phi[b, g(r \times \mathbf{I}), r]$$

Corollary 29 (Computability of Displaced Backward Integral): *If b, g and r are computable, S is recursively discriminable, and r reduces S, then $\int_b g\bar{\partial}r$ is computable.*

Corollary 30 (Fundamental Theorem: Displaced Backward Integral of Difference): *If r reduces S, then*

$$\int_{fi} \bar{\partial}_r f \bar{\partial} r = f$$

Theorem 49 (Existence of Undisplaced Forward Integral): *If h is bijective and h^{-1} reduces S, then*

$$\int_b g\delta h = \int_b g \bar{\partial} h^{-1}$$

Corollary 31 (Computability of Undisplaced Forward Integral): *Suppose h is bijective and h^{-1} reduces S. Then $\int_b g\delta h$ is computable if b, g and h^{-1} are computable.*

Corollary 32 (Fundamental Theorem: Undisplaced Forward Integral of Difference): *If h is bijective, and h^{-1} reduces S then*

$$\int_{fi} \delta_h f \delta h = f$$

3.4 Fixed Differences and Integrals

Consider the difference equation:

$$(fjh)_x = (G_x f)_x \tag{35}$$

If G_x does not depend on x, that is, $G_x = \gamma$ for some fixed $\gamma : T \longrightarrow T$, then f is said to have a *fixed* (i.e., nonvarying) difference γ. In this section we explore the properties of fixed differences and their integrals. We have previously investigated fixed differences (1986, 1987), but here we treat them as a special case of the general (variable) differences.

Theorem 50 (Uncurried Fixed Differences): *If $fjh = \gamma f$ then*

$$\delta_h f \doteq \gamma\rho, \quad \bar{\delta}_h f \doteq \gamma\rho$$

If $fj = \gamma fr$ then

$$\partial_r f \doteq \gamma\rho, \quad \bar{\partial}_r f \doteq \gamma\rho$$

Note that the kernels of the differences are not the same, so we cannot conclude for example that $\delta_h f \doteq \bar{\delta}_h f$.

Corollary 33 : *If $rh = \mathbf{I}$ and either $fjh = \gamma f$ or $fj = \gamma fr$, then*

$$\delta_h f \doteq \gamma\rho \doteq \bar{\partial}_r f$$

Proof: In this case the general forms of the differences are equal, since r is surjective and so $\mathrm{Im}(\mathbf{I}; f) = \mathrm{Im}(\mathbf{I}; f)r$. □

We do not have a general form for integrals, but we do have the following characterization of a family of solutions to the difference equation.

Theorem 51 (Families of Fixed Forward Integrals): *If f_0 is a solution of the (fixed) forward difference equation $fh = \gamma f$, then so are $\gamma^k f_0$ and $f_0 h^k$ for all $k \geq 0$. If either γ or h is bijective, then the corresponding results hold also for $k < 0$.*

Proof: We must show $fh = \gamma\rho(\mathbf{I}; f)$ in each case. In the first case let $f = \gamma^k f_0$ and derive:

$$\gamma\rho(\mathbf{I}; f) = \gamma f = \gamma(\gamma^k f_0) = \gamma^k(\gamma f_0) = \gamma^k f_0 h = fh$$

In the second case let $f = f_0 h^k$ and derive:

$$\gamma\rho(\mathbf{I}; f) = \gamma f = (\gamma f_0)h^k = (f_0 h)h^k = (f_0 h^k)h = fh$$

□

Theorem 52 (Families of Fixed Backward Integrals): *If f_0 is a solution of the (fixed) backward difference equation $f = \gamma fr$, then so are $\gamma^k f_0$ and $f_0 r^k$ for all $k \geq 0$. If either γ or r is bijective, then the corresponding results hold also for $k < 0$.*

Proof: We must show $f = \gamma\rho(\mathbf{I}; fr)$. In the first case let $f = \gamma^k f_0$ and derive:

$$\gamma\rho(\mathbf{I}; fr) = \gamma fr = \gamma(\gamma^k f_0)r = \gamma^k(\gamma f_0 r) = \gamma^k f_0 = f$$

In the second case let $f = f_0 r^k$ and derive:

$$\gamma\rho(\mathbf{I}; fr) = \gamma fr = \gamma(f_0 r^k)r = (\gamma f_0 r)r^k = f_0 r^k = f$$

□

Note: The following two theorems define families of integrals for the case where the integrand has a fixed partial difference (with respect to its second argument).

