THE JOURNAL OF SYMBOLIC LOGIC
Volume 63, Number 2, June 1998

THE LOGIC OF RECURSIVE EQUATIONS

A. J. C. HURKENS, MONICA McARTHUR, YIANNIS N. MOSCHOVAKIS,
LAWRENCE S. MOSS, AND GLEN T. WHITNEY

Abstract. We study logical systems for reasoning about equations involving recursive definitions. In particular, we are interested in "propositional" fragments of the *functional language of recursion* FLR [18, 17], i.e., without the value passing or abstraction allowed in FLR. The "pure," propositional fragment FLR_0 turns out to coincide with the *iteration theories* of [1]. Our main focus here concerns the sharp contrast between the simple class of valid identities and the very complex consequence relation over several natural classes of models.

In [18, 17], Moschovakis introduces the language FLR to study general recursive definitions of the form

$$p(u) = f(u, p).$$

The functional f determines how to compute values of the function (or "program") p based perhaps on other values of p. A key special case consists of simple fixed-point equations

$$p = f(p),$$

in which the dependence of p on some "input" has been eliminated or suppressed. Therefore, we investigate here two "propositional" fragments of FLR: FLR_0 (first introduced as the language $\mathscr{L}$ in [19]) and ELR. Completeness questions for FLR_0 and ELR are already far from trivial, and the simpler language makes broader classes of models more accessible.

To briefly summarize, FLR_0 takes a primitive stock of variables and function symbols, and it forms functional terms in the usual way and *recursion terms* by the construction

$$B \texttt{ where } \{x_1 = A_1, \dots , x_n = A_n\}. \tag{1}$$

The variables x_i typically occur in the terms A_i, so in fact FLR_0 can express systems of simultaneous fixed-point equations. Definitions of this kind occur in logic and computer science in a wide variety of settings.

Of course, each time a term of the form (1) is used, one must be able to find well-defined semantics. (This is to avoid terms like $x \texttt{ where } \{x = 1 + x\}$ or $R \texttt{ where } \{x \in R \leftrightarrow x \notin R\}$.) The most common general way to provide semantics

Received October 4, 1996.

During the preparation of this paper, Whitney was supported by the Fannie and John Hertz Foundation and by a grant from the National Science Foundation.

0022-4812/98/6302-0009/$3.80

for fixed point terms is by appeal to some sort of result on fixed points of monotone maps on a directed-complete partial order (cpo). This covers many of the cases for computer science. On the other hand, fixed points arise in many other contexts, and for this foundational study we present a very general notion of structure for FLR_0. In particular, we wish to admit "intensional" models, in which distinct functions may take identical values at all points of a structure but be assigned different fixed points; this type of intensionality is at the core of many studies of concurrent computation.

Early attempts to understand and prove properties of recursive definitions in various specific contexts include [5, 6, 15, 13]. The general study of recursion equations has been pursued under several guises since then:[1] as *recursive applicative program schemes* [4], *μ-calculus* [2], and perhaps most notably as the *iteration theories* [10, 1] of Bloom and Ésik. The latter work builds on Lawvere's introduction of *algebraic theories* [14] in order to get categorical presentations of universal algebra, and Elgot's use of these in connection with flowchart schemes [7]. Thus, the relationship of this paper and [1] is roughly that between ordinary equational logic and algebraic theories. In particular, we will provide an explicit "dictionary" showing that the categories of FLR_0-structures and iteration theories are equivalent, so that answers to many basic questions can be read off from known results in the iteration theory context. On the other hand, we present new results and questions suggested by the logical formulation, and this presentation will hopefully make the subject more accessible to those with a mathematical logic background.

§1. Elementary formal language of recursion.

1.1. Syntax. Fix a countably infinite set $\{\mathsf{v}_1, \mathsf{v}_2, \mathsf{v}_3, \dots\}$ of variables. A **signature** τ is a ranked set of **function symbols**; in other words, each symbol f has an associated *arity*, the (nonnegative integer) number of formal arguments it will take. Write τ_n for the subset of τ consisting of n-ary symbols. The following induction defines the **terms** of the language $\mathrm{FLR}_0(\tau)$.

(1) Any variable x is a term by itself.

(2) $\mathsf{f}(E_1, \dots, E_n)$ is a term if $\mathsf{f} \in \tau_n$ and E_1 through E_n are terms.

(3) E_0 `where` $\{x_1 = E_1, \dots, x_n = E_n\}$ is a term for any distinct variables $x_1, \dots, x_n$ and any terms $E_0, \dots, E_n$.

Intuitively, the second clause corresponds to function composition, and the third clause gives syntax for the solution of systems of recursive equations. The following expression schematically summarizes the whole definition:

$$E := x \mid f(E_1, \dots, E_n) \mid E_0 \texttt{ where } \{x_1 = E_1, \dots, x_n = E_n\}.$$

Syntactic notions concerning FLR_0 are defined as usual, including closed and open terms, substitutions, free substitutions, and fresh variables. The `where` in clause 3 binds variables x_1 through x_n.

We fix some notational conventions. The plain symbol g will abbreviate $\mathsf{g}()$ for nullary function symbols (i.e., constant symbols). The special symbol $\perp$ will

[1] Another approach to axiomatic theories of recursion, *algebraic recursion theory* [21, 11, 12], has a different emphasis: it confines itself to least (or initial) fixed points, and seeks to understand the additional combinatorial properties needed to support stronger results analogous to classical recursion theory.

abbreviate the term x `where` $\{x = x\}$ (the exact choice of variable is irrelevant). Informal vector notation will be used throughout, e.g., E `where` $\{\vec{x} = \vec{A}\}$ abbreviates E `where` $\{x_1 = A_1, \ldots, x_n = A_n\}$. When s is a function from variables to terms, we write $E[s]$ for the result of substituting the term $s(x)$ for the free occurrences of x in E, for each x in the domain of s. Sometimes the substitution s may be displayed explicitly, e.g., $E[\vec{M}/\vec{x}]$ denotes the result of substituting the FLR$_0$ term M_i for x_i, for each x_i in the sequence $\vec{x}$. Further, if the term E has been written as $E(x_1, \ldots, x_n)$, displaying (some of) its free variables, then the substitution $E[\vec{M}/\vec{x}]$ may also be written $E(M_1, \ldots, M_n)$. Finally, $A \equiv B$ means that the expressions A and B are identical.

Alphabetic variants. Suppose $\sim$ is a relation on some variables. Define $A \equiv_\sim B$ to mean that A and B are syntactically identical up to $\sim$, i.e., that there is some $C(z_1, \ldots, z_n)$ and lists of variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ (which may have repetitions) such that $A \equiv C(x_1, \ldots, x_n)$, $B \equiv C(y_1, \ldots, y_n)$, and $x_i \sim y_i$ for all i.

Formulas. If A and B are terms, then for any sequence of distinct variables $\vec{x}$ (including the empty sequence), $\forall\vec{x}\,(A = B)$ is a **formula** of FLR$_0(\tau)$. Intuitively, the formulas will be used to express equations, such as $f(x) = g(y)$, which might hold for some particular x and y, and identities, like $\forall x\,(f(x) = f(f(x)))$, expressing the idempotence of f. This $\forall\vec{x}$ is another variable-binding operator: all occurrences of variables from $\vec{x}$ which are free in A and B are bound in $\forall\vec{x}\,(A = B)$, and all other occurrences remain free or bound as they were. A closed formula will be known as an **identity**. Substitutions apply to formulas as a whole; $\forall\vec{x}\,(A = B)[\vec{M}/\vec{y}]$ replaces free occurrences of the variables in the list $\vec{y}$ wherever they occur in A or B, but of course this substitution will not replace any occurrences of variables from $\vec{x}$. If $\phi \equiv \forall\vec{x}\,(A = B)$ is a formula, then $\forall y\,(\phi)$ is an abbreviation for $\forall y, \vec{x}\,(A = B)$, so that it makes sense to quantify any formula universally.

1.2. Poset semantics for FLR$_0$. Interpreting the variables of FLR$_0$ as ranging over the elements of a poset D leads to three natural classes of semantic structures which will be central throughout this paper. The full definitions follow, but briefly the classes are **Cont**, in which D is complete and the functions are continuous; **Mon**, in which D is complete and the functions are merely monotone; and **Wk**, the "weak" structures in which the poset may not be complete but nevertheless still contains "enough" fixed points to interpret FLR$_0$.[2]

First, let D be (directed-)complete. Given a signature τ, choose a monotone function $\mathsf{f}_\Lambda\colon D^n \to D$ for each f in τ_n. Now assign a **denotation**

$$\Lambda(\vec{x})\,E\colon D^n \to D$$

to each FLR$_0(\tau)$ term E and list of *distinct* variables $\vec{x} = x_1, \ldots, x_n$ containing the free variables of E, via the following induction:

(1) $\Lambda(x_1, \ldots, x_n)\,x_i$ is the usual ith projection from D^n to D.

(2) If $\Lambda(\vec{x})\,E_i = g_i$, then $\Lambda(\vec{x})\,\mathsf{f}(E_1, \ldots, E_m)$ is the function which takes $\vec{d} \in D^n$ to $\mathsf{f}_\Lambda(g_1(\vec{d}), \ldots, g_m(\vec{d}))$, i.e., the composition of the m-ary f_Λ with g_1 through g_m.

(3) Suppose that $E \equiv E_0$ `where` $\{y_1 = E_1, \ldots, y_m = E_m\}$. To compute $(\Lambda(\vec{x})E)(\vec{d})$, first suppose that none of the x_j happen to coincide with any of the

[2] These classes are not themselves new; indeed, **Cont** is the standard environment of "domain theory," and in [9], **Mon** is called $\mathscr{M}$ and **Wk** is the class of "concrete Park theories."

y_i. Let $f_i = \Lambda(y_1, \ldots, y_m, \vec{x})\, E_i$ for i from 0 to m. The functions f_1 through f_m form a system of m functions from D^{m+n} to D, which will in fact all be monotone. (This fact is proved by an easy simultaneous induction with this definition.) Thus, by a standard theorem on least fixed points (such as the Knaster-Tarski Theorem) there will be least elements $c_1, \ldots, c_m$ in D satisfying

$$\begin{array}{rcl} c_1 & = & f_1(\vec{c}, \vec{d}) \\ \vdots & & \vdots \\ c_m & = & f_m(\vec{c}, \vec{d}). \end{array} \tag{2}$$

These least fixed points are obtained by iterating f_1 through f_m (perhaps transfinitely), starting from the least element of D. Then for $\vec{d} \in D^n$, set

$$(\Lambda(\vec{x})E)(\vec{d}) = f_0(c_1, \ldots, c_m, \vec{d}).$$

If some of the y_i do occur in $\vec{x}$, choose a list of variables $\vec{x}\,'$ in which each y_i has been replaced by a variable fresh for the E_i. (It does not matter which sequence $\vec{x}\,'$ is chosen.) Set $\Lambda(\vec{x})\, E = \Lambda(\vec{x}\,')\, E$ in this case.

This operation Λ taking the n-tuple $\vec{x}$ and term E to a function $D^n \to D$ is called the *standard denotation map* for the assignment $\mathsf{f} \mapsto \mathsf{f}_\Lambda$. As a simple example, note that the denotation $\Lambda()\perp$ of $\perp$ is always the least element of the poset D. It is easy to see that the denotation of every open term is in fact a monotone function, which means it will always be possible to find the least fixed points required in the third clause. Let $\mathbf{Mon}$ be the collection of all standard denotation maps. If f_Λ is continuous for every function symbol, then the denotation of every open term is continuous as well; $\mathbf{Cont}$ is the corresponding subclass of $\mathbf{Mon}$.

On the other hand, it may happen that the poset D is not complete, but that the system of equations (2) nevertheless always has **least prefixed points** $\hat{c}_i$ in the sense that whenever $f_i(\vec{z}, \vec{d}) \leq z_i$ for each i, then $\hat{c}_i \leq z_i$. In this case, the inductive definition above still makes sense and yields a denotation map; let $\mathbf{Wk}$ be the collection of all denotation maps that arise in this way. Clearly $\mathbf{Cont} \subseteq \mathbf{Mon} \subseteq \mathbf{Wk}$.

For an example in $\mathbf{Cont}$, consider $D = \mathbf{N} \rightharpoonup \mathbf{N}$, the collection of all partial functions on the natural numbers, with its usual partial order. Suppose Exp_Λ is the map which takes a partial function $f(n)$ to $f'(n)$ defined as follows:

$$f'(n) = \begin{cases} 1 & \text{if } n = 0 \\ 2f(n-1) & \text{otherwise.} \end{cases}$$

Then the denotation $\Lambda()\ x\ \mathtt{where}\ \{x = \mathsf{Exp}(x)\}$ will be the exponential function $g(n) = 2^n$. This recursion on D is essentially ordinary recursion on the natural numbers, although FLR_0 (unlike the full FLR) cannot refer to specific natural numbers themselves or the way in which a partial function depends on its argument.

