This is a preprint of a paper to be given at the Second Colloquium on Automata, Languages, and Programming, Saarbrucken, July 29 - August 2, 1974. A preliminary draft of this paper was entitled "A Justification of Continuations.

ON THE RELATION BETWEEN DIRECT AND CONTINUATION SEMANTICS†

John C. Reynolds

Systems and Information Science

Syracuse University

ABSTRACT: The use of continuations in the definition of programming languages has gained considerable currency recently, particularly in conjunction with the lattice-theoretic methods of D. Scott. Although continuations are apparently needed to provide a mathematical semantics for non-applicative control features, they are unnecessary for the definition of a purely applicative language, even when call-by-value occurs. This raises the question of the relationship between the direct and the continuation semantic functions for a purely applicative language. We give two theorems which specify this relationship and show that, in a precise sense, direct semantics are included in continuation semantics.

The heart of the problem is the construction of a relation which must be a fixed-point of a non-monotonic "relational functor." A general method is given for the construction of such relations between recursively defined domains.

Two Definitions of the Same Language

The use of continuations in the definition of programming languages, introduced by Morris(1) and Wadsworth,(2) has gained considerable currency recently,(3) particularly in conjunction with the lattice-theoretic methods of D. Scott.(4) Continuations are apparently needed to provide a mathematical semantics for non-applicative control features such as labels and jumps, Landin's J-operator,(5) or Reynolds' escape functions.(3) However, a purely applicative language, even including call-by-value (to the author's chagrin(3)), can be defined without using continuations. In this paper we will investigate the two kinds of definitions of such a purely applicative language, and prove that they satisfy an appropriate relationship.

The language which we consider is a variant of the lambda calculus which permits both call-by-name and call-by-value. Let V be a denumerably infinite set of variables. Then R, the set of expressions, is the minimal set satisfying:

(1) If $x \in V$, then $x \in R$.

(2) If $r_1, r_2 \in R$, then $(r_1\ r_2) \in R$.

(3) If $x \in V$ and $r \in R$, then $(\lambda x.r) \in R$.

(4) If $x \in V$ and $r \in R$, then $(\lambda_{val} x.r) \in R$.

†Work supported by Rome Air Force Development Center Contract No. 30602-72-C-0281, ARPA Contract No. DAHC04-72-C-0003, and National Science Foundation Grant GJ-41540.

Expressions of the fourth form are meant to denote functions which call their arguments by value.

Our first definition uses a typical Scott model of the lambda calculus in which some but not all domain elements are functions.(6) Let P be any domain of "primitive values." Then let D be the minimal domain satisfying the isomorphism

$$D = P + (D \to D)$$

where $\to$ denotes the formation of a domain of continuous functions, and + denotes the formation of a separated sum. More precisely, $D_1 + D_2$ is the domain

$$\{\bot, \top\} \cup \{\langle 1, x_1\rangle \mid x_1 \in D_1\} \cup \{\langle 2, x_2\rangle \mid x_2 \in D_2\} ,$$

with the partial ordering $x \sqsubseteq y$ iff

$$\begin{aligned} & x = \bot \text{ or } y = \top \\ & \text{or } x = \langle 1, x_1\rangle \text{ and } y = \langle 1, y_1\rangle \text{ and } x_1 \sqsubseteq y_1 \\ & \text{or } x = \langle 2, x_2\rangle \text{ and } y = \langle 2, y_2\rangle \text{ and } x_2 \sqsubseteq y_2 , \end{aligned}$$

We introduce the following classification, selection, and embedding functions for the lattice sum:

$$\tau \in D \to \text{Bool}$$

$$\iota_P \in D \to P \qquad \iota_F \in D \to (D \to D)$$

$$\rho_P \in P \to D \qquad \rho_F \in (D \to D) \to D$$

which satisfy

$$\iota_P \cdot \rho_P = I_P \qquad \iota_F \cdot \rho_F = I_{D\to D}$$

$$\lambda x \in D.\ \text{cond}(\tau(x), \rho_P(\iota_P(x)), \rho_F(\iota_F(x))) = I_D \quad .$$

Here Bool denotes the usual four-element domain of truth values, I_D denotes the identity function on a domain D, and cond denotes the conditional function which is doubly strict in its first argument (i.e., which maps $\bot$ into $\bot$ and $\top$ into $\top$).