Theorem 53 (Characterization of a Family of Forward Integrals): *Suppose f_0 is any solution of the forward difference equation $fh = g(\mathbf{I}; f)$ and that $\delta_{(\mathbf{I}\times k)}g \doteq k\rho$. Then every function of the form $f = kf_0$ is a solution.*

Proof: Derive:

$$\begin{aligned} g(\mathbf{I}; f) &= g(\mathbf{I}; kf_0) \\ &= g(\mathbf{I} \times k)(\mathbf{I}; f_0) \\ &= k\rho(\mathbf{I}; g)(\mathbf{I}; f_0) \\ &= kg(\mathbf{I}; f_0) \\ &= kf_0 \\ &= f \end{aligned}$$

□

Theorem 54 (Characterization of a Family of Backward Integrals): *Suppose f_0 is any solution of the backward difference equation $f = g(\mathbf{I}; fr)$ and that $\delta_{(\mathbf{I}\times k)}g \doteq k\rho$. Then every function of the form $f = kf_0$ is a solution.*

Proof: Derive:

$$\begin{aligned} g(\mathbf{I}; fr) &= g(\mathbf{I}; kf_0 r) \\ &= g(\mathbf{I} \times k)(\mathbf{I}; f_0 r) \\ &= k\rho(\mathbf{I}; g)(\mathbf{I}; f_0 r) \\ &= kg(\mathbf{I}; f_0 r) \\ &= kf_0 \\ &= f \end{aligned}$$

□

Theorem 55 (Recursion Based on Fixed Differences): *If $fj = \gamma fr$ and $b = fi$, then*

$$f = b \mid \gamma fr$$

Corollary 34 : *If $hr = I$, $b = fi$, and $fjh = \gamma f$, then*

$$f = b \mid \gamma fr$$

Theorem 56 (Expansion by Fixed Differences):

1. *If $fj = \gamma fr$, and r reduces S, then*

$$f_x = \gamma^{|x|} b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|} : x \quad \textit{where} \quad b = fi$$

2. *If $fjh = \gamma f$, $rh = \mathbf{I}$, and r reduces S, then*

$$f_x = \gamma^{|x|} b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|} : x \quad \textit{where} \quad b = fi$$

Proof: For proposition (a), apply Thm. 41 and note

$$\begin{aligned}
[\Phi(b, \kappa_\gamma, r)]_x &= \left\{ \prod_{k=1}^{|x|} [(\kappa_{\mathbf{I}} \mid \kappa_\gamma)(i \mid r)^{k-1}]_x \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}]_x \\
&= \left(\prod_{k=1}^{|x|} \gamma \right) : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}]_x, \text{ since } (i \mid r)^{k-1} : x \in S' \\
&= \gamma^{|x|} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}]_x
\end{aligned}$$

□

Corollary 35 : *If r reduces S, then $\int_b \gamma\rho\partial r : x = \gamma^{|x|} b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|} : x$.*

Corollary 36 : *If $fj = fr$ and r reduces S then*

$$f_x = fi(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|} : x.$$

Corollary 37 : *If $f = i \mid fr$ and r reduces S then $f_x = x_0$ where $x_0 = (i \mid r)^{|x|} : x$.*

Corollary 38 : *If r reduces S and $S_0 = \{x_0\}$, then:*

1. $\int_b \gamma\rho\partial r : x = \gamma^{|x|} b : x_0$
2. *If in addition $rh = \mathbf{I}$ then $\int_b \gamma\rho\delta h : x = \gamma^{|x|} b : x_0$*

Corollary 39 : *If $fjh_k = \gamma_k f$ for $1 \leq k \leq n$, then $fh_1 \cdots h_n = \gamma_1 \cdots \gamma_n f$.*

4 Some Concrete Differences and Integrals

In this section we make the preceding ideas more concrete by showing the differences and integrals of a few simple functions.

4.1 Arithmetic Operations on Natural Numbers

If we let $S = \mathbf{N}$, the natural numbers, then $S_0 = \{0\}$ and $S' = \{1, 2, \ldots\}$.

Notation (Successor and Predecessor): For convenience we represent the successor and predecessor operations:

$$\begin{aligned} \sigma_x &= \sigma : x = x + 1 \\ \sigma_x^{-1} &= \sigma^{-1} : x = x - 1 \end{aligned}$$

Clearly, σ^{-1} reduces $\mathbf{N}$ and $\mathbf{N}$ is recursively discriminable.

Theorem 57 (Curried Addition): *Represent addition in curried form $\Sigma : m : n = m + n$. This operator satisfies the following equations:*

$$\begin{aligned} \delta_\sigma \Sigma_m &\doteq \sigma\rho, \quad \partial_\sigma^{-1}\Sigma_m \doteq \sigma\rho, \quad \bar{\delta}_\sigma \Sigma_m \doteq \sigma\rho, \quad \bar{\partial}_\sigma^{-1}\Sigma_m \doteq \sigma\rho \\ \Sigma_m &= \int_{0 \mapsto m} \sigma\rho\delta\sigma = \int_{0 \mapsto m} \sigma\rho\partial\sigma^{-1} = \int_{0 \mapsto m} \sigma\rho\bar{\partial}\sigma^{-1} = \int_{0 \mapsto m} \sigma\rho\delta\sigma \end{aligned}$$

where $0 \mapsto m$ denotes the function defined on 0 that maps 0 into m.