Another important example is the poset $\mathbf{str}(A)$ of **streams** over an alphabet A. This consists of two parts: the "divergent" streams, finite and infinite strings from A; and the "convergent" streams, finite strings consisting of symbols from A followed by one $\mathbf{t}$, a "termination" symbol not in A. The streams are ordered by the usual initial substring relation, so that the maximal elements of $\mathbf{str}(A)$ are exactly the convergent streams and the infinite streams. For each symbol a there is a natural prefixing operation $x \mapsto ax$ on $\mathbf{str}(A)$, which inserts an a at the beginning of the stream x. Now the FLR_0 expression

$$x \texttt{ where } \{x = ax\}$$

will denote the (divergent, infinite) stream $a^\infty = aaa\dots$.

Standard identities. Call $\forall \vec{x}\,(A = B)$ a **standard identity** if for all standard denotation maps $\Lambda \in \mathbf{Mon}$, $\Lambda(\vec{x})\,A = \Lambda(\vec{x})\,B$. Insofar as monotone, least-fixed-point recursion is the characteristic type of recursion, the standard identities capture the valid *laws of recursion*. As an example of a standard identity, for any unary function symbol f we have

$$x \texttt{ where } \{x = f(x)\} = f(\,x \texttt{ where } \{x = f(x)\}\,). \tag{3}$$

In other words, the fixed point of f *is* fixed by f. The class of standard identities is well-understood, as we'll describe below.

1.3. FLR_0 **structures.** Although the examples from the previous section employ the familiar recursion operation of taking least-fixed-points, any "abstract clone" can interpret FLR_0, if equipped with a notion of recursion.

Definition. An $\mathrm{FLR}_0(\tau)$ **structure** is a pair $R = (\Phi, \Lambda)$ where Φ is a ranked set called the **universe** of the structure, and Λ is a **denotation map** on $\mathrm{FLR}_0(\tau \cup \Phi)$, i.e., for any term $E \in \mathrm{FLR}_0(\tau \cup \Phi)$ and sequence $\vec{x} = x_1, \dots, x_n$ of variables containing all of the free variables of E,

$$\Lambda(x_1, \dots, x_n)\,E \in \Phi_n.$$

Every element of the universe acts as a symbol for itself; that is, Λ is required to satisfy

$$\Lambda(x_1, \dots, x_n)\,f(x_1, \dots, x_n) = f, \text{ for any } f \in \Phi_n. \tag{4}$$

Finally, the denotation map must be *compositional*, i.e., Λ must satisfy the following conditions for any term E and free substitutions s and t defined on $\vec{x} = x_1, \dots, x_n$:

$$\Lambda(\vec{y})\,\big(\Lambda(\vec{x})\,E\big)(s(x_1), \dots, s(x_n)) = \Lambda(\vec{y})\,E[s]. \tag{5}$$

$$\text{If } \Lambda(\vec{y})\,s(x_i) = \Lambda(\vec{z})\,t(x_i) \text{ for all } i, \text{ then } \Lambda(\vec{y})\,E[s] = \Lambda(\vec{z})\,E[t]. \tag{6}$$

If Φ is the universe of an FLR_0 structure R, call the elements of Φ_0 the **individuals** of R, and call the other elements (unary, binary, etc.) **transformations**. In fact, we will use the symbol R to denote the set of individuals Φ_0 as well, in much the same way that the name of a group is used for its set of elements. Now, we can think of the transformations as acting on R (in a possibly intensional way). Since everything in the universe names itself, each transformation $f \in \Phi_n$ induces a bona fide function $\bar{f} : R^n \to R$, called the **extension** of f, by

$$\bar{f}(r_1, \dots, r_n) = \Lambda()\,f(r_1, \dots, r_n), \text{ for each } r_1, \dots, r_n \in R = \Phi_0.$$

The FLR_0 structure R is called *extensional* if transformations are determined by their extensions, i.e., if $\bar{f} = \bar{g}$ implies $f = g$, and *intensional* otherwise. (The possibility of intensional FLR_0 structures motivates our use of the neutral term "transformations" for the members of Φ_n, as opposed to a more familiar term like "operations" which might suggest that these objects must be functions.)

The structure R **satisfies** a formula $\phi = \forall\vec{x}\,(A = B)$ if for all lists of variables $\vec{y}$ including all free variables of ϕ, $\Lambda(\vec{x}, \vec{y}\,A = \Lambda(\vec{x}, \vec{y})\,B$. (It suffices to check the minimal list of variables $\vec{y}$.) R satisfies a set of formulas if it satisfies each one in the set.

We regard each denotation map in $\mathbf{Cont}, \mathbf{Mon}, \mathbf{Wk}$ as an FLR_0 structure in which Φ_0 is a poset D, Φ_n is some collection of monotone (continuous) maps $D^n \to D$, and Λ is extended in the obvious way to $\mathrm{FLR}_0(\tau \cup \Phi)$, with each object naming itself. These familiar, extensional structures serve as the main motivating examples for the more general theory presented here.

The interpretation of `where` is so far left entirely open, and could be trivial: every term in which `where` occurs might be assigned the same denotation. To avoid such uninteresting examples, we will only consider **normal** FLR_0 structures R, which satisfy the following additional conditions:

(1) If $\Lambda(\vec{x}, \vec{y})\,A_i = \Lambda(\vec{x}, \vec{y})\,B_i$ for each i, then

$$\Lambda(\vec{y})\ A_0 \ \texttt{where}\ \{\vec{x} = \vec{A}\} = \Lambda(\vec{y})\ B_0\ \texttt{where}\ \{\vec{x} = \vec{B}\}\,.$$

(In other words, recursion is compositional in the same sense as function application above.)

(2) R satisfies all standard identities.

Note that the standard identity (3) already prevents a completely trivial recursion operation, unless every transformation fixes the same individual. It might appear that assuming all standard identities is too strong, but the axiomatization of the standard identities below and the wide range of examples of normal structures show that the assumption is not overly restrictive. For example, although the structures in $\mathbf{Mon}$ and $\mathbf{Cont}$ are clearly normal, all of the structures in the much broader class $\mathbf{Wk}$ are normal as well, a non-trivial fact shown in Section 3.1.

Any class $\mathscr{R}$ of FLR_0 structures gives rise to a corresponding relation of semantic **consequence**. Let Γ be a set of formulas and ϕ be any formula. Write $\Gamma[s]$ for $\{\gamma[s] \mid \gamma \in \Gamma\}$. Then ϕ is a consequence of Γ over $\mathscr{R}$, written $\Gamma \vDash_{\mathscr{R}} \phi$, when the following condition holds for all $R \in \mathscr{R}$ and substitutions s: If R satisfies $\Gamma[s]$, then R satisfies $\phi[s]$.

For example, $x = x' \vDash f(x) = f(x')$ for any class of structures, by the compositionality condition (6). Also,

$$\forall x\,(f(x) = g(x)) \vDash_{\mathbf{Mon}} x\ \texttt{where}\ \{x = f(x)\} = x\ \texttt{where}\ \{x = g(x)\}$$

since identical monotone functions have the same least fixed point; but

$$f(x) = g(x) \nvDash_{\mathbf{Mon}} x\ \texttt{where}\ \{x = f(x)\} = x\ \texttt{where}\ \{x = g(x)\}$$

since it might happen that $f(d) = g(d)$, for a particular d, with no implications for the fixed points of the right-hand side.

In the special case with no hypotheses, we write just $\vDash_{\mathscr{R}} \phi$ and say that ϕ is *valid* in $\mathscr{R}$. This simply means every structure in $\mathscr{R}$ satisfies ϕ. Note that validity makes no distinction between equations and identities: $A = B$ is valid if and only if $\forall\vec{x}\,(A = B)$ is valid, for any list of variables $\vec{x}$. For this reason, we will loosen the terminology slightly, e.g., "standard identity" will refer to any formula (closed or open) which is valid in $\mathbf{Mon}$.

TABLE 1. **The proof system for** FLR_0.

Logical Axioms

(L1) $\phi \vdash \phi$.
(L2) (Equality axioms) $\vdash A = A$; $\quad A = B \vdash B = A$; $\quad A = B, B = C \vdash A = C$.
(L3) (Replacement) $A = B \vdash E[A/x] = E[B/x]$, provided the substitutions are free.
(L4) (Specialization) $\forall x\,(\phi(x)) \vdash \phi(E)$, provided the substitution is free.

Logical Inference Rules

(L5) (Weakening) If $\Gamma \vdash \phi$, then $\Gamma \cup \Delta \vdash \phi$.
(L6) (Cut) If $\Gamma, \psi \vdash \phi$ and $\Gamma \vdash \psi$, then $\Gamma \vdash \phi$.
(L7) (Generalization) If $\Gamma \vdash \phi(x)$ and x is not free in Γ, then $\Gamma \vdash \forall x\,(\phi(x))$.

Recursion Axioms

(R1) (Head)

$$\vdash A(x_1, \ldots, x_n) \texttt{ where } \{\vec{x} = \vec{B}\}$$
$$= A(\, x_1 \texttt{ where } \{\vec{x} = \vec{B}\}, \ldots, x_n \texttt{ where } \{\vec{x} = \vec{B}\}\,).$$

(R2) (Bekič-Scott)

$$\vdash A \texttt{ where } \{\vec{y} = \vec{C}, \vec{x} = \vec{B}\}$$
$$= \big(A \texttt{ where } \{\vec{y} = \vec{C}\}\big) \texttt{ where } \{\ldots, x_i = B_i \texttt{ where } \{\vec{y} = \vec{C}\}, \ldots\}.$$

(R3) (Fixpoint) $\vdash A \texttt{ where } \{x = A\} = x \texttt{ where } \{x = A\}$.

Recursion Inference Rule

(RI) Suppose we are given Γ and FLR_0 terms

$$A \equiv A_0 \texttt{ where } \{x_1 = A_1, \ldots, x_n = A_n\} \quad \text{and}$$
$$B \equiv B_0 \texttt{ where } \{y_1 = B_1, \ldots, y_m = B_m\},$$

where no x_i occurs in B, no y_j occurs in A, and none of the variables x_i or y_j occur free in Γ. If there is a set of equations Σ each of the form $x_i = y_j$ such that $\Gamma, \Sigma \vdash A_0 = B_0$ and $\Gamma, \Sigma \vdash A_i = B_j$ for each $(x_i = y_j) \in \Sigma$, then $\Gamma \vdash A = B$.

1.4. The standard identities. The clauses in Table 1 above inductively define a provability relation $\Gamma \vdash \phi$, where Γ is an arbitrary set of formulas and ϕ is a single formula. Note that the clauses labeled "Axioms" are really axiom schema, since A, B, C, and E range over arbitrary FLR_0 terms. For convenience, we write just $\psi \vdash \phi$ for $\{\psi\} \vdash \phi$.

The logical axioms and rules (L1–7) are standard and correspond to ordinary equational logic. The remaining items reflect the special properties which characterize fixed-point recursion. The Head Axiom (R1) captures the idea that a "`where`" term is evaluated by solving a system of equations and substituting the results into the head term (to the left of `where`). The Bekič-Scott Axiom (R2) corresponds to the theorem of the same name, relating simultaneous and iterated recursion. References for this theorem (due independently to Bekič and Scott) may be found in de Bakker [5], who was apparently the first to use this theorem to support a proof rule for "program equivalence." Finally, the Recursion Inference Rule (RI) plays a central role in this axiomatization, and is certainly the most complex and interesting rule. The only other proof rule we know of equivalent to Recursion Inference is the "Functorial Dagger Implication" (for base morphisms) of Bloom and Ésik [1]. However, one may regard (RI) as a descendent of the principle called "Scott induction" or "fix-point induction" (see Stoy [22]). The rule (RI) formulates a set of instances of this principle which are valid for general (not necessarily continuous, as in [22]) least fixed points, and which suffice to establish all valid identities. The references [9, 8] investigate this connection in greater detail. The following proposition is straightforward; the case of Recursion Inference is proved by induction on the stages of a recursion.

PROPOSITION 1 (Soundness). *The provability relation $\vdash$ is* ***sound*** *for $\models_{\mathbf{Mon}}$, i.e., $\Gamma \vdash \phi$ implies that $\Gamma \models_{\mathbf{Mon}} \phi$. In particular, every provable identity is a standard identity.*

In fact, the proof system axiomatizes the standard identities. Call a normal $\mathrm{FLR}_0(\tau)$ structure $R = (\Phi, \Lambda)$ **reasonably free** if the following conditions are satisfied for each pair of distinct function symbols f and g in τ:

(1) The extension $\bar{f}$ of f is injective, and its image does not contain $\perp_R = \Lambda() \perp$. (Recall $\perp \equiv x$ `where` $\{x = x\}$.)