If we take D to be the set of values described by our language, then the meaning of an expression is a continuous function from environments to values, where an environment is a function from variables to values. More precisely, the meaning of expressions is given by a function $M \in R \Rightarrow D^V \to D$, where the environment domain D^V is the set of functions from V to D, partially ordered by the pointwise extension of the partial ordering on D. The following equations define M for each of the cases in the syntactic definition of R:

(1) $M[x](e) = e(x)$

(2) $M[r_1\ r_2](e) = \text{cond}(\tau(M[r_1](e)), \bot_D, \iota_F(M[r_1](e))(M[r_2](e)))$

(3) $M[\lambda x.\ r](e) = \rho_F(\lambda a \in D.\ M[r][e|x|a])$

(4) $M[\lambda_{val} x.\ r](e) = \rho_F(\alpha(\lambda a \in D.\ M[r][e|x|a]))$

where $\alpha \in (D \to D) \to (D \to D)$ is the function such that $\alpha(f)(\bot) = \bot$, $\alpha(f)(\top) = \top$, and $\alpha(f)(a) = f(a)$ otherwise.

Here $[e|x|a]$ denotes the environment $\lambda y \in V.$ if $y = x$ then a else $e(y)$.

The only thing surprising about this definition is the fourth case. Essentially, we are interpreting a call-by-value function as the retraction of the corresponding call-by-name function into a doubly-strict function. (This was suggested to the author by G. Plotkin.) Note that the continuity of α depends upon the fact that $\top$ is an isolated point in the domain D, i.e., it is not the limit of any directed set which does not contain $\top$.

To motivate our second definition, consider the meaning of an expression bound to an environment (or a functional procedure bound to its arguments). Operationally, this meaning is a piece of code (more precisely, a closure) which is executed after being given a return address. When the meaning is purely applicative (and terminating), control will eventually come back to the return address along with the value of the expression, allowing further code at the return address to determine the final output of the program. But in non-applicative situations, control may never come back to the return address, so that the final output of the program is completely determined by the meaning of the expression being evaluated.

To mirror this situation mathematically, let O be a domain of <u>final outputs</u>, and E be a domain of <u>explicit values</u>, i.e., the kind of values which are passed to return addresses. Then the meaning of the code at a return address is a function from E to O called a <u>continuation</u>, and the meaning of an expression bound to an environment (or a functional procedure bound to its arguments) is a function from continuations to O called an <u>implicit value</u>. When this meaning is applicative and terminating we expect it to be a function $\lambda c \in E \to O.\ c(b)$, where b is some explicit value, but in other situations (as discussed in References 1 to 4) it may be a function which is independent of its argument.

To describe call-by-name, we must capture the idea that the arguments of functional procedures and the "values" assigned by environments are expressions which are bound to environments without being evaluated. But we have seen that the meanings of such entities are implicit values.

Thus in brief, let O be a domain of "final outputs" (whose exact nature we leave unspecified) and let C, D', and E be the minimal domains satisfying

$$C = E \to O \qquad D' = C \to O \qquad E \simeq P + (D' \to D') .$$

Then the meaning of expressions is given by a function $N \in R \Rightarrow D'^{V} \to D'$.

Again, we introduce classification, selection, and embedding functions for the lattice sum:

$$\tau' \in E \to \mathrm{Bool}$$

$$\iota'_P \in E \to P \qquad\qquad \iota'_F \in E \to (D' \to D')$$

$$\rho'_P \in P \to E \qquad\qquad \rho'_F \in (D' \to D') \to E$$

which satisfy

$$\iota'_P \cdot \rho'_P = I_P \qquad\qquad \iota'_F \cdot \rho'_F = I_{D' \to D'}$$

$$\lambda \bar{x} \in E.\ \mathrm{cond}(\tau'(x),\ \rho'_P(\iota'_P(x)),\ \rho'_F(\iota'_F(x))) = I_E$$

Then the following equations define N for each of the cases in the syntactic definition of R:

(1) $N[x](e')(c) = e'(x)(c)$

(2) $N[r_1\ r_2](e')(c) =$
$$N[r_1](e')(\lambda f \in E.\ cond(\tau'(f),\ error,\ \iota'_F(f)(N[r_2](e'))(c)))$$

(3) $N[\lambda x.\ r](e')(c) = c(\rho'_F(\lambda a' \in D'\ .\ N[r][e'|x|a']))$

(4) $N[\lambda_{val} x.\ r](e')(c) =$
$$c(\rho'_F(\lambda a' \in D'\ .\ \lambda c' \in C.\ a'(\lambda b \in E.\ N[r][e'|x|\lambda c'' \in C.\ c''(b)](c'))))$$

Here error denotes a member of O which is used in the second case to express the fact that application of a nonfunctional value will cause an immediate error stop without further computation. In the fourth case, the nature of a call-by-value function is expressed by having the function evaluate its implicit argument to obtain an explicit value b, and then bind the formal parameter to a new implicit value created from b. This is reminiscent of the definition of call-by-value used in the original Algol report.(7)