Proof: Note that the functions satisfy the following difference equations:

$$\begin{aligned} \Sigma_m j \sigma &= \sigma \Sigma_m \\ \Sigma_m j &= \sigma \Sigma_m \sigma^{-1} \end{aligned}$$

Hence we can apply Thm. 50 to get the differences. Since σ^{-1} reduces $\mathbf{N}$ to 0, we know that the integrals exist and are unique. □

Theorem 58 (Uncurried Addition): *Let* sum *be the uncurried addition operation,* $\mathsf{sum} : (m, n) = m + n$*. Then the following results hold*

$$\begin{aligned} \delta_{\sigma \times \mathbf{I}} \mathsf{sum} &\doteq \sigma\rho, \ etc. \\ \delta_{\mathbf{I} \times \sigma} \mathsf{sum} &\doteq \sigma\rho, \ etc. \\ \mathsf{sum} &= \int_{(0,n) \mapsto n} \sigma\rho\delta(\sigma \times \mathbf{I}), \ etc. \\ \mathsf{sum} &= \int_{(m,0) \mapsto m} \sigma\rho\delta(\mathbf{I} \times \sigma), \ etc. \end{aligned}$$

In each case 'etc.' means that the differences all have the same form, although with different kernels. (The backward differences are of course with respect to $\mathbf{I} \times \sigma^{-1}$ and $\sigma^{-1} \times \mathbf{I}$.)

Theorem 59 (Curried Multiplication): *Represent multiplication in curried form $\Pi : m : n = m \times n$. It satisfies:*

$$\begin{aligned} \delta_\sigma \Pi_m &\doteq \Sigma_m \rho, \ \text{etc.} \\ \Pi_m &= \int_{0 \mapsto 0} \Sigma_m \rho\delta\sigma, \ \text{etc.} \end{aligned}$$

Theorem 60 (Uncurried Multiplication): *Represent multiplication in uncurried form,* $\mathsf{prod} : (m,n) = mn$. *Then,*

$$
\begin{aligned}
\delta_{\sigma\times\mathbf{I}}\mathsf{prod} &\doteq \mathsf{sum}(\rho\times\mathbf{I}) \\
\delta_{\mathbf{I}\times\sigma}\mathsf{prod} &\doteq \mathsf{sum}(\lambda\times\mathbf{I}) \\
\partial_{\sigma^{-1}\times\mathbf{I}}\mathsf{prod} &\doteq \mathsf{sum}(\rho\times\mathbf{I}) \\
\partial_{\mathbf{I}\times\sigma^{-1}}\mathsf{prod} &\doteq \mathsf{sum}(\lambda\times\mathbf{I}) \\
\mathsf{prod} &= \int_{(0,n)\mapsto 0}\mathsf{sum}(\rho\times\mathbf{I})\delta(\sigma\times\mathbf{I}) \\
\mathsf{prod} &= \int_{(m,0)\mapsto 0}\mathsf{sum}(\lambda\times\mathbf{I})\delta(\mathbf{I}\times\sigma) \\
\mathsf{prod} &= \int_{(0,n)\mapsto 0}\mathsf{sum}(\rho\times\mathbf{I})\partial(\sigma^{-1}\times\mathbf{I}) \\
\mathsf{prod} &= \int_{(m,0)\mapsto 0}\mathsf{sum}(\lambda\times\mathbf{I})\partial(\mathbf{I}\times\sigma^{-1})
\end{aligned}
$$

Note that the differences of prod are not fixed. Here '$(0,n)\mapsto 0$' represents the function κ_0 on $\{0\}\times\mathbf{N}$ and $(m,0)\mapsto 0$ represents the function κ_0 on $\mathbf{N}\times\{0\}$.

Theorem 61 (Curried Exponentiation): *Represent exponentiation in curried form* $\mathsf{E} : m : n = m^n$. *It satisfies:*

$$
\begin{aligned}
\delta_\sigma\mathsf{E}_m &\doteq \Pi_m\rho, \text{ etc.} \\
\mathsf{E}_m &= \int_{0\mapsto 1}\Pi_m\rho\delta\sigma, \text{ etc.}
\end{aligned}
$$

Notice how these operations form a hierarchy of integrals. For example:

$$\mathsf{E}_m = \int_{0\mapsto 1}\int_{0\mapsto 0}\int_{0\mapsto m}\sigma\rho\delta\sigma\circ\rho\delta\sigma\circ\rho\delta\sigma$$

Alternately,

$$\mathsf{E}_m = \int_{0\mapsto 1}\int_{0\mapsto 0}\int_{0\mapsto m}\sigma\rho\partial\sigma^{-1}\circ\rho\partial\sigma^{-1}\circ\rho\partial\sigma^{-1}$$

4.2 Sequence Processing Operations

Note that a sequence, like a LISP list, is just a nesting of pairs:

$$(x_1,x_2,\ldots,x_n) = (x_1,(x_2,\ldots,x_n))$$

We assume $x_n = \mathsf{nil}$, always. Therefore λ and ρ select the "first" (`car`) and "rest" (`cdr`) of sequences:

$$
\begin{aligned}
\lambda : (x_1,x_2,\ldots,x_n) &= \lambda : (x_1,(x_2,\ldots,x_n)) = x_1 \\
\rho : (x_1,x_2,\ldots,x_n) &= \rho : (x_1,(x_2,\ldots,x_n)) = (x_2,\ldots,x_n)
\end{aligned}
$$

Also, π_x prefixes x onto a sequence:

$$\pi_x : (x_1, \ldots, x_n) = (x, (x_1, \ldots, x_n)) = (x, x_1, \ldots, x_n)$$

Thus, for sequences, $S_0 = \{\mathsf{nil}\}$ and $S' = T \times S$, where T is the element type of the sequences. The type of sequences is recursively discriminable and reducible by ρ.