(2) $\bar{f}$ and $\bar{g}$ have disjoint images.

THEOREM 2 (Completeness/Decidability). *If R is a reasonably free interpretation for $\mathrm{FLR}_0(\tau)$, then R satisfies a formula ϕ of $\mathrm{FLR}_0(\tau)$ if and only if $\vdash \phi$. There is a reasonably free structure in $\mathbf{Cont}$ for each signature τ, so that this validity also coincides with the formula being a standard identity ($\models_{\mathbf{Mon}} \phi$) and with $\models_{\mathbf{Cont}} \phi$. Finally, this common validity is decidable.*

As the decidability portion of this result (for non-nested "`where`") appears in [4], and all of its claims are implicit in [1] (see Theorem 6.1.2, and the discussion on pp. 191–3), we only briefly indicate a proof, highlighting features that will be useful later. A similar proof, in greater detail, appears in [10].

The proof of Theorem 2 proceeds in two major steps. The first is purely syntactic, and reduces each FLR_0 term to one in **simplified form**, i.e.,

$$E \equiv x_e \texttt{ where } \{x_1 = E_1, \dots, x_n = E_n\}$$

where $1 \leq e \leq n$ and for each i, either $E_i \equiv x_i$, or $E_i \equiv f_i(z_1, \dots, z_{m_i})$ where f_i is some function symbol and each z_j is one of the x_k.

LEMMA 3 (Simplification). *Every closed term of FLR_0 is provably equal to one in simplified form.*

The second step attempts to construct an application of the Recursion Inference rule between these reduced forms, in such a way that the denotation of the two terms will be different if this construction fails. Given two simplified terms,

$$A \equiv x_a \texttt{ where } \{x_1 = A_1, \ldots, x_m = A_m\}$$
$$B \equiv y_b \texttt{ where } \{y_1 = B_1, \ldots, y_n = B_n\},$$

first consider A_a and B_b. If they are simply x_a and y_b, respectively, both A and B provably equal $\perp$ and we are done. If only $A_a \equiv x_a$ (or vice-versa), then A and B cannot denote the same individual, as $\perp_R$ is not in the range of the extension of any function symbol. Otherwise, A_a and B_b must use the same function symbol (since extensions of distinct symbols have disjoint ranges), and in fact corresponding arguments to this function must also be equal since its extension is injective. This leads to a new sequence of pairs to examine. Proceeding by induction, one obtains either a list of equations for applying the Recursion Inference rule, or a demonstration that A and B have distinct denotations in any reasonably free structure.

In fact, this argument shows slightly more. The terms to which the Recursion Inference rule is applied in the end are just alphabetic variants of each other, so the proofs of the hypotheses needed in RI are trivial. Thus, the inference rule RI can be replaced with the following schema of identities:

(Alphabetic Identification identity) Suppose we are given FLR_0 terms

$$A \equiv x_a \texttt{ where } \{x_1 = A_1, \ldots, x_n = A_n\} \quad \text{and}$$
$$B \equiv y_b \texttt{ where } \{y_1 = B_1, \ldots, y_m = B_m\},$$

where no x_i occurs in B and no y_j occurs in A. If there is a relation $\sim$ between $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_m\}$ such that $x_a \sim y_b$ and $A_i \equiv_\sim B_j$ whenever $x_i \sim y_j$, then $\vdash A = B$.

The only other consequences of Recursion Inference used (in the Simplification Lemma) are the relatively trivial:

(Part Replacement rule) If $\Gamma \vdash \forall\vec{x}(A_i = B_i)$ for each i, $0 \leq i \leq n$, then

$$\Gamma \vdash A_0 \texttt{ where } \{\vec{x} = \vec{A}\} = B_0 \texttt{ where } \{\vec{x} = \vec{B}\}$$

(Permutation identity) For any permutation ρ of $\{1 \ldots n\}$,

$$\vdash A_0 \texttt{ where } \{\vec{x} = \vec{A}\} = A_0 \texttt{ where } \{x_{\rho 1} = A_{\rho 1}, \ldots, x_{\rho n} = A_{\rho n}\}.$$

Note that the Part Replacement rule just corresponds to compositionality of `where`, the first condition on normal structures. To sum up these observations, let $\vdash_w$ be the provability relation obtained by replacing the Recursion Inference Rule with the Permutation and Alphabetic Identification axiom schemes and Part Replacement inference rule.

COROLLARY 4. *$\vdash_w \phi$ if and only if $\vdash \phi$.*

The reasonably free $\mathrm{FLR}_0(\tau)$ structure referred to in Theorem 2 is supplied by the literally free object in the category of posets interpreting the signature τ with continuous maps as morphisms. This construction matches that of $\tau_\perp\mathbf{TR}$ on p. 249 of [1], but we provide a sketch here as the details will be useful below.

Intuitively, the elements of the free structure $\tau_\perp\mathbf{TR}$ are (finite and infinite) trees with nodes labeled by function symbols in τ. Technically, define a τ-tree to be a function $t\colon T \to \tau$ where T is a set of finite sequences of natural numbers closed under initial segments, such that if $\alpha\,\hat{}\,\langle n\rangle$ is in T where α is a sequence and $n \in \mathbf{N}$,

TABLE 2. **Iteration Theory axioms.**

In the following axioms, f and g are arbitrary morphisms with appropriate source and target. $f \cdot g$ denotes the composition of f followed by g. $\langle f, g\rangle$ is the pairing of $f : n \to p$ and $g : m \to p$, possible since $n + m$ is the coproduct of n and m in an iteration theory. $f \oplus g$ is the "separated sum" of $f : m \to k$ and $g : n \to l$, yielding $f \oplus g : n + m \to k + l$. Finally, a **base morphism** $\rho : m \to n$ is one of the form $\langle (i_1)_n, \dots , (i_m)_n\rangle$, made up from distinguished morphisms by pairing. The base morphism ρ is said to be **surjective** if every j_n for $1 \le j \le n$ appears as one of $(i_1)_n$ through $(i_m)_n$.

- (Parameter Identity) $(f \cdot (\mathrm{id}_n \oplus g))^\dagger = f^\dagger \cdot g$.
- (Pairing Identity) $\langle f, g\rangle^\dagger = \langle f^\dagger \cdot \langle h^\dagger, \mathrm{id}_p\rangle, h^\dagger\rangle$, where $f : n \to n + m + p$, $g : m \to n + m + p$, and $h = g \cdot \langle f^\dagger, \mathrm{id}_{m+p}\rangle$.
- (Fixed Point Identity) $f^\dagger = f \cdot \langle f^\dagger, \mathrm{id}_p\rangle$.
- (Commutative Identity) For any $f = \langle f_1, \dots , f_n\rangle : n \to m + p$ and $g_i : m \to k$, write $f \parallel (g_1, \dots , g_n)$ for the morphism

$$\langle f_1 \cdot (g_1 \oplus \mathrm{id}_p), \dots , f_n \cdot (g_n \oplus \mathrm{id}_p)\rangle : n \to k + p.$$

Then,

$$((\rho \cdot f) \parallel (\rho_1, \dots , \rho_m))^\dagger = \rho \cdot (f \cdot (\rho \oplus \mathrm{id}_p))^\dagger,$$

where $f : n \to m + p$, $\rho : m \to n$ is a surjective base morphism, and the $\rho_i : m \to m$ are base morphisms satisfying $\rho_i \cdot \rho = \rho$.

---

then n is less than the arity of $t(\alpha)$. The partial order on $\tau_\perp\mathbf{TR}$ is the (labeled) subtree ordering. Finally, to each n-ary function symbol f there corresponds an n-ary function on trees $f(t_0, \dots , t_{n-1})$, operating as follows: If $u = f(t_0, \dots , t_{n-1})$, then $u(\langle\rangle) = \mathsf{f}$ (i.e., the root of u is labeled with f) and for longer sequences, $u(\langle i\rangle \hat{}\, \alpha) = t_i(\alpha)$ (i.e., the trees t_0 through t_{n-1} are attached to the root in order). The reasonably free properties are not difficult to verify; for the proof this structure is actually free, see [1].

§2. **Correspondence with Iteration Theories.** The goal of this section is to show that FLR_0 structures capture exactly the same mathematical structure as the **iteration theories** of Bloom and Ésik. For this purpose, it is convenient to make a thumbnail sketch of the definitions; details are in [1]. An **iteration theory** is a category T with objects N, **distinguished morphisms** $1_n, 2_n, \dots , n_n : 1 \to n$ for each $n \in \mathrm{N}$ which make the object n the n-fold coproduct of 1, and an **iteration operation** $\cdot^\dagger$ which takes a morphism $f : n \to n + p$ and yields $f^\dagger : n \to p$. In addition, the iteration operation is required to satisfy certain properties; Bloom and Ésik give many alternative axiomatizations of these. One such axiomatization is summarized in Table 2; note that it is shorter than our axiomatization for $\vdash$ because requiring T to be a category already summarizes the logical properties (L1–7).

One intuition for these definitions is that a morphism $g : n \to m$ is a system of n operations depending on m indeterminates, and composition with $h : m \to p$ corresponds to substituting a system of m other operations for these indeterminates.

TABLE 3. **Correspondence between FLR_0 and iteration theories.**

FLR_0	Iteration Theories
Compositionality conditions (4, 5, 6)	Algebraic Theory Properties
Fixed Point Identity	Fixed Point Identity
Bekič-Scott Identity	Pairing Identity
Alphabetic Identification Identity	Commutative Identity
Recursion Inference Rule	Functorial Dagger Identity for base morphisms

The iteration operation should set the ith indeterminate equal to the ith operation and "solve."

Table 3 gives a rough overview of the correspondence between FLR_0 and iteration theories.

The appendix to this paper gives a detailed sketch of the proof of the following theorem:

THEOREM 5. *The categories of iteration theories with iteration theory morphisms and* FLR_0 *structures with their homomorphisms are equivalent.*

Thus, iteration theory and the study of FLR_0 examine the identical mathematical structure, albeit from different viewpoints. Bloom and Esik discuss a similar correspondence between iteration theories and the μ-calculus in [2]. The choice of formalism highlights certain features and suppresses others: for example, FLR_0 focuses attention on systems of equations and Scott-Bekič in comparison to μ-calculus, where this law is tacitly assumed throughout. Iteration theory emphasizes the relationship between substitution and fixed-points with its Parameter Identity which is implicit in the other formalisms. In any case, we have "dictionaries" that can translate results back and forth.

§3. **The standard identities, part II.** Unlike ordinary first-order logic, there are striking differences between the valid identities of FLR_0 and its full consequence relation with hypotheses. The standard identities are decidable, and remarkably robust over a vast array of different interpretations of `where`. On the other hand, the consequence relation depends very critically on the class of models considered, and is typically much more complex: consequence over $\mathbf{Cont}$ is not axiomatizable.

It is well known that many classes of structures have the same collection of valid identities: [1] surveys a wide range of such classes, and we would also like to call attention to the *player structures* of [19, 20], which are a class of highly intensional models of concurrency which satisfy exactly the standard identities. The results of this section illustrate two other, important aspects of this robustness. First, the standard identities might seem to say something about an iterative process of finding fixed points, particularly because the simplest, natural proofs of Recursion Inference (and the weaker Alphabetic Identification) involve induction on stages. However, it turns out that the weak structures satisfy the standard identities, even though it may not be possible to iterate to reach fixed points on these structures – the required suprema of increasing sequences may not exist. Second, exactly the standard identities hold on very restrictive subclasses of $\mathbf{Cont}$: it takes very little structure to ensure no additional identities will be valid.

3.1. Weak structures are normal. Before proceeding to the theorem, it may not be immediately clear that there are any weak structures not in $\mathbf{Mon}$. However, suppose $(D, \leq, f_1, \dots, f_n)$ is an ordinary model for first order logic in which $\leq$ is a partial order, the operations are monotone, and every finite system of monotone functions which are elementarily definable from parameters has a least fixed point. These properties are clearly first order axiomatizable, the last one by a r.e. list of axioms. Every such model defines a weak structure (by dropping $\leq$ and taking all the monotone, definable operations as the universe of the structure). Now by compactness, it is easy to find such structures in which the underlying poset $(D, \leq)$ is not directed-complete. In fact, every weak structure arises in this way as a reduct of a first-order model, and this point of view will help later to show that the consequence relation for $\mathbf{Wk}$ is recursively enumerable.

THEOREM 6. $\models_{\mathbf{Wk}} \phi$ *if and only if* $\models_{\mathbf{Mon}} \phi$. *In other words, the weak structures satisfy exactly the standard identities.*

This theorem actually follows from the results in [9], which show that in fact least *pre*-fixed points are not necessary: least fixed points for every system of equations plus the Scott-Bekič law imply all standard identities. But the following proof for just Theorem 6 is much shorter and more direct.