A Relation between the Two Definitions

We will later show that there exists two relations $\eta \in D \nrightarrow D'$ and $\theta \in (D \rightarrow D) \nrightarrow (D' \rightarrow D')$ with the following properties:

η: $a \mapsto a'$ if and only if

$a = \bot_D$ and $(a' = \bot_{D'}$, or $a' = \lambda c \in C.\ error)$

or $a = \top_D$ and $a' = \top_{D'}$

or $(\exists p \in P)\ a = \rho_P(p)$ and $a' = \lambda c \in C.\ c(\rho'_P(p))$

or $(\exists f \in D{\rightarrow}D,\ f' \in D'{\rightarrow}D')\ a = \rho_F(f)$ and $a' = \lambda c \in C.\ c(\rho'_F(f'))$

and θ: $f \mapsto f'$

θ: $f \mapsto f'$ if and only if

$(\forall\ a \in D,\ a' \in D')\ \eta$: $a \mapsto a'$ implies η: $f(a) \mapsto f'(a')$

It follows that if we evaluate the same expression according to our two definitions, using environments whose corresponding components are related by η, then the two definitions will give results which are related by η. More precisely:

Theorem 1 If, for all $x \in V$, η: $e(x) \mapsto e'(x)$, then, for all $r \in R$, η: $M[r](e) \mapsto N[r](e')$.

Proof: We use structural induction on R, i.e., we assume that, for all $x \in V$, η: $e(x) \mapsto e'(x)$, and prove that η: $M[r](e) \mapsto N[r](e')$ for each case in the syntactic definition of R, using an induction hypothesis that the theorem holds for the subexpressions of r.

(1) Obvious.

(2) By the induction hypothesis, η: $M[r_1](e) \mapsto N[r_1](e')$ and η: $M[r_2](e) \mapsto N[r_2](e')$. There are four subcases:

(2a) $M[r_1](e) = \bot_D$ and $N[r_1](e')(c)$ is either $\bot_O$ or error. Then $M[r_1\ r_2](e) = \bot_D$ (since cond is doubly strict) and $N[r_1\ r_2](e')(c)$ is either $\bot_O$ or error.

(2b) $M[r_1](e) = \top_D$ and $N[r_1](e')(c) = \top_O$. Similar to (2a).

(2c) $M[r_1](e) = \rho_P(p)$ and $N[r_1](e')(c) = c(\rho_P'(p))$. Then $M[r_1\ r_2](e) = \perp_D$ and $N[r_1\ r_2](e')(c) = \text{error}$.

(2d) $M[r_1](e) = \rho_F(f)$ and $N[r_1](e')(c) = c(\rho_F'(f'))$, where $\theta: f \mapsto f'$. Then $M[r_1\ r_2](e) = f(M[r_2](e))$ and $N[r_1\ r_2](e')(c) = f'(N[r_2](e'))(c)$. The rest follows from the induction hypothesis for r_2 and the property of θ.

(3) Let $f = \lambda a \in D.\ M[r][e|x|a]$ and $f' = \lambda a' \in D'.\ N[r][e'|x|a']$. It is sufficient to show that $\theta: f \mapsto f'$. But if $\eta: a \mapsto a'$, then for all $y \in V$, η: $[e|x|a](y) \mapsto [e'|x|a'](y)$, so that the induction hypothesis gives $\eta: f(a) \mapsto f'(a')$.

(4) Let f and f' be as in case (3). As before $\theta: f \mapsto f'$, but now we must show that $\theta: \alpha(f) \mapsto \alpha'(f')$, where

$$\alpha'(f') = \lambda a' \in D'.\ \lambda c' \in C.\ a'(\lambda b \in E.\ f'(\lambda c'' \in C.\ c''(b))(c'))$$

Thus suppose $\eta: a \mapsto a'$. Then $\eta: \alpha(f)(a) \mapsto \alpha'(f')(a')$ follows from the following three subcases:

(4a) $a = \perp_D$ and $a' = \perp_{D'}$, or $\lambda c \in C.$ error. Then $\alpha(f)(a) = \perp_D$ and $\alpha'(f')(a')$ is $\perp_{D'}$, or $\lambda c' \in C.$ error.

(4b) $a = \top_D$ and $a' = \top_{D'}$. Similar to (4a).

(4c) Otherwise, $\alpha(f)(a) = f(a)$ and a' must have the form $\lambda c \in C.\ c(x)$, so that $\alpha'(f')(a') = \lambda c' \in C.\ f'(\lambda c'' \in C.\ c''(x))(c') = f'(a')$. Then $\theta: f \mapsto f'$ implies $\eta: \alpha(f)(a) \mapsto \alpha'(f')(a')$.