Theorem 62 (Length):

$$\begin{aligned}
\delta_{\pi_x}\mathsf{length} &\doteq \sigma\rho \doteq \bar{\partial}_\rho \mathsf{length} \\
\partial_\rho \mathsf{length} &\doteq \sigma\rho \\
\bar{\delta}_{\pi_x}\mathsf{length} &\doteq \sigma\rho \\
\mathsf{length} &= \int_{\mathsf{nil} \mapsto 0} \sigma\rho\partial\rho, \text{ etc.}
\end{aligned}$$

Proof: Apply Thm. 50 and Cor. 33. □

Corollary 40 :

$$\mathsf{length}\, \pi_{x_1} \cdots \pi_{x_n} = \sigma^n \mathsf{length}$$

Proof: This follows from Cor. 38(b). □

Definition (Elt): The application $\mathsf{elt} : (S, n)$ returns the n-th element of S; it satisfies the equations:

$$\begin{aligned}
\mathsf{elt} : (S, 1) &= \lambda : S \\
\mathsf{elt} : (S, n) &= \mathsf{elt} : (\rho : S, n-1), \ n > 1
\end{aligned}$$

Theorem 63 (Elt):

$$\begin{aligned}
\partial_{(\rho \times \sigma^{-1})}\mathsf{elt} &\doteq \rho \\
\mathsf{elt} &= \int_{(S,1) \mapsto \lambda : S} \rho\partial(\rho \times \sigma^{-1})
\end{aligned}$$

Since the differences are fixed, they are all equal, as are the integrals. (Here $\gamma = \mathbf{I}$).

Corollary 41 :

$$\mathsf{elt} : (S, n) = [i \mid (\rho \times \sigma^{-1})]^{n-1} : (S, n)$$

Proof: This follows from Cor. 37 and the observation that $|(S, n)| = n - 1$. □

Definition (Cat): The application $\mathsf{cat} : (S, T)$ catenates the sequences S and T; it satisfies:

$$\begin{aligned}
\mathsf{cat} : (\mathsf{nil}, T) &= T \\
\mathsf{cat} : (S, T) &= [\lambda : S, \mathsf{cat} : (\rho : S, T)], \ S \neq \mathsf{nil}
\end{aligned}$$

Theorem 64 (Cat):

$$\begin{aligned}
\delta_{(\pi_x \times \mathbf{I})}\mathsf{cat} &\doteq \pi_x\rho \\
\bar{\delta}_{(\pi_x \times \mathbf{I})}\mathsf{cat} &\doteq \pi_x\rho \\
\partial_{(\rho \times \mathbf{I})}\mathsf{cat} &\doteq \lambda^2 \times \mathbf{I} \\
\mathsf{cat} &= \int_{(\mathsf{nil},T)\mapsto T} (\lambda^2 \times \mathbf{I})\partial(\rho \times \mathbf{I})
\end{aligned}$$

Note that $\bar{\partial}_{(\rho\times\mathbf{I})}\mathsf{cat}$ *does not exist.*

Proof: This follows from the difference equations:

$$\begin{aligned}
\mathsf{cat}^j &= (\lambda^2 \times \mathbf{I})[\mathbf{I}; \mathsf{cat}(\rho \times \mathbf{I})] \\
\mathsf{cat}^j(\pi_x \times \mathbf{I}) &= \pi_x\rho(\mathbf{I}; \mathsf{cat})
\end{aligned}$$

which in turn follow from the definition of cat. For example, for $S \neq \mathsf{nil}$,

$$\begin{aligned}
\mathsf{cat} : (S, T) &= [\lambda : S, \mathsf{cat} : (\rho : S, T)] \\
&= [\lambda^2 : (S, T), \mathsf{cat}(\rho \times \mathbf{I}) : (S, T)] \\
&= [\lambda^2; \mathsf{cat}(\rho \times \mathbf{I})] : (S, T)
\end{aligned}$$

Hence,

$$\mathsf{cat}^j = [\lambda^2; \mathsf{cat}(\rho \times \mathbf{I})] = (\lambda^2 \times \mathbf{I})[\mathbf{I}; \mathsf{cat}(\rho \times \mathbf{I})]$$

□

Corollary 42 :

$$\mathsf{cat}(\pi_{x_1} \cdots \pi_{x_n} \times \mathbf{I}) = \pi_{x_1} \cdots \pi_{x_n}\mathsf{cat}$$

Proof: The proof makes use of Cor. 39 and the following simple identity:

$$\left(\prod_{k=1}^{n} \pi_{x_k}\right) \times \mathbf{I} = \prod_{k=1}^{n} (\pi_{x_k} \times \mathbf{I})$$

□

Definition (Accumulation From the Right):

$$\mathsf{accr}(f, i) : S = f : \{S_1, \ldots, f : [S_{n-1}, f : (S_n, i)] \ldots\},\ n \geq 0$$

Theorem 65 (Accumulation From the Right):

$$\begin{aligned}
\delta_{\pi_x}\mathsf{accr}(f, i) &= f\pi_x\rho \\
\partial_\rho\mathsf{accr}(f, i) &\doteq f(\lambda \times \mathbf{I}) \\
\mathsf{accr}(f, i) &= \int_{\mathsf{nil}\mapsto i} f(\lambda \times \mathbf{I})\partial\rho
\end{aligned}$$