PROOF. The "only if" direction is obvious. For the other direction, it suffices by Corollary 4 to check the soundness of the proof system $\vdash_w$.

The standard proofs of soundness for rules (L1–7) and (R1–3) only use the leastness of the (pre-)fixed points for a system of equations, and hence hold for weak structures. The Part Replacement rule is trivial because weak structures are extensional and identical functions have identical least prefixed points. The Permutation identity is also easy; the order of a sequence of functions cannot affect their least prefixed points, either. Only the Alphabetic Identification identity remains to be checked.

Recall that the Alphabetic Identification axiom scheme says that two terms A and B are equal if

$$A \equiv x_a \ \texttt{where} \ \{x_1 = A_1, \dots, x_n = A_n\} \quad \text{and}$$
$$B \equiv y_b \ \texttt{where} \ \{y_1 = B_1, \dots, y_m = B_m\},$$

and there is a relation $\sim$ between the disjoint sets of variables $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_m\}$ such that $x_a \sim y_b$ and $A_i \equiv_\sim B_j$ whenever $x_i \sim y_j$.

One can simplify the relation $\sim$ allowed in Alphabetic Identification in two ways. First, any variable x_i not related to anything can be eliminated from A using only the Head (R1) and Scott-Bekič (R2) axioms. Second, any relation decomposes as a many-one relation followed by a one-many relation. Thus, it suffices just to allow $\sim$ to relate each x_i to exactly one y_j.

An example may make this second simplification clearer. Alphabetic Identification as stated above shows (in a single step) that the following two terms are equal:

$$x_1 \texttt{ where } \{x_1 = f(x_2, x_3), x_2 = f(x_2, x_3), x_3 = g(x_1, x_2)\} \quad \text{and}$$
$$z_2 \texttt{ where } \{z_1 = f(z_2, z_3), z_2 = f(z_1, z_3), z_3 = g(z_2, z_2)\},$$

by relating $x_1 \sim z_1$, $x_1 \sim z_2$, $x_2 \sim z_1$, $x_2 \sim z_2$, and $x_3 \sim z_3$. However, the reduced form of Alphabetic Identification can show them both to equal

$$y \texttt{ where } \{y = f(y, y'), y' = g(y, y)\}$$

using relations that associate each x_i or z_j to one of y or y' but not both. (Specifically, $x_1 \sim y$, $x_2 \sim y$, $x_3 \sim y'$, and similarly for the z_j.)

For notational convenience, first consider the case of only one recursion variable in B, so that every x_i is related to this one variable y. Then there is a term $C(z_1, \dots, z_l, \vec{w})$ so that $B \equiv y \texttt{ where } \{y = C(y, y, \dots, y, \vec{w})\}$ and A has the form

$$x_a \texttt{ where } \{\dots, x_i = C(x_{\pi(i,1)}, x_{\pi(i,2)}, \dots, x_{\pi(i,l)}, \vec{w}), \dots\},$$

where π is some map from $\{1 \dots n\} \times \{1 \dots l\}$ to $\{1 \dots n\}$.

Let $f : D^{l+k} \to D$ be the denotation of $C(z_1, \dots, z_l, \vec{w})$ in some weak interpretation. By definition of a weak interpretation, there is a least prefixed point $\hat{y}(\vec{w})$ satisfying $\hat{y} = f(\hat{y}, \hat{y}, \dots, \hat{y}, \vec{w})$ as well as least prefixed points $\hat{x}_i(\vec{w})$ satisfying

(7) $\quad \hat{x}_1 = f(\hat{x}_{\pi(1,1)}, \hat{x}_{\pi(1,2)}, \hat{x}_{\pi(1,l)}, \vec{w}), \dots, \hat{x}_n = f(\hat{x}_{\pi(n,1)}, \hat{x}_{\pi(n,2)}, \dots, \hat{x}_{\pi(n,l)}, \vec{w}).$

Since the index a may be anything between 1 and n, the soundness of Alphabetic Identification amounts to showing that all of the $\hat{x}_i$ are equal to $\hat{y}$.

Clearly the n-tuple $\hat{y}, \hat{y}, \dots, \hat{y}$ satisfies the system defining the $\hat{x}_i$, so by the latter's leastness, each $\hat{x}_i(\vec{w}) \leq \hat{y}(\vec{w})$ for all $\vec{w}$. If all of the $\hat{x}_i$ were equal to each other, the same argument in reverse would show that they are greater than $\hat{y}$. A priori, the $\hat{x}_i$ might be different, but they do satisfy a "symmetrized" version of the system (7), which will allow the same conclusion.

For any sequence $\mathbf{u} = \langle u_1, \dots, u_n \rangle$ of n indices from 1 to n, let $\pi^*(\mathbf{u}, j)$ denote the sequence $\langle \pi(u_1, j), \dots, \pi(u_n, j) \rangle$. Pick a collection of fresh variables $x_{\mathbf{u}}$ where $\mathbf{u}$ ranges over all such sequences. Now consider the longer term

(8) $\quad x_{\langle 1,2,\dots,n \rangle} \texttt{ where } \{\dots, x_{\mathbf{u}} = C(x_{\pi^*(\mathbf{u},1)}, \dots, x_{\pi^*(\mathbf{u},l)}, \vec{w}), \dots\},$

in which the equations inside of `where` range over all sequences $\mathbf{u}$. These equations of course lead to least prefixed points $\hat{x}_{\mathbf{u}}(\vec{w})$, which satisfy

(9) $\quad \hat{x}_{\mathbf{u}} = f(\hat{x}_{\pi^*(\mathbf{u},1)}, \dots, \hat{x}_{\pi^*(\mathbf{u},l)}, \vec{w})$

for each $\mathbf{u}$. Now, for constant sequences $\langle i, \dots, i \rangle$, written $\langle i \rangle$ for brevity, we have $\pi^*(\langle i \rangle, j) = \langle \pi(i, j) \rangle$. Hence, the entire original system (7) is reproduced as the portion of (8) with constant sequences for indices, and as a result $\hat{x}_{\langle i \rangle} = \hat{x}_i$ for each i.

Exploiting the symmetry of (8) will lead to the desired conclusion. Let functions $\rho\colon \{1, \dots, n\} \to \{1, \dots, n\}$ act on sequences as follows:

$$\rho\langle u_1, \dots, u_n \rangle = \langle u_{\rho(1)}, \dots, u_{\rho(n)} \rangle.$$

Under this action, $\pi^*(\rho\mathbf{u}, j) = \rho\pi^*(\mathbf{u}, j)$. This fact means that setting each $x_{\mathbf{u}}$ to $\hat{x}_{\rho\mathbf{u}}$ leads to another solution of the equations in (8).

Thus, the leastness of the $\hat{x}_{\mathbf{u}}$ shows that for each $\mathbf{u}$ and ρ, $\hat{x}_{\mathbf{u}} \leq \hat{x}_{\rho\mathbf{u}}$. In particular, let $\mathbf{t} = \langle 1, 2, \ldots, n \rangle$ be the sequence containing every index in order. Then $\hat{x}_{\mathbf{t}} \leq \hat{x}_{\pi^*(\mathbf{t},j)}$ for each j (using the maps $\pi(\cdot, j)$ for ρ), and $\hat{x}_{\mathbf{t}} \leq \hat{x}_{\langle i \rangle} = \hat{x}_i$ for each i (using constant maps). So $\hat{x}_{\mathbf{t}}$ is a common lower bound for all of the solutions to the original system (7).

Finally, these properties imply

$$f(\hat{x}_{\mathbf{t}}(\vec{w}), \hat{x}_{\mathbf{t}}(\vec{w}), \ldots, \hat{x}_{\mathbf{t}}(\vec{w}), \vec{w}) \leq f(\hat{x}_{\pi^*(\mathbf{t},1)}(\vec{w}), \hat{x}_{\pi^*(\mathbf{t},2)}(\vec{w}), \ldots, \hat{x}_{\pi^*(\mathbf{t},l)}(\vec{w}), \vec{w})$$
$$= \hat{x}_{\mathbf{t}}(\vec{w}).$$

(The equality comes from (9).) Leastness of $\hat{y}$ yields that $\hat{y}(\vec{w}) \leq \hat{x}_{\mathbf{t}}(\vec{w}) \leq \hat{x}_i(\vec{w})$, so each $\hat{x}_i = \hat{y}$ as desired.

If there is more than one recursion variable in the term B, the proof proceeds similarly. For example, we need to show that the denotation of

$$x_1 \; \texttt{where} \; \{x_1 = f(x_1, x_3), x_2 = f(x_1, x_4), x_3 = g(x_2, x_4), x_4 = g(x_1, x_4)\}$$

is equal in every weak interpretation to that of

$$y_1 \; \texttt{where} \; \{y_1 = f(y_1, y_4), y_4 = g(y_1, y_4)\}.$$

Here, the relation $\sim$ associates x_1 and x_2 to y_1 and x_3 and x_4 to y_4. Therefore, the "symmetrized" term corresponding to (8) would have fresh variables $x_{\langle 11 \rangle}, x_{\langle 12 \rangle}, x_{\langle 21 \rangle}, x_{\langle 22 \rangle}$ and $x_{\langle 33 \rangle}, x_{\langle 34 \rangle}, x_{\langle 43 \rangle}, x_{\langle 44 \rangle}$. Only maps which rearrange indices of variables corresponding by $\sim$ to the same y_j are used. Thus in this example, there are essentially two copies of the above proof going on in parallel, one to show that $\hat{x}_1$ and $\hat{x}_2$ both equal $\hat{y}_1$, and the other to show that $\hat{x}_3$ and $\hat{x}_4$ both equal $\hat{y}_4$. In general, there would be m copies of the proof going on, one among each set of variables associated to the same y_j. ⊣

3.2. Subclasses of Cont. The result of this section illustrates the fact that it is possible to restrict to small classes of posets without making any additional identities valid. In other words, it is not necessary to have "much room" in the poset to falsify a non-standard identity, just freedom in choosing the monotone (continuous) maps used in the FLR_0 structure.

Let $\mathscr{P}$ denote the class of continuous FLR_0 structures whose underlying partially ordered set is of the form $(\mathrm{Pow}(A), \subseteq)$ for some set A. Let $\mathbf{Strm}$ denote the class of continuous structures whose underlying poset is $\mathbf{str}(A)$ for some set A, and $\mathbf{Strm}_a$ be the structures on $\mathbf{str}(\{a\})$. Similarly, let $\mathbf{Lin}$ be the continuous structures on linear orders, and $\mathbf{FLin}$ be the structures on *finite* linear orders. Finally, let $\mathscr{R}_{\omega+1}$ be the continuous structures on the single poset $\omega + 1$ with its usual (ordinal) order.

THEOREM 7. *The valid identities for all of the following classes of structures are exactly the standard identities*: $\mathbf{Cont}$, $\mathscr{P}$, $\mathbf{Strm}$, $\mathbf{Strm}_a$, $\mathbf{Lin}$, $\mathbf{FLin}$, *and* $\mathscr{R}_{\omega+1}$.

PROOF. There is nothing new to prove for $\mathbf{Cont}$, and all of the others are subclasses of $\mathbf{Cont}$ so certainly the standard identities hold in them. We need to verify that no additional identities hold.

For $\mathscr{P}$, it suffices to find a reasonably free structure in this class for any signature τ. In fact, since any formula only contains finitely many symbols, it is enough to consider finite τ. Choose an injective pairing $\langle \cdot, \cdot \rangle$ on the natural numbers, and an injective map $\iota\colon \tau \to \mathbf{N}$. Consider the standard denotation map Λ on $(\mathrm{Pow}(\mathbf{N}), \subseteq)$ produced by assigning the following (continuous) function to the symbol f:

$$\mathsf{f}_\Lambda(A_1, \ldots, A_n) = \{\langle 0, \iota(\mathsf{f})\rangle\} \cup \{\, \langle i, a\rangle \mid 1 \le i \le n \text{ and } a \in A_i \,\}\,.$$

Check that Λ satisfies the conditions for being reasonably free.

Next, we attack **FLin** and **Lin**. As before, fix a finite signature τ, and choose an arbitrary linear order on τ. This choice induces a linear "breadth-first lexicographic" ordering $\preceq$ on $\tau_\bot\mathbf{TR}$ as follows: For any two distinct trees u and t, find the shortest, lexicographically first sequence $\alpha \in \mathbf{N}^*$ such that $t(\alpha) \neq u(\alpha)$. If $t(\alpha)$ is undefined, then $t \preceq u$ (and vice versa). Otherwise, both $t(\alpha)$ and $u(\alpha)$ are defined, so use the order on τ to choose which of t and u is $\preceq$-smaller.