(This operation is sometimes called folding; *see MacLennan (1990) for a discussion of accumulation and reduction.)*

Proof: Follows from the difference equations:

$$\begin{aligned}\mathsf{accr}(f,i)j\pi_x &= f\pi_x\rho[\mathbf{I};\mathsf{accr}(f,i)]\\ \mathsf{accr}(f,i)j &= f(\lambda\times\mathbf{I})[\mathbf{I};\mathsf{accr}(f,i)\rho]\end{aligned}$$

□

Definition (Reduction From the Right):

$$\mathsf{redr}f : S = f : \{S_1,\ldots,f : [S_n-2, f : (S_{n-1},S_n)]\ldots\},\ n>0$$

Corollary 43 (Reduction From the Right):

$$\begin{aligned}\delta_{\pi_x}\mathsf{redr}f &= f\pi_x\rho\\ \partial_\rho\mathsf{redr}f &\doteq f(\lambda\times\mathbf{I})\\ \mathsf{redr}f &= \int_{(x,\mathsf{nil})\mapsto x} f(\lambda\times\mathbf{I})\partial\rho\end{aligned}$$

Note: Reduction has the same differences as accumulation; they differ only in boundary conditions.

5 Conclusions

In this paper we have defined several functional difference operators, which reduce global changes to local changes, and the inverse operations of functional integration, which accumulate local changes into global changes. We have proved a number of fundamental results concerning the existence, uniqueness, computability and interrelationships of these operators. These operators can be used to improve our comprehension and derivation of functional programs.

It might be objected that, for real programs, the differences will be very complicated and the difference equations will be unsolvable. This may be true. Nevertheless, as is the case with differential equations in the natural sciences, we expect that the idea of a functional difference equation will prove illuminating even if the equation cannot be solved analytically. An answer to this question must await practical applications of this calculus.

6 Acknowledgements

This work was supported in part by the Office of Naval Research under contract N0001487WR-24037 to the Naval Postgraduate School, in part by the University of Tennessee under Faculty Research Awards and in part by the Mathematical Sciences Section of Oak Ridge National Laboratory.

7 References

Backus, John (1978). "Can Programming be Liberated from the von Neumann Style? A Functional Style and its Algebra of Programs," *Communications of the ACM,* Vol. 21, No. 8 (August 1978), pp. 613–641.

Backus, John (1981). "The Algebra of Functional Programs: Function Level Reasoning, Linear Equations, and Extended Definitions," *Formalization of Programming Concepts,* J. Diaz and I. Ramos (eds.), Berlin: Springer-Verlag Lecture Notes in Computer Science 107, pp. 1–43.

MacLennan, Bruce J. (1986). "Preliminary Investigation of a Calculus of Functional Differences: Fixed Differences," Naval Postgraduate School Computer Science Department Technical Report NPS52-86-010 (February 1986).

MacLennan, Bruce J. (1987). "An Algebraic Approach to a Calculus of Functional Differences: Fixed Differences and Integrals," Naval Postgraduate School Computer Science Department Technical Report NPS52-87-041 (September 1987).

MacLennan, Bruce J. (1990). *Functional Programming: Practice and Theory,* Reading: Addison-Wesley.

Paige, Robert A. (1981). *Formal Differentiation — A Program Synthesis Technique.* Ann Arbor: UMI Research Press.

Paige, Robert A., and Koenig, Shaye (1982). "Finite Differencing of Computable Expressions," *ACM Transactions on Programming Languages and Systems,* Vol. 4, No. 3 (July 1982), pp. 402–454.

APPENDICES

A General Definitions

1. $\mu = (\mathbf{I} \times \lambda); (\mathbf{I} \times \rho)$

2. $\nu = (\lambda^2; \lambda\rho); (\rho\lambda; \rho^2)$

3. $\tau(i \times i' \mid j \times j') = \mathbf{I}$

4. $\upsilon = (i \times \mathbf{I} \mid j \times \mathbf{I})^{-1}$

5. $r \uparrow_m^n = r_m r_{m+1} r_{m+2} \cdots r_{n-1} r_n = \prod_{k=m}^{n} r_k, \quad m \leq n$

6. $r \downarrow_m^n = r_n r_{n-1} r_{n-2} \cdots r_{m+1} r_m = \prod_{k=n}^{m} r_k, \quad m \leq n$

7. $|x| = \min\{n \geq 0 \mid \mathsf{1st}?[(i \mid r)^n : x]\}$

8. $\Phi(b, G, r)]_x = \left\{ \prod_{k=1}^{|x|} [(\kappa_{\mathbf{I}} \mid G)(i \mid r)^{k-1}]_x \right\} : [b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}]_x$

9. $\phi(b, g, r) : x = \rho \left(\prod_{k=1}^{|x|} \{\lambda; (\rho \mid g)\upsilon[(i \mid r)^{k-1} \times \mathbf{I}]\} \right) [\mathbf{I}; b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|}] : x$

Arithmetic Functions

10. $\sigma_x = \sigma : x = x + 1$

11. $\sigma_x^{-1} = \sigma^{-1} : x = x - 1$

12. $\Sigma : m : n = m + n$

13. $\Pi : m : n = m \times n$

14. $\mathsf{prod} : (m, n) = m \times n$

15. $\mathsf{E} : m : n = m^n$

List Processing Functions

16. $\mathsf{length}(S_1, \ldots, S_n) \doteq n$

17. $\mathsf{elt}(S, k) \doteq S_k$

18. $\mathsf{cat}(S, T) \doteq (S_1, \ldots, S_m, T_1, \ldots, T_n)$

19. $\mathsf{accr}(f, i) : S = f : \{S_1, \ldots, f : [S_{n-1}, f : (S_n, i)] \ldots\},\ n \geq 0$

20. $\mathsf{redr} f : S = f : \{S_1, \ldots, f : [S_n - 2, f : (S_{n-1}, S_n)] \ldots\},\ n > 0$

B Table of Difference Equations

1. $fjh = \mathsf{S}(\Delta_h f) f$
2. $fjh = \delta_h f(\mathbf{I}; f)$
3. $fjh = \mathsf{S}(\bar{\Delta}_h f \circ h) f$
4. $fjh = \bar{\delta}_h f(h; f)$
5. $fj = \mathsf{S}(\nabla_r f)(fr)$
6. $fj = \partial_r f(\mathbf{I}; fr)$
7. $fj = [\mathsf{S}(\bar{\nabla}_r f) f] r$
8. $fj = \bar{\partial}_r f(\mathbf{I}; f) r$
9. $f = fi \mid \partial_r f(\mathbf{I}; fr)$
10. $f = fi \mid \bar{\partial}_r f(\mathbf{I}; f) r$, if r injective
11. $f = fi \mid \delta_h f(\mathbf{I}; f) r$, if $hr = \mathbf{I}$
12. $f = fi \mid \bar{\delta}_h f(\mathbf{I}; fr)$, if $hr = \mathbf{I}$

C Table of Functional Differences

Forward Differences

1. $\delta_h \int_b g \delta h \doteq g$
2. $\bar{\delta}_h \int_b g \bar{\delta} h \doteq g$
3. $\Delta_{\mathbf{I}} f \doteq \kappa_{\mathbf{I}}, \quad \delta_{\mathbf{I}} f \doteq \rho$
4. $[\Delta_h(f; g)]_x \doteq (\Delta_h f)_x \times (\Delta_h g)_x$
5. $\delta_h(f; g) \doteq (\delta_h f \times \delta_h g)\mu$
6. $[\Delta_{h \times k}(f \times g)]_{(x,y)} \doteq (\Delta_h f)_x \times (\Delta_k g)_y$

7. $\delta_{h\times k}(f\times g)\doteq(\delta_h f\times\delta_k g)\nu$

8. $(\Delta_h fg)_x\doteq[\Delta_{(\Delta_h g)_x}f]g:\,x$

9. $(\delta_h fg)\pi_x\doteq\delta_{\gamma_x}f(g\times\mathbf{I})\pi_x$ where $\gamma_x\doteq(\delta_h g)\pi_x$

10. $\delta_{h+k}(f+g)\doteq(\delta_h f+\delta_k g)\tau$

11. $\delta_{h+k}(f\mid g)\doteq(\delta_h f\mid\delta_k g)\upsilon$

12. $(\Delta_{hk}f)_x\doteq[(\Delta_h fi)k]_x(\Delta_k f)_x$

13. $\delta_{hk}f\doteq\delta_h fi(k\lambda;\delta_k f)$

14. $(\Delta_{h_1\cdots h_n}f)_x\doteq\prod_{k=1}^{n}[(\Delta_{h_k}fi_k)h\uparrow_{k+1}^{n}]_x$

15. $(\Delta_{h_n\cdots h_1}f)_x\doteq\prod_{k=n}^{1}[(\Delta_{h_k}fi_{k-1})h\downarrow_1^{k-1}]_x$

16. $\delta_{h_n\cdots h_1}f\doteq\rho\prod_{k=n}^{1}(h_k\lambda;\delta_{h_k}fi_{k-1})$

17. $(\Delta_{(jh)^n}f)_x\doteq\prod_{k=n}^{1}[(\Delta_h f)(jh)^{k-1}]_x$

18. $\delta_{(jh)^n}f\doteq\rho(jh\lambda;\delta_h f)^n$

Backward Differences

19. $\partial_r\int_b g\partial r\doteq g$

20. $\bar\partial_r\int_b g\bar\partial r\doteq g$

21. $\nabla_{\mathbf{I}}f\doteq\kappa_{\mathbf{I}},\quad\nabla_i f\doteq\kappa_{\mathbf{I}},\quad\nabla_j f\doteq\kappa_{\mathbf{I}}$

22. $\partial_{\mathbf{I}}f\doteq\rho,\quad\partial_i f\doteq\rho,\quad\partial_j f\doteq\rho$

23. $\partial_r(f;g)\doteq(\partial_r f\times\partial_r g)\mu$

24. $\partial_{r\times s}(f\times g)\doteq(\partial_r f\times\partial_s g)\nu$

25. $\nabla_{r|s}f\doteq\nabla_r f\mid\nabla_s f$

26. $\partial_{r|s}f\doteq(\partial_r f\mid\partial_s f)\upsilon$

27. $(\partial_r fg)\pi_x\doteq\delta_{\gamma_x}f(gr\times\mathbf{I})\pi_x$ where $\gamma_x\doteq(\partial_r g)\pi_x$