The order $\preceq$ refines the ordinary subtree ordering $\le$ on $\tau_\bot\mathbf{TR}$ and in fact preserves the sups of $\le$-chains. Furthermore, the interpretation of each function symbol as defined in the description of $\tau_\bot\mathbf{TR}$ is monotone in $\preceq$ as well as $\le$. Thus, the least fixed points of a system of equations in $\tau_\bot\mathbf{TR}$ are also the least $\preceq$-fixed points. This property holds even in the strong sense: if f is a unary function with $\le, \preceq$-least fixed point $\hat{x}$, for example, then

$$f(y) \preceq y \text{ implies } x \preceq y. \tag{10}$$

Suppose that A and B are FLR_0 expressions in simplified form which have different denotations on $\tau_\bot\mathbf{TR}$; we construct a monotone structure on a finite linear order in which A and B also have different denotations, completing the proof for **FLin** and hence for **Lin**. We may write

$$A \equiv x_a \texttt{ where } \{x_1 = f_1(\vec{x}), \ldots, x_n = f_n(\vec{x})\}\,,$$
$$B \equiv y_b \texttt{ where } \{y_1 = g_1(\vec{x}), \ldots, y_m = g_m(\vec{x})\}\,.$$

The two systems of equations inside "`where`" have least fixed points $\hat{x}_i$ and $\hat{y}_j$ in $\tau_\bot\mathbf{TR}$, with $\hat{x}_a \neq \hat{y}_b$ by hypothesis.

Now consider the finite linear order

$$L = (\{\bot\} \cup \{\, \hat{x}_i \mid 1 \le i \le n \,\} \cup \{\, \hat{y}_j \mid 1 \le j \le m \,\}, \preceq).$$

Define a monotone function f^L corresponding to each function symbol $\mathsf{f} \in \tau$ as follows:

$$f^L(z_1, \ldots, z_k) = \inf_{\preceq} \{\, l \in L \mid f(z_1, \ldots, z_k) \preceq l \,\}\,. \tag{11}$$

Here f is the function corresponding to f in $\tau_\bot\mathbf{TR}$; and the inf on the right hand side is taken to be the maximum element of L if the given set is empty.

Let $\tilde{x}_i$ and $\tilde{y}_j$ be the $\preceq$-least fixed points in L of the same systems of equations, from A and B. Certainly the old fixed points $\hat{x}_i$ and $\hat{y}_j$ still satisfy these equations in L, so $\tilde{x}_i \preceq \hat{x}_i$ and $\tilde{y}_j \preceq \hat{y}_j$.

On the other hand, by definition and monotonicity of f_i,

$$\hat{x}_i = f_i(\hat{x}_{i_0}, \ldots, \hat{x}_{i_l}) \succeq f_i(\tilde{x}_{i_0}, \ldots, \tilde{x}_{i_l}).$$

Since $\hat{x}_i \in L$, the inf on the right of (11) will be non-empty, meaning that

$$f_i(\tilde{x}_{i_0}, \ldots, \tilde{x}_{i_l}) \preceq f_i^L(\tilde{x}_{i_0}, \ldots, \tilde{x}_{i_l}) = \tilde{x}_i.$$

This last relationship (which holds for all $i \le n$) yields $\hat{x}_i \preceq \tilde{x}_i$ because of the strong $\preceq$-leastness of the $\hat{x}_i$ in $\tau_\bot\mathbf{TR}$, as in equation (10).

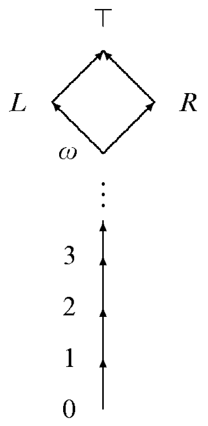

FIGURE 1. The complete poset Y.

The same properties hold for the y_j, meaning that A and B have the same denotations in L as they do in $\tau_\perp \mathbf{TR}$. In particular, A and B still have distinct denotations, as desired.

Any monotone function on a finite linear order can obviously be extended to one on $\omega + 1$ with the "same" least fixed point, so the result extends to $\mathscr{R}_{\omega+1}$. A similar tactic serves for $\mathbf{Strm}_a$ and hence $\mathbf{Strm}$. ⊣

§4. **Consequence relations.** Questions concerning FLR_0 consequence relations (with hypotheses) seem in general much more difficult than questions about valid identities. For example, no complete axiomatization for the consequence relation on any natural class of FLR_0 structures is known, other than the following which is obtained by a standard "term-model"-type construction.

THEOREM 8. *Let $\mathscr{N}$ be the collection of* all *normal* FLR_0 *structures. Then* $\Gamma \models_{\mathscr{N}} \phi$ *if and only if* $\Gamma \vdash_w \phi$.

Moreover, the consequence relation $\Gamma \models_{\mathscr{R}} \phi$ depends very sensitively on the collection of models $\mathscr{R}$. This is shown very extensively by the large list of distinct quasi-varieties of iteration theories examined in [3]. (A quasi-variety of iteration theories corresponds to the collection of FLR_0 structures in which some list of consequences holds.) We highlight here just the situation for our three basic classes.

THEOREM 9.

$$\Gamma \models_{\mathbf{Wk}} \phi \Longrightarrow \Gamma \models_{\mathbf{Mon}} \phi \Longrightarrow \Gamma \models_{\mathbf{Cont}} \phi,$$

but neither of the reverse implications hold.

PROOF. Since $\mathbf{Cont} \subseteq \mathbf{Mon} \subseteq \mathbf{Wk}$, the implications are obvious. The first implication cannot be reversed because *the Recursion Inference rule is not sound for weak structures*, as shown in [9], Theorem 13.2. The non-reversibility of the second implication actually follows from Theorem 13.32 in the same paper, but the following argument is more direct and provides a concrete countermodel.

For any unary function symbols f and g,

$$(12) \quad \{ f(\bot) = g(\bot), \forall x\, (f(g(x)) = g(f(x))) \} \vDash_{\mathbf{Cont}} \\ x \texttt{ where } \{x = f(x)\} \; = \; x \texttt{ where } \{x = g(x)\}\,.$$

(This consequence holds because under the hypotheses,

$$f(f(\bot)) = f(g(\bot)) = g(f(\bot)) = g(g(\bot))$$

and similarly $f^n(\bot) = g^n(\bot)$; and the fixed points in the conclusion are just the suprema of these iterates.)

This consequence (12) does not hold for $\vDash_{\mathbf{Mon}}$, however. Let Y be the poset pictured in Figure 1, consisting of a copy of ω with a diamond at the top. Let l and r be the monotone functions on Y defined by:

- $l(n) = r(n) = n + 1$ for $n \in \omega$.
- $l(\omega) = L$, $r(\omega) = R$.
- $l(L) = L$, $r(R) = R$, $l(R) = r(L) = \top$.
- $l(\top) = r(\top) = \top$.

Check that l and r satisfy the hypotheses of consequence (12), but not the conclusion. ⊣

We believe that a complete axiomatization of any one of these consequence relations would provide deep insight into the nature of (least-fixed-point) recursion. The next two theorems show that such an axiomatization is at least feasible for $\vDash_{\mathbf{Wk}}$, but impossible for $\vDash_{\mathbf{Cont}}$. The former fact is one of the main motivations for introducing the weak interpretations.

THEOREM 10. *The relation (for finite sets of formulas Γ and formulas ϕ) of $\Gamma \vDash_{\mathbf{Wk}} \phi$ is recursively enumerable.*

SKETCH OF PROOF. Associate effectively with each Γ and ϕ first order sentences Γ^*, ϕ^* in an expanded signature such that $\Gamma \vDash_{\mathbf{Wk}} \phi$ is equivalent to $\Gamma^* \vDash \phi^*$, i.e., ordinary predicate logic consequence. Then appeal to the Completeness Theorem of Predicate Logic.

In slightly more detail, first add a binary predicate $\leq$ and the set of sentences which insure that $\leq$ is a partial order with a minimum element, that all the function symbols have monotone interpretations, and that all systems of equations of FLR_0 expressions have least fixed points. This is an infinite but r.e. set of sentences. Then, translate all the formulas in Γ and ϕ by the sentences which say that the least-fixed-points of the systems on both sides in each identity actually yield the same value. The equivalence of $\Gamma \vDash_{\mathbf{Wk}} \phi$ and $\Gamma^* \vDash \phi^*$ is then immediate and the fact that Γ^* is infinite poses no problem, since it is r.e. ⊣

Say that a term or formula of FLR_0 is **explicit** if it makes no use of `where`, and **semi-explicit** if `where` is only present in occurrences of $\bot$. The consequence relation for continuous monotone structures is undecidable, even for hypotheses that only use recursion trivially:

THEOREM 11. *The relation $\Gamma \vDash_{\mathbf{Cont}} \phi$ for Γ restricted to a finite set of semi-explicit identities is complete Π^0_2.*

PROOF. First we sketch a proof that $\Gamma \models_{\mathbf{Cont}} \phi$ is in Π^0_2. Fix the signature τ. Let $\leq_\Gamma$ be the smallest preorder on the semi-explicit terms (open and closed) of $\mathrm{FLR}_0(\tau)$ such that every function symbol is $\leq_\Gamma$-monotone, $\perp$ is the $\leq_\Gamma$-least element, and such that $A \leq_\Gamma B$ and $B \leq_\Gamma A$ for each $(A = B) \in \Gamma$. Let $=_\Gamma$ be the induced equivalence relation. Both of these relations are r.e., since they contain only the pairs forced into them by the above conditions.

Now take the collection of *closed* terms and mod out by $=_\Gamma$, so that $\leq_\Gamma$ becomes a partial order. Further, take the free directed-complete poset F over the resulting partial order, and extend the interpretations of the function symbols to F by continuity. This construction produces a continuous, monotone structure for the signature τ which satisfies Γ. One can show that for closed terms E and M, $\Gamma \models_{\mathbf{Cont}} E = M$ if and only if $E = M$ holds in F.

To decide whether this happens in a Π^0_2 way, first put E and M in simplified form by the Simplification Lemma. Then E and M each have a natural sequence of iterates $E^{(i)}$ and $M^{(i)}$ for $i \in \mathbf{N}$. Each iterate is a semi-explicit term, with $E^{(0)}$, for example, equal to the head term E_0 of E with $\perp$ substituted for each recursion variable. It is not difficult to show that $E = M$ holds in F if and only if

$$\forall i\, \forall j\, \exists k\, \exists l\, (E^{(i)} \leq_\Gamma M^{(k)} \text{ and } M^{(j)} \leq_\Gamma E^{(l)}).$$

For completeness, choose a recursive function g such that $\forall x \exists y\, g(x, y, z) = 0$ is a complete Π^0_2 relation on z. We will construct a finite set of semi-explicit identities Γ and a recursive sequence of identities ϕ_n such that

$$\forall x\, \exists y\, g(x, y, z) = 0 \iff \Gamma \models_{\mathbf{Cont}} \phi_z.$$

The universe of the intended model for Γ consists of the union of three separate posets with their bottom elements identified: the flat poset on $\mathbf{N}$, the ordinal $\omega + 1$, and the poset of ternary partial functions on $\mathbf{N}$ with its usual order. We put a number of function symbols into the signature and identities into Γ to enforce key properties of these components. First, add symbols $0, S, P$, and Z for the usual 0, successor, predecessor, and characteristic function of $\{0\}$ on $\mathbf{N}$, where 0 is used for "true" and $1 \equiv S(0)$ for "false." In the intended model, these functions take on the value $\perp$ when their argument is not in $\mathbf{N}$. Similarly, add a ternary conditional symbol $\texttt{if} \cdot \texttt{then} \cdot \texttt{else} \cdot$. Let Γ contain the following identities:

(1) $Z(0) = 0,\ Z(1) = 1,\ Z(SSx) = Z(Sx),\ Z(\perp) = \perp$.
(2) $S(\perp) = \perp,\ P(0) = 0,\ P(1) = 0,\ P(SSx) = Sx$.
(3) $\texttt{if}\, 0\, \texttt{then}\, x\, \texttt{else}\, y = x,\ \texttt{if}\, 1\, \texttt{then}\, x\, \texttt{else}\, y = y,\ \texttt{if}\, \perp \texttt{then}\, x\, \texttt{else}\, y = \perp$.

Let Δ_n be the usual numeral for the natural number n, i.e., S applied n times to 0. The above identities guarantee

LEMMA 12. *Let $D = (\Phi, \Lambda)$ be any* FLR_0 *structure satisfying Γ. If any two terms have distinct denotations, then the denotations of the Δ_n are all distinct from each other and $\perp$ and the map $n \mapsto \Lambda()\, \Delta_n$ is an isomorphism of the structure of the natural numbers* $(\mathbf{N}, 0, S, P, Z)$.