28. $\partial_{r+s}(f+g) \doteq (\partial_r f + \partial_s g)\tau$

29. $\partial_{r+s}(f \mid g) \doteq (\partial_r f \mid \partial_s g)\upsilon$

30. $\partial_{rs} f \doteq \partial_s f i[\lambda; \partial_r f(s \times \mathbf{I})]$

31. $(\nabla_{rs} f)_x \doteq (\nabla_s f i)_x[(\nabla_r f)s]_x$

32. $(\nabla_{r_n \cdots r_1} f)_x \doteq \prod_{k=1}^{n}[(\nabla_{r_k} f i_k) r \downarrow_1^{k-1}]_x$

33. $(\nabla_{r_1 \cdots r_n} f)_x \doteq \prod_{k=n}^{1}[(\nabla_{r_k} f i_{k-1}) r \uparrow_{k+1}^{n}]_x$

34. $\partial_{r_1 \cdots r_n} f \doteq \rho \prod_{k=n}^{1}[\lambda; \partial_{r_k} f i_{k-1}(r \uparrow_{k+1}^{n} \times \mathbf{I})]$

35. $(\nabla_{(i|r)^n} f)_x \doteq \prod_{k=1}^{n}[(\kappa_{\mathbf{I}} \mid \nabla_r f)(i \mid r)^{k-1}]_x$

36. $\partial_{(i|r)^n} f \doteq \rho \prod_{k=1}^{n}\{\lambda; (\rho \mid \partial_r f)\upsilon[(i \mid r)^{k-1} \times \mathbf{I}]\}$

37. $(\nabla_{r^n} f)_x \doteq \prod_{k=1}^{n}(\nabla_r f r^{k-1})_x$

38. $\partial_{r^n} f \doteq \rho \prod_{k=1}^{n}[\lambda; \partial_r f \upsilon(r^{k-1} \times \mathbf{I})]$

Relations Between Forward and Backward Differences

39. $\delta_h f = f j h \lambda \iota \mid c$

40. $\delta_h f \doteq \bar{\partial}_{h^{-1}} f$, if h bijective

41. $\bar{\delta}_h f \doteq \partial_{h^{-1}} f$, if h bijective

Differences of Arithmetic Functions

42. $\delta_\sigma \Sigma_m \doteq \sigma\rho$, $\partial_\sigma^{-1} \Sigma_m \doteq \sigma\rho$, $\bar{\delta}_\sigma \Sigma_m \doteq \sigma\rho$, $\bar{\partial}_\sigma^{-1} \Sigma_m \doteq \sigma\rho$

43. $\delta_{\sigma \times \mathbf{I}} \mathsf{sum} \doteq \sigma\rho$, etc.

44. $\delta_{\mathbf{I} \times \sigma} \mathsf{sum} \doteq \sigma\rho$, etc.

45. $\delta_\sigma \Pi_m \doteq \Sigma_m \rho$, etc.

46. $\delta_{\sigma \times \mathbf{I}} \mathsf{prod} \doteq \mathsf{sum}(\rho \times \mathbf{I})$

47. $\delta_{\mathbf{I} \times \sigma} \mathsf{prod} \doteq \mathsf{sum}(\lambda \times \mathbf{I})$

48. $\partial_{\sigma^{-1} \times \mathbf{I}} \mathsf{prod} \doteq \mathsf{sum}(\rho \times \mathbf{I})$

49. $\partial_{\mathbf{I} \times \sigma^{-1}} \mathsf{prod} \doteq \mathsf{sum}(\lambda \times \mathbf{I})$

50. $\delta_\sigma \mathsf{E}_m \doteq \Pi_m \rho,$ etc.

Differences of List Processing Functions

51. $\delta_{\pi_x}\mathsf{length} \doteq \sigma\rho$

52. $\bar{\delta}_{\pi_x}\mathsf{length} \doteq \sigma\rho$

53. $\bar{\partial}_{\rho}\mathsf{length} \doteq \sigma\rho$

54. $\partial_{\rho}\mathsf{length} \doteq \sigma\rho$

55. $\partial_{(\rho\times\sigma^{-1})}\mathsf{elt} \doteq \rho$

56. $\delta_{(\pi_x\times\mathbf{I})}\mathsf{cat} \doteq \pi_x\rho$

57. $\bar{\delta}_{(\pi_x\times\mathbf{I})}\mathsf{cat} \doteq \pi_x\rho$

58. $\partial_{(\rho\times\mathbf{I})}\mathsf{cat} \doteq \lambda^2 \times \mathbf{I}$

59. $\delta_{\pi_x}\mathsf{accr}(f,i) = f\pi_x\rho$

60. $\partial_{\rho}\mathsf{accr}(f,i) \doteq f(\lambda \times \mathbf{I})$

61. $\delta_{\pi_x}\mathsf{redr} f = f\pi_x\rho$

62. $\partial_{\rho}\mathsf{redr} f \doteq f(\lambda \times \mathbf{I})$

D Table of Expansion Formulas

1. $(fi_n h_n \cdots h_1)_x = \prod_{k=n}^{1} [(\Delta_{h_k} f i_{k-1}) h \downarrow_1^{k-1}]_x : f_x$