Turning next to the copy of $\omega + 1$, add a binary function symbol **inc** intended to stand for the function

$$\mathbf{inc}(\alpha, b) = \begin{cases} \alpha & \text{if } \alpha = \omega \text{ or } b = \perp \\ \alpha + 1 & \text{otherwise,} \end{cases}$$

for α an ordinal in $\omega + 1$. Also add a function symbol W for the natural map from $\mathbf{N}$ to ω taking n to the nth ordinal. Add to Γ the equations

$$\mathbf{inc}(x, \bot) = x,\ W(0) = \bot,\ W(Sx) = \mathbf{inc}(W(x), 0).$$

Write W_ω to abbreviate the term w `where` $\{w = \mathbf{inc}(w, 0)\}$. These identities guarantee that in any continuous structure, the denotation of W_ω will be the sup of the $W(\Delta_n)$, i.e., it acts as a numeral for ω. Note that these equations do not enforce that the copy of $\omega + 1$ is standard, as the previous group did for $\mathbf{N}$ — there may be non-trivial models of Γ in which $W[\mathbf{N}]$ is a finite linear order.

The final component of the intended model is handled by adding a function symbol **ap** of four arguments, to denote the application of a ternary partial function to three natural numbers. The only identities Γ will need for **ap** enforce that it is strict in each of its arguments:

$$\mathbf{ap}(\bot, x, y, z) = \bot,\ \mathbf{ap}(w, \bot, y, z) = \bot,\ \text{etc.}$$

Returning to the original recursive function g selected, choose a system of Herbrand-Gödel-Kleene equations which prove that g is recursive. Add the necessary function symbols to the signature and the exact system of equations to Γ. In the intended model, these function symbols denote the (partial) recursive functions (on the $\mathbf{N}$-component) that they define, and take the value $\bot$ for arguments not in $\mathbf{N}$. Since the rules of the H-G-K system are substitution and replacement which are valid in our proof system, and the intended model satisfies Γ, we have

$$g(l, n, m) = w \iff \Gamma \vDash g(\Delta_l, \Delta_n, \Delta_m) = \Delta_w. \tag{13}$$

Now we are ready to define the FLR_0 expressions which will capture the relation $\forall x\, \exists y\, g(x, y, z) = 0$. Add three more symbols c, d, and e to the signature. For c, add the identity

$$\mathbf{ap}(c(v), x, y, z) = \texttt{ if } Z(g(x, y, z)) \texttt{ then } 0 \texttt{ else } \mathbf{ap}(v, x, Sy, z)$$

to Γ. This identity makes c an operator on ternary partial functions. Intuitively, the fixed point of c will be a partial function $\hat{v}(x, y, z)$ satisfying

$$\hat{v}(x, y, z) = \begin{cases} 0 & \text{if } g(x, y, z) = 0 \\ \hat{v}(x, y + 1, z) & \text{otherwise.} \end{cases}$$

In other words, $\mathbf{ap}(v, x, 0, z)$ `where` $\{v = c(v)\}$ will be 0 (true) if there is a y such that $g(x, y, z) = 0$ and $\bot$ otherwise.

Next, add the following equation defining d to Γ:

$$\mathbf{ap}(d(u, v), x, 0, z) =$$
$$\texttt{if } \mathbf{ap}(v, x, 0, z) \texttt{ then } (\texttt{ if } Zx \texttt{ then } 0 \texttt{ else } \mathbf{ap}(u, Px, 0, z)) \texttt{ else } 1.$$

Similarly to c,

$$\mathbf{ap}(u, x, 0, z) \texttt{ where } \{u = d(u, v), v = c(v)\}$$

will be 0 if for each $x' \leq x$ there exists y s.t. $g(x', y, z) = 0$ and $\bot$ otherwise. (Note that the "1" in the final `else` clause can never be reached, since $\hat{v}(x, 0, z)$ is either 0 or $\bot$.)

Finally, add the following equations for e:

$$e(W(0), u, z) = \bot$$
$$e(W(Sn), u, z) = \mathbf{inc}(e(W(n), u, z), \mathbf{ap}(u, n, 0, z)).$$

We will of course look at what e does in a context where u is the fixed point $\hat{u} = d(\hat{u}, \hat{v})$. The idea here is that we are defining e "by induction" on the finite ordinals so that for fixed z its values will be unbounded if and only if for every n, $\exists y\, g(n, y, z) = 0$. In other words, let ϕ_z be the formula

$$e(W_\omega, u, \Delta_z) \texttt{ where } \{u = d(u, v), v = c(v)\} \ = \ W_\omega,$$

and finish off the proof with:

Claim. For every natural number z,

$$\forall x\, \exists y\, g(x, y, z) = 0 \iff \Gamma \vDash_{\mathbf{Cont}} \phi_z.$$

One direction of the claim is easy: if $\exists x \forall y\, g(x, y, z) \neq 0$, then the intended model does not satisfy ϕ_z even though it does satisfy Γ. On the intended model, $\hat{u}(n, 0, z) = \bot$ for n greater than the offending x, so $e(\alpha', \hat{u}, z) = e(\alpha, \hat{u}, z)$ for ordinals α greater than x, so $e(\omega, \hat{u}, z)$ is the xth ordinal and not equal to ω.

On the other hand, suppose that in fact $\forall x \exists y\, g(x, y, z) = 0$. On a non-trivial model of Γ, the numerals Δ_n are isomorphic to the natural numbers and g computes correctly, so in fact for every x there is a y such that $g(\Delta_x, \Delta_y, \Delta_z) = 0$. Let $v^0 = \bot, v^1 = c(\bot), v^2 = c(v^1), \ldots$ be the iterates for the fixed point $\sup_n v^n = \hat{v} = c(\hat{v})$. The equation for c in Γ means $\mathbf{ap}(c(v^0), \Delta_x, \Delta_y, \Delta_z) = 0$, so in turn $\mathbf{ap}(c(v^1), \Delta_x, \Delta_{(y-1)}, \Delta_z) = 0$. By induction and continuity of $\mathbf{ap}$, we get $\mathbf{ap}(\hat{v}, \Delta_x, 0, \Delta_z) = 0$ for each x. Similarly, using the equation for d and induction, we calculate that $\mathbf{ap}(\hat{u}, \Delta_x, 0, \Delta_z) = 0$ for all x.

Therefore, by induction on n, $e(W(n), \hat{u}, \Delta_z) = W(n)$ since

$$e(W(Sn), \hat{u}, \Delta_z) = \mathbf{inc}(W(n), 0) = W(Sn)$$

by the equations for e, $\mathbf{inc}$, and W. As mentioned above, the term W_ω denotes the sup of the $W(n)$ in every model of Γ. Hence the continuity of e means that $e(W_\omega, \hat{u}, \Delta_z) = W_\omega$ as desired. ⊣

The complexity of the general consequence relation $\Gamma \vDash_{\mathbf{Cont}} \phi$ for arbitrary finite sets of identities Γ is much greater; it appears to be at least Δ^0_ω, but we have not pinned this down precisely. In any case, the next natural question to ask is

Question 13. *What is the degree of unsolvability of* $\Gamma \vDash_{\mathbf{Mon}} \phi$, *for arbitrary or for explicit* Γ? *Is either r.e.*?

Because of Theorem 11, the provability relation $\Gamma \vdash \phi$ axiomatized in Table 1 is clearly incomplete for $\vDash_{\mathbf{Cont}}$. However, it's also incomplete for $\vDash_{\mathbf{Mon}}$:

Proposition 14. *The consequence*

$$f(\bot) = \bot \vDash x \texttt{ where } \{x = f(x)\} = \bot \tag{14}$$

holds in $\mathbf{Wk}$ *(and hence* $\mathbf{Mon}$ *and* $\mathbf{Cont}$*), but is not provable.*

This follows from Section 5.6 of [3], but it is not difficult to see directly; the smallest countermodel is an intensional FLR_0 structure with just two individuals,

$\bot$ and $\top$, and just two unary transformations, which both fix both individuals but which are assigned distinct fixed points.

Although we have no real evidence that this fact captures the only obstacle to completeness, it is nevertheless tempting to ask

QUESTION 15. *Augment* $\vdash$ *with an axiom scheme similar to* (14) *for all "finitely terminating" recursions.*[3] *Is the resulting proof system complete for* $\vDash_{\mathbf{Mon}}$?

Looked at another way, the incompleteness could result from the current axiomatization being insufficient to handle recursion in the hypotheses.

QUESTION 16. *Restrict* Γ *to be a finite set of explicit identities. In this case, does* $\Gamma \vDash_{\mathbf{Mon}} \phi$ *imply that* $\Gamma \vdash \phi$?

Of course, completely open but perhaps more fruitful in the light of the recursive enumerability of Theorem 10 is:

QUESTION 17. *Find a useful, complete axiomatization for* $\vDash_{\mathbf{Wk}}$.[4]

One attraction of this problem is that a solution provides a sound, albeit incomplete, axiomatization for $\vDash_{\mathbf{Mon}}$. Furthermore, adding the Recursion Inference rule to such a system would give a reasonable candidate for a complete proof system for $\vDash_{\mathbf{Mon}}$.

§5. **Adding the conditional.** The proof of Theorem 11 made heavy use of a natural conditional construct, which in fact the full language FLR of [18, 17] includes as a *logical* symbol, along with parameter passing. In this section, we add this conditional to FLR$_0$, and show that decidability and completeness still hold for this larger fragment of FLR. Section 12.4 of [1] is similar in flavor to this section; the primary difference seems to be that [1] is concerned with a finite list of specific predicates, whereas we wish to axiomatize here a general construct in which *any* FLR$_0$ expression can be substituted as the *condition*.

For a signature τ, the terms E of the Equational Logic of Recursion ELR(τ) are given by

$$E := \mathtt{ff} \mid \mathtt{tt} \mid x \mid f(E_1, \dots, E_n) \mid E_0 \;\mathtt{where}\; \{\vec{x} = \vec{E}\} \mid \mathtt{if}\; E_0 \;\mathtt{then}\; E_1 \;\mathtt{else}\; E_2 \,.$$

$\mathtt{ff}$ and $\mathtt{tt}$ are new logical constant symbols, and $\mathtt{if} \cdot \mathtt{then} \cdot \mathtt{else} \cdot$ is a new logical ternary symbol; all other syntactic notions are the same for ELR as FLR$_0$. An ELR(τ) *structure* is just an FLR$_0(\tau \cup \{\mathtt{ff}, \mathtt{tt}, \mathtt{if} \cdot \mathtt{then} \cdot \mathtt{else} \cdot\})$ structure (Φ, Λ) satisfying

(1) $\Lambda()\,\mathtt{ff}, \Lambda()\,\mathtt{tt}$, and $\Lambda()\,\bot$ are all distinct.

(2) $\Lambda(\vec{x})\,(\,\mathtt{if}\;\mathtt{tt}\;\mathtt{then}\;E_1\;\mathtt{else}\;E_2\,) = \Lambda(\vec{x})\,E_1$ for all terms E_1 and E_2 and appropriate lists $\vec{x}$.

(3) Similarly, $\Lambda(\vec{x})\,(\,\mathtt{if}\;\mathtt{ff}\;\mathtt{then}\;E_1\;\mathtt{else}\;E_2\,) = \Lambda(\vec{x})\,E_2$.

(4) $\Lambda(\vec{x})\,(\,\mathtt{if}\;\bot\;\mathtt{then}\;E_1\;\mathtt{else}\;E_2\,) = \Lambda(\vec{x})\,\bot$, i.e., the conditional is *strict* in its first argument.

---

[3]It turns out it is only necessary to axiomatize explicitly the recursions which terminate at stage two; similar properties for recursions that close at later finite stages then become provable.

[4]This question is intentionally imprecise. The stipulation "useful" is included to avoid trivial axiomatizations supplied by the proof of Theorem 10 together with the Craig Lemma, which states that every r.e. set of sentences closed under (predicate) logical consequence is the set of consequences of a primitive recursive set of sentences.

Note that no assumption is placed on $\texttt{if}\ E\ \texttt{then}\ \cdot\ \texttt{else}\ \cdot$ in case E does not have the same value as $\texttt{ff}$, $\texttt{tt}$, or $\perp$. An ELR structure is called *monotone* if it is monotone as an FLR_0 structure.

For an example of a monotone structure, consider the poset $D = \mathbf{N} \rightharpoonup \mathbf{N}$ of partial functions on the natural numbers with the usual ordering. The signature and the monotone functions interpreting the function symbols can be arbitrary. $\perp$, of course, denotes the totally undefined function, the least element of D. For consistency with the proof of Theorem 11, interpret $\texttt{ff}$ and $\texttt{tt}$ as the constant functions equal to 1 and 0 on all natural numbers, respectively. Finally, interpret $\texttt{if}\ \cdot\ \texttt{then}\ \cdot\ \texttt{else}\ \cdot$ by the monotone operation on D which takes p, q, and r to the partial function c defined by

$$c(n) = \begin{cases} q(n) & \text{if } p(n) = 0, \\ r(n) & \text{if } p(n) > 0, \\ \text{undefined} & \text{if } p(n) \text{ is undefined.} \end{cases}$$

It is straightforward to check that the conditions for an ELR structure are satisfied. This is the "usual" conditional, subject to our convention for true and false: for example, if Z denotes the characteristic function of $\{0\}$ as above, then the closed term

$$\texttt{if}\ \mathsf{Z}\ \texttt{then}\ E_1\ \texttt{else}\ E_2$$

denotes the function whose value at 0 is that of E_1 and whose value everywhere else is equal to E_2. As before, ELR is a weak language for talking about the properties of D and this conditional, since it cannot refer to the natural numbers themselves.

Nevertheless, with an appropriate modest signature interpreted by the usual initial functions and closure operations, the closed terms of ELR will denote every partial recursive function on $\mathbf{N}$. Extending the above example, suppose that $\mathsf{pr}(\cdot,\cdot)$ denotes the pairing operation which takes p and q to the function $\langle n, m\rangle \mapsto \langle p(n), q(m)\rangle$ (relative to some standard pairing function $\langle\cdot,\cdot\rangle$). We need constant symbols for the two component functions (projections) as well: fst and sec, with $\mathsf{fst}_{\mathbf{A}}\colon \langle n, m\rangle \mapsto n$ and similarly for $\mathsf{sec}_{\mathbf{A}}$. Let id, Z, S and P be constant symbols for the usual identity, zero, successor and predecessor functions, and let $\mathsf{ap}_1(\cdot,\cdot)$ and $\mathsf{ap}_2(\cdot,\cdot,\cdot)$ denote unary and binary application of one partial function to others: $(\mathsf{ap}_2)_{\mathbf{A}}(p,q,r)$ is the function $n \mapsto p(\langle q(n), r(n)\rangle)$. Then the following term defines the ordinary sum of two natural numbers, i.e., the function $\langle n, m\rangle \mapsto n + m$:

$$x\ \texttt{where}\ \{x = \texttt{if}\ \mathsf{ap}_1(\mathsf{Z},\mathsf{sec})\ \texttt{then}\ \mathsf{fst}\ \texttt{else}\ \mathsf{ap}_1(\mathsf{S}, \mathsf{ap}_2(x, \mathsf{fst}, \mathsf{ap}_1(\mathsf{P},\mathsf{sec})))\}\,.$$

The corresponding, more apt, term in FLR would simply be

$$s\ \texttt{where}\ \{s(n,m) = \texttt{if}\ m = 0\ \texttt{then}\ n\ \texttt{else}\ \mathsf{S}(s(n, \mathsf{P}(m)))\}\,.$$

We conclude with the promised completeness and decidability result for ELR. An identity between two terms of ELR is **cond-standard** if it holds in all monotone interpretations. Define the provability relation $\Gamma \vdash_c \phi$ by the same induction as $\vdash$, except replace the Recursion Inference rule with the Bottom Recursion rule (below), and add the following axioms:

(C1) $\Gamma \vdash_c \texttt{if}\ \texttt{tt}\ \texttt{then}\ E_1\ \texttt{else}\ E_2 = E_1$.
(C2) $\Gamma \vdash_c \texttt{if}\ \texttt{ff}\ \texttt{then}\ E_1\ \texttt{else}\ E_2 = E_2$.
(C3) $\Gamma \vdash_c \texttt{if}\ \perp\ \texttt{then}\ E_1\ \texttt{else}\ E_2 = \perp$.

(R′4) (Bottom Recursion rule) Suppose we are given Γ and ELR terms

$$A \equiv A_0 \texttt{ where } \{x_1 = A_1, \dots , x_n = A_n\} \quad \text{and}$$
$$B \equiv B_0 \texttt{ where } \{y_1 = B_1, \dots , y_m = B_m\},$$

where no x_i occurs in B, no y_j occurs in A, and none of the variables x_i or y_j occur free in Γ. Suppose also that there is a set of equations Σ each of the form $x_i = y_j$, $x_i = \bot$ or $y_j = \bot$ such that

- $\Gamma, \Sigma \vdash_c A_0 = B_0$,
- $\Gamma, \Sigma \vdash_c A_i = B_j$ for each $(x_i = y_j) \in \Sigma$,
- $\Gamma, \Sigma \vdash_c A_i = \bot$ for each $(x_i = \bot) \in \Sigma$, and
- $\Gamma, \Sigma \vdash_c B_j = \bot$ for each $(y_j = \bot) \in \Sigma$.

Then $\Gamma \vdash_c A = B$.

The soundness of $\vdash_c$ for monotone structures of ELR is another exercise in least-fixed-point recursion, similar to Proposition 1 (Soundness of $\vdash$). Also note that $\vdash_c$ is strictly stronger than $\vdash$: if $\Gamma \vdash \phi$, then $\Gamma \vdash_c \phi$.

Theorem 18 (Completeness/Decidability for ELR). *An identity $A = B$ is cond-standard if and only if $\vdash_c A = B$. Moreover, this common validity is decidable.*

By the Simplification Lemma 3 for FLR_0, every closed ELR term A is provably equal to $A' \equiv x_a \texttt{ where } \{x_1 = A_1, \dots , x_n = A_n\}$, where each A_i is either x_i, f(variables), $\texttt{ff}$, $\texttt{tt}$, or $\texttt{if } x_j \texttt{ then } x_k \texttt{ else } x_l$. We need to strengthen this simplified form. Call each variable x_i a *bottom, functional, true, false, or conditional* variable, respectively, according to which case A_i corresponds to. In the case of a conditional variable x_i, the variable x_j is called the *immediate specifier* of x_i. Say that the variable x_m is a **specifier** of x_i if it is in the transitive closure of the “immediate specifier” relation from x_i. Thus, there is a natural sequence $x_{j_1}, x_{j_2}, \dots , x_m$ of specifiers of x_i, with each variable being the immediate specifier of the preceding one. This sequence of course ends if a non-conditional variable is reached. So every conditional variable has at most one non-conditional specifier. Finally, say that a term A' is **unconditionally simple** if every conditional variable in A' has a *functional* specifier.

Lemma 19 (Unconditional Simplification). *Every closed term of* ELR *is provably equivalent to an unconditionally simple one.*

Proof. First, put the closed term into simplified form A' as just described. Next, eliminate any conditional variable x_i with no non-conditional specifier as follows. Since there are only finitely many variables, the only way this can happen is for some variable to be a specifier of itself. Suppose it is x_1 (the order of the equations in the `where` clause is irrelevant), so that there is a sequence of variables $x_1, x_2, \dots , x_k$, with each variable the immediate specifier of the preceding one and x_1 the immediate specifier of x_k. Then all of the variables in this loop will come out to be $\bot$. More technically, the Bottom Recursion rule and axiom (C3) for conditionals shows that A' is equal to

$$y_a \texttt{ where } \{y_1 = y_1, y_2 = y_2, \dots , y_k = y_k, y_{k+1} = A_{k+1}(\vec{y}), \dots , y_n = A_n(\vec{y})\}.$$

The set of equations Σ used in the bottom recursion rule consists of $x_j = y_j$ for each $j \le n$ together with $x_j = \bot$ and $y_j = \bot$ for each $j \le k$.

Finally, repeatedly use conditional axioms (C1–3) to eliminate conditional variables whose immediate specifier is a true, false, or bottom variable, respectively. ⊣

Next, adjust the notion of a reasonably free structure: (Φ, Λ) is *cond-free* if the following properties hold for arbitrary distinct function symbols f and g:

(1) $\bar{f}$ is injective, and its image does not contain $\Lambda()\,\bot$, $\Lambda()\,\mathtt{ff}$, or $\Lambda()\,\mathtt{tt}$.

(2) $\bar{f}$ and $\bar{g}$ have disjoint images.

(3) The extension of the conditional $\Lambda()\ \mathtt{if}\ a\ \mathtt{then}\ b\ \mathtt{else}\ c$ is injective except for a equal to $\Lambda()\,\bot$, $\Lambda()\,\mathtt{ff}$, or $\Lambda()\,\mathtt{tt}$.

(4) The image of $\bar{f}$ is disjoint from

$$\{\,\Lambda()\ \mathtt{if}\ a\ \mathtt{then}\ b\ \mathtt{else}\ c\ \mid a \notin \{\Lambda()\,\bot, \Lambda()\,\mathtt{ff}, \Lambda()\,\mathtt{tt}\}\,\}\,.$$

Furthermore, this latter set does not contain $\Lambda()\,\bot$, $\Lambda()\,\mathtt{ff}$, or $\Lambda()\,\mathtt{tt}$.

LEMMA 20. *Suppose $A' \equiv x_a\ \mathtt{where}\ \{\vec{x} = \vec{A}\}$ is unconditionally simple. For each variable x_i in A', let $\hat{x}_i$ be the denotation $\Lambda()\ x_i\ \mathtt{where}\ \{\vec{x} = \vec{A}\}$ in a cond-free* ELR *structure. Then $\hat{x}_i$ equals $\Lambda()\,\bot$, $\Lambda()\,\mathtt{ff}$, or $\Lambda()\,\mathtt{tt}$ if and only if x_i is a bottom, false, or true variable, respectively.*

PROOF. The "if" direction is trivial, of course; and for functional variables, the "only if" direction is easy as well: If x_i is functional, then by the Fixed Point axiom (R3),

$$\hat{x}_i = f(x_{j_1}, \dots, x_{j_n})\ \mathtt{where}\ \{\vec{x} = \vec{A}\}\,.$$

But the extension of each function symbol does not contain 0, 1, or $\bot$ in its image.

The argument for conditional variables is similar, but goes by induction: if x_j is the immediate specifier of x_i and $\hat{x}_j$ is not 0, 1, or $\bot$, then $\hat{x}_i$ is not 0, 1 or $\bot$ either. The base case is provided by the fact that every conditional variable has a functional specifier, since A' is in unconditionally simple form. ⊣

This collection of lemmas ensures that the same outline that worked for FLR_0 goes through for the extended language.

PROOF OF COMPLETENESS/DECIDABILITY FOR ELR. There is essentially nothing left to do, except show that there is a cond-free monotone structure. Given $A = B$ which holds in such a model, convert both sides to A' and B' in unconditionally simple form, an effective operation. Apply the same algorithm for constructing an application of the Recursion Inference rule as used for FLR_0 (considering $\mathtt{if}\ \cdot\ \mathtt{then}\cdot\mathtt{else}\cdot$ to be an ordinary function symbol). The immediately preceding lemma together with the requirements for being cond-free guarantees that the algorithm will succeed if A and B really do have the same denotation (and fail otherwise, completing the decision procedure).

Finally, to build a cond-free monotone structure for $\mathrm{ELR}(\tau)$, choose a reasonably free structure for $\mathrm{FLR}_0(\tau \cup \{F, T, C\})$ where F and T are new constant symbols (whose values will interpret $\mathtt{ff}$ and $\mathtt{tt}$), and C is a new ternary function symbol. Let $\bot$ be the least element of the complete poset the reasonably free interpretation is based on. Now interpret the conditional construction by

$$\texttt{if } a \texttt{ then } b \texttt{ else } c = \begin{cases} \bot & \text{if } a = \bot \\ b & \text{if } a = T \\ c & \text{if } a = F \\ C(a,b,c) & \text{otherwise.} \end{cases}$$

Check that this defines a monotone function and that the conditions for a cond-free structure are satisfied. ⊣

Appendix. Equivalence between FLR$_0$ structures and iteration theories. This appendix is devoted to an outline of the proof of Theorem 5 from Section 2, which states that the categories of FLR$_0$ structures and iteration theories (with the corresponding appropriate notions of homomorphisms) are equivalent. Most of the work of this proof consists in building the functors in each direction that effect the equivalence.

5.1. From FLR$_0$ **structures to iteration theories.** Given a normal FLR$_0$ structure $R = (\Phi, \Lambda)$, build an iteration theory T_R as follows:

(1) The set of morphisms in T_R from n to k is $T_R(n,k) = (\Phi_k)^n$.

Note: we take $(\Phi_k)^1 = \Phi_k$ here, and $(\Phi_k)^0$ to be some singleton set $\{*_k\}$.

(2) $\mathrm{id}_n = (\Lambda(\vec{\mathsf{v}})\mathsf{v}_1, \dots, \Lambda(\vec{\mathsf{v}})\mathsf{v}_n)$.