2. $fi_n h_n \cdots h_1 = \rho \left[\prod_{k=n}^{1} (h_k\lambda; \delta_{h_k} f i_{k-1}) \right] (\mathbf{I}; f)$

3. $f(jh)^n = \rho(jh\lambda; \delta_h f)^n (\mathbf{I}; f)$

4. $fi_n : x = \prod_{k=n}^{1} [(\nabla_{r_k} f i_{k-1}) r \uparrow_{k+1}^{n}]_x : (fr_1 \cdots r_n)_x$

5. $fi_n = \rho \left\{ \prod_{k=n}^{1} [\lambda; \partial_{r_k} f i_{k-1} (r \uparrow_{k+1}^{n} \times \mathbf{I})] \right\} (\mathbf{I}; fr_1 \cdots r_n)$

6. $f_x = \left\{ \prod_{k=1}^{n} [(\kappa_{\mathbf{I}} \mid \nabla_r f)(i \mid r)^{k-1}]_x \right\} : [f(i \mid r)^n]_x$

7. $f_x = \left\{ \prod_{k=1}^{n} (\nabla_r f r^{k-1})_x \right\} : (f r^n)_x$

8. $f = \rho \left(\prod_{k=1}^{n} \{\lambda; (\rho \mid \partial_r f) v[(i \mid r)^{k-1} \times \mathbf{I}]\} \right) [\mathbf{I}; f(i \mid r)^n]$

9. $f = \rho \left\{ \prod_{k=1}^{n} [\lambda; \partial_r f v(r^{k-1} \times \mathbf{I})] \right\} (\mathbf{I}; f r^n)$

E Table of Functional Integrals

In the following it is assumed that r reduces S and that h is bijective and h^{-1} reduces S.

1. $\int_{fi} \delta_h f \delta h = f$

2. $\int_{fi} \bar{\delta}_h f \bar{\delta} h = f$

3. $\int_{fi} \partial_r f \partial r = f$

4. $\int_{fi} \bar{\partial}_r f \bar{\partial} r = f$

5. $\int_b g \delta h = \int_b g \bar{\partial} h^{-1}$

6. $\int_b g \bar{\delta} h = \int_b g \partial h^{-1}$

7. $\int_b g \partial r = \phi(b, g, r)$

8. $\int_b g \bar{\partial} r = \phi[b, g(r \times \mathbf{I}), r]$

9. $\int_b \gamma \rho \partial r : x = \gamma^{|x|} b(\mathbf{I} \mid \kappa_\omega)(i \mid r)^{|x|} : x$

Arithmetic Functions

10. $\Sigma_m = \int_{0 \mapsto m} \sigma \rho \delta \sigma = \int_{0 \mapsto m} \sigma \rho \partial \sigma^{-1} = \int_{0 \mapsto m} \sigma \rho \bar{\partial} \sigma^{-1} = \int_{0 \mapsto m} \sigma \rho \delta \sigma$

11. $\mathsf{sum} = \int_{(0,n) \mapsto n} \sigma \rho \delta(\sigma \times \mathbf{I}),$ etc.

12. $\mathsf{sum} = \int_{(m,0) \mapsto m} \sigma \rho \delta(\mathbf{I} \times \sigma),$ etc.

13. $\Pi_m = \int_{0 \mapsto 0} \Sigma_m \rho \delta \sigma$, etc.

14. $\mathsf{prod} = \int_{(0,n) \mapsto 0} \mathsf{sum}(\rho \times \mathbf{I}) \delta(\sigma \times \mathbf{I})$

15. $\mathsf{prod} = \int_{(m,0) \mapsto 0} \mathsf{sum}(\lambda \times \mathbf{I}) \delta(\mathbf{I} \times \sigma)$

16. $\mathsf{prod} = \int_{(0,n) \mapsto 0} \mathsf{sum}(\rho \times \mathbf{I}) \partial(\sigma^{-1} \times \mathbf{I})$

17. $\mathsf{prod} = \int_{(m,0) \mapsto 0} \mathsf{sum}(\lambda \times \mathbf{I}) \partial(\mathbf{I} \times \sigma^{-1})$

18. $\mathsf{E}_m = \int_{0 \mapsto 1} \Pi_m \rho \delta \sigma$, etc.

List Processing Functions

19. $\mathsf{length} = \int_{\mathsf{nil} \mapsto 0} \sigma \rho \partial \rho$, etc.

20. $\mathsf{elt} = \int_{(S,1) \mapsto \lambda :\, S} \rho \partial(\rho \times \sigma^{-1})$

21. $\mathsf{cat} = \int_{(\mathsf{nil},T) \mapsto T} (\lambda^2 \times \mathbf{I}) \partial(\rho \times \mathbf{I})$

22. $\mathsf{accr}(f, i) = \int_{\mathsf{nil} \mapsto i} f(\lambda \times \mathbf{I}) \partial \rho$

23. $\mathsf{redr} f = \int_{(x,\mathsf{nil}) \mapsto x} f(\lambda \times \mathbf{I}) \partial \rho$