(3) For $1 \le i \le n$, $i_n = \Lambda(\mathsf{v}_1, \dots, \mathsf{v}_n)\mathsf{v}_i$.

(4) Given $f = (f_1, \dots, f_n) \colon n \to m$ and $g = (g_1, \dots, g_m) \colon m \to k$, we set $f \cdot g \colon n \to k$ to be the n-tuple whose ith entry is

$$\Lambda(\mathsf{v}_1, \dots, \mathsf{v}_k) f_i(g_1(\vec{\mathsf{v}}), \dots, g_m(\vec{\mathsf{v}})). \tag{15}$$

(When $n = 0$, $f = *_n$ and we set $f \cdot g = *_k$. We'll ignore the case of $n = 0$ in what follows, however.)

(5) Given $f = (f_1, \dots, f_n) \colon n \to n + p$, we set $f^\dagger \colon n \to p$ to be the n-tuple whose ith entry is

$$\Lambda(y_1, \dots, y_p)\, x_i \texttt{ where } \{x_1 = f_1(\vec{x}, \vec{y}), \dots x_n = f_n(\vec{x}, \vec{y})\}\,). \tag{16}$$

Here $x_1 = \mathsf{v}_1, \dots, x_n = \mathsf{v}_n, y_1 = \mathsf{v}_{n+1}, \dots, y_p = \mathsf{v}_{n+p}$.

Essentially, the logical rules (L1–7) of Table 1 guarantee that T_Φ is a category under this composition. One must show that the iteration theory axioms of Table 2 hold in the resulting theory T_Φ. The two sides of the Parameter Identity translate to syntactically identical tuples of FLR$_0$ expressions, and so are equal. The remaining axioms correspond as in the table above.

So far, we have not considered maps between FLR$_0$ structures. However, there is a natural notion of (FLR$_0$) **homomorphism**: Given two FLR$_0$ structures $P = (\Phi, \Lambda_P)$ and $Q = (\Psi, \Lambda_Q)$, a homomorphism from P to Q is an arity-preserving map σ from Φ to Ψ which respects the denotation maps, as follows: σ can be extended by substitution (on the function symbols, which are just the elements of Φ, since we are considering structures for the empty signature) to a map from FLR$_0(\Phi)$ to FLR$_0(\Psi)$. This extended map must satisfy

$$\sigma(\Lambda_P(\vec{x})\, E) = \Lambda_Q(\vec{x})\, (\sigma(E))$$

for all terms E of $\mathrm{FLR}_0(\Phi)$. This requirement guarantees that σ preserves composition and recursion, much the way that a homomorphism of rings must preserve addition and multiplication.

The correspondence $R \mapsto T_R$ now extends to a functor on the respective categories of FLR_0 structures and iteration theories. Any FLR_0 homomorphism $\sigma\colon P \to Q$ as above determines a theory morphism $T_\sigma\colon T_P \to T_Q$, as follows. The object map of T_σ is the identity, the map on morphisms is just σ, and the Λ-preserving property of σ ensures that T_σ preserves the theory operations, the $\dagger$-operation and the distinguished morphisms.

5.2. From iteration theories to FLR_0 structures. In the other direction, any iteration theory T gives rise to an FLR_0 structure $(\Phi, \Lambda) = (\Phi_T, \Lambda_T)$ in the following way:

- Φ is given by $\Phi_n = T(1, n)$.
- $\Lambda(x_1, \dots, x_n)x_i = i_n$.
- $\Lambda(x_1, \dots, x_m)f(E_1, \dots, E_n) = f \cdot \langle \Lambda(\vec{x})E_1, \dots, \Lambda(\vec{x})E_n\rangle$.
- $\Lambda(x_1, \dots, x_m)\, E_0$ `where` $\{y_1 = E_1, \dots, y_n = E_n\}$ is

$$\Lambda(\vec{y}, \vec{x})E_0 \cdot \langle\langle \Lambda(\vec{y}, \vec{x})E_1, \dots, \Lambda(\vec{y}, \vec{x})E_n\rangle^\dagger, \mathrm{id}_m\rangle.$$

We must make a provision here to cover the case when the sequence $\vec{y}$, $\vec{x}$ has repeated elements. We deal with this case in the following way: Let $w_1, \dots, w_k$ be the subsequence of $\vec{x}$ containing the variables which occur in $\vec{y}$. Let $\vec{z}$ be the sequence $\vec{x}$ with w_i replaced by the ith variable (in the natural order) which is not among the xs or ys. Then $\vec{y}$ and $\vec{z}$ have no overlaps, and we set $\Lambda(\vec{x})\ E_0$ `where` $\{\vec{y} = \vec{E}\}$ to

$$\Lambda(\vec{y}, \vec{z})E_0 \cdot \langle\langle \Lambda(\vec{y}, \vec{z})E_1, \dots, \Lambda(\vec{y}, \vec{z})E_n\rangle^\dagger, \mathrm{id}_p\rangle.$$

As before, the iteration theory axioms ensure that the resulting structure actually satisfies the standard identities. The details in this direction are somewhat trickier. The first critical lemma is that substitution in $\mathrm{FLR}_0(\Phi_T)$ corresponds to composition in the original iteration theory T.

LEMMA 21. *Let E be a term whose free variables are among $y_1, \dots, y_n$. If s is free for E, then*

$$\Lambda_T(x_1, \dots, x_m)\, E[s] = (\Lambda_T(y_1, \dots, y_n)\, E) \cdot \langle \Lambda_T(\vec{x})\, s(y_1), \dots, \Lambda_T(\vec{x})\, s(y_n)\rangle.$$

The compositionality conditions defining an FLR_0 structure and the logical laws (L1–7) just come from the fact that T is a category. The Head axiom (R1) is trivial from the iteration theory point of view. The Scott-Bekič and Fixed Point axioms (R2,3) correspond as they did in the other direction. The Part Replacement rule corresponds to the simple fact that $\cdot^\dagger$ is well-defined. The Permutation Identity is a simple corollary of the Commutative identity. Finally, Alphabetic Identification also follows from the Commutative Identity (via its equivalent generalized form in Proposition 5.3.26 of [1]). By Corollary 4, the standard identities hold in (Φ_T, Λ_T). Of course, the correspondence in this direction also extends to a functor in the natural way.

These constructions supply most of the proof of the main theorem of the appendix.

PROOF OF THEOREM 5. So far we have shown how to take an iteration theory T and produce an FLR_0 structure $(\Phi_T, \Lambda_T) = (\Phi(T), \Lambda(T))$ and also how to take an FLR_0 structure (Φ, Λ) and produce an iteration theory $T_{(\Phi,\Lambda)} = T(\Phi, \Lambda)$.

We want to consider the composites of these functors, and show they are both naturally isomorphic to the identity functor.

First, $(\Phi(T(\Phi, \Lambda)), \Lambda(T(\Phi, \Lambda))) = (\Phi, \Lambda)$. This exact equality hinges on our identification of X^1 with X in the definition of $T(\Phi, \Lambda)$; without that we would just have an isomorphism. Similarly, composition in this direction leaves FLR_0-homomorphisms unchanged, so there is nothing to prove for naturality.

Second, consider $U = T(\Phi(T), \Lambda(T))$. Here, $T \cong U$: The isomorphism takes a morphism $f : n \to m$ of T to the tuple $(1_n \cdot f, \dots, n_n \cdot f)$ of elements of $\Phi(T)_m = T(1, m)$. (The morphism i_n is taken to the ith projection function on n-tuples.) Furthermore, the isomorphism $T \cong U$ is natural in T, essentially because $U(1, m)$ is literally equal to $T(1, m)$ for each m, and the $T(1, m)$ together with the tupling operations $\langle \vec{\cdot} \rangle$ determine T. ⊣

Acknowledgments. Thanks to the referee for improving the exposition in this paper, to Zoltán Ésik and Stephen Bloom for their helpful discussion of the connections with their work, to Andreas Blass for his comments on more than one draft, and to Kevin Rowney for tracking down many of the references.

REFERENCES

[1] STEPHEN L. BLOOM and ZOLTÁN ÉSIK, *Iteration theories: The equational logic of iterative processes*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Berlin, 1993.

[2] ———, *Solving polynomial fixed-point equations*, ***Mathematical foundations of computer science '94***, Lecture Notes in Computer Science, no. 841, Springer-Verlag, Berlin, 1994.

[3] ———, *Some quasi-varieties of iteration theories*, ***Mathematical foundations of programming semantics '93***, Lecture Notes in Computer Science, no. 802, Springer-Verlag, Berlin, 1994.

[4] B. COURCELLE, G. KAHN, and J. VUILLEMIN, *Algorithmes d'equivalence et de reduction a des expressions minimales dans une classe d'equations recursives simples*, ***Automata, languages, and programming, 2nd colloquium*** (J. Loeckx, editor), Lecture Notes in Computer Science, no. 14, Springer-Verlag, Berlin, 1974.

[5] J. W. DE BAKKER, *Recursive procedures*, ***Mathematical Centre Tracts 24***, Mathematisch Centrum Amsterdam, Amsterdam, 1971.

[6] J. W. DE BAKKER and W. P. DE ROEVER, *A calculus for recursive program schemes*, ***Automata, languages, and programming*** (M. Nivat, editor), North-Holland, Amsterdam, 1973.

[7] C. C. ELGOT, *Monadic computation and iterative algebraic theories*, ***Logic colloquium 1973*** (J. C. Shepherdson, editor), Studies in Logic, no. 80, North-Holland, Amsterdam, 1975.

[8] Z. ÉSIK and L. BERNÁTSKY, *Scott induction and equational proofs*, ***Mathematical foundations of programming semantics '95***, Electronic Notes in Theoretical Computer Science, no. 1, 1995.

[9] ZOLTÁN ÉSIK, *Completeness of park induction*, ***Theoretical Computer Science***, to appear.

[10] ———, *Identities in iterative and rational algebraic theories*, ***Comptational Linguistics and Computer Languages***, vol. 14 (1980), pp. 183–207.

[11] L. L. IVANOV, ***Algebraic recursion theory***, Ellis Horwood Limited, Chichester, 1986.

[12] ———, *Categorial generalization of algebraic recursion theory*, ***Journal of Pure and Applied Algebra***, vol. 101 (1995), pp. 91–128.

[13] G. KAHN, *The semantics of a simple language for parallel programming*, ***Information processing 74*** (J. L. Rosenfeld, editor), North-Holland, Amsterdam, 1974.

[14] F. W. LAWVERE, *Functorial semantics of algebraic theories*, ***Proceedings of the National Academy of Sciences of the United States of America***, vol. 50 (1963), pp. 869–872.

[15] Z. Manna and J. Vuillemin, *Fixpoint approach to the theory of computation*, ***Automata, languages, and programming*** (M. Nivat, editor), North-Holland, Amsterdam, 1973.

[16] R. Milner, ***Communication and concurrency***, Prentice-Hall, 1989.

[17] Y. N. Moschovakis, *The logic of functional recursion*, to appear in ***Proceedings of the 10th International Congress of Logic, Methodology and the Philosophy of Science***, held in Florence, 1995.

[18] ———, *The formal language of recursion*, this Journal, vol. 54 (1989), pp. 1216–1252.

[19] ———, *A model of concurrency with fair merge and full recursion*, ***Information and Computation***, vol. 93 (1991), pp. 114–171.

[20] Y. N. Moschovakis and G. T. Whitney, *Powerdomains, powerstructures, and fairness*, ***Proceedings of computer science logic '94, Kasimierz, Poland*** (L. Pacholski and J. Tiuryn, editors), Lecture Notes in Computer Science, no. 933, Springer-Verlag, Berlin, 1995, pp. 382–396.

[21] D. G. Skordev, ***Computability in combinatory spaces***, Kluwer Academic Publishers, Dordrecht, 1992.

[22] J. E. Stoy, ***Denotational semantics: The Scott-Strachey approach to programming language theory***, MIT Press, Cambridge, MA, 1977.

[23] G. T. Whitney, *Recursion structures for non-determinism and concurrency*, ***Ph.D. thesis***, University of California, Los Angeles, 1994.

KLAASSTOKSEWEG 7
5443 NS HAPS, THE NETHERLANDS
E-mail: hurkens@sci.kun.nl

DEPARTMENT OF MATHEMATICS
INDIANA UNIVERSITY
BLOOMINGTON IN 47405, USA
E-mail: mcarthur@iu-math.math.indiana.edu

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF CALIFORNIA, LOS ANGELES, CA 90024, USA
E-mail: ynm@math.ucla.edu

DEPARTMENTS OF MATHEMATICS AND COMPUTER SCIENCE
INDIANA UNIVERSITY, BLOOMINGTON IN 47405, USA
E-mail: lmoss@ucs.indiana.edu

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF MICHIGAN
ANN ARBOR, MI 48109, USA
E-mail: glen@rentec.com