A Retraction between the Two Definitions

Theorem 1 hardly implies that our two definitions of the same language are equivalent; indeed we cannot expect this since there are a variety of extensions of the language R which could be accommodated by the second style of definition but not the first. But at least we can go beyond Theorem 1 to show that the second definition "includes" the first, by exhibiting a pair of functions between D and D' which permit M to be expressed in terms of N.

In fact this development is only possible if the domain O of final outputs is rich enough to contain representations of all the members of the domain E of explicit values. Specifically, we will assume the existence of a retraction pair α, β between E and O, i.e., functions $\alpha \in E \to O$ and $\beta \in O \to E$ such that $\beta \cdot \alpha = I_E$. The retraction condition implies that β is doubly strict; we will also assume that $\beta(\text{error}) = \perp_E$.

Now let $\phi \in D \to D'$ and $\psi \in D' \to D$ be defined by

$$\phi = \bigsqcup_{n=0}^{\infty} \phi_n \qquad \psi = \bigsqcup_{n=0}^{\infty} \psi_n$$

$$\phi_0(a) = \perp_{D'} \qquad \psi_0(a') = \perp_D$$

$$\phi_{n+1}(a) = \text{cond}(\tau(a),\ \lambda c \in C.\ c(\rho_P'(\iota_P(a))),\ \lambda c \in C.\ c(\rho_F'(\phi_n \cdot \iota_F(a) \cdot \psi_n)))$$

$$\psi_{n+1}(a') = (\lambda b \in E.\ \text{cond}(\tau'(b),\ \rho_P(\iota_P'(b)),\ \rho_F(\psi_n \cdot \iota_F'(b) \cdot \phi_n)))\ (\beta(a'(\alpha)))$$

so that ϕ and ψ are the least solutions of the last two equations with the numerical subscripts omitted. Then:

Lemma 1 For all $a \in D$, η: $a \mapsto \phi(a)$. For all $a \in D$ and $a' \in D'$, η: $a \mapsto a'$ implies $a = \psi(a')$.

Proof: It can be shown from the construction of the recursively defined domain D that $I_D = \bigsqcup_{n=o}^{\infty} I_n$ where

$$I_0(a) = \perp_D$$
$$I_{n+1}(a) = \mathrm{cond}(\tau(a), \rho_P(\iota_P(a)), \rho_F(I_n \cdot \iota_F(a) \cdot I_n)) \ .$$

By induction on n, one can show

For all $a \in D$, η: $I_n(a) \mapsto \phi_n(a)$

For all $a \in D$ and $a' \in D'$, η: $a \mapsto a'$ implies $I_n(a) = \psi_n(a')$.

(The details are left to the reader.) The second result immediately shows that η: $a \mapsto a'$ implies $a = \psi(a')$. We will later show that η satisfies a continuity condition such that the first result gives η: $a \mapsto \phi(a)$.

Theorem 1 and Lemma 1 lead directly to:

Theorem 2 The functions ϕ, ψ are a retraction pair such that, for all $r \in R$ and $e \in D^V$, $M[r](e) = \psi(N[r](\phi \cdot e))$.

so that the semantics provided by M is included in the semantics provided by N.

Function Pairs and Domain Functors

We are left with the problem of constructing relations η and θ which satisfy the previously stated properties. This is a special case of a general and important problem: the construction of relations between recursively defined domains. In the rest of the paper we present a general method for this construction which we hope will be applicable to a variety of problems in language definition. We begin by summarizing the construction of recursively defined domains themselves. The basic construction is due to Scott,[8] but our exposition follows that of Reference 9.

Definition We write $D \leftrightarrow D'$ to denote the domain $(D \to D') \times (D' \to D)$. The elements of $D \leftrightarrow D'$ are called function pairs from D to D'. When $D = D'$, the function pair $\mathcal{I}_D \equiv \langle I_D, I_D \rangle$ is called the identity element of $D \leftrightarrow D$. When $\langle\phi, \psi\rangle \in D \leftrightarrow D'$, the function pair $\langle\phi, \psi\rangle^{\dagger} \equiv \langle\psi, \phi\rangle \in D' \leftrightarrow D$ is called the reflection of $\langle\phi, \psi\rangle$. When $\langle\phi, \psi\rangle \in D \leftrightarrow D'$ and $\langle\phi', \psi'\rangle \in D' \leftrightarrow D''$, the function pair $\langle\phi', \psi'\rangle \cdot \langle\phi, \psi\rangle \equiv \langle\phi' \cdot \phi, \psi \cdot \psi'\rangle \in D \leftrightarrow D''$ is called the composition of $\langle\phi, \psi\rangle$ with $\langle\phi', \psi'\rangle$.

When $p = \langle\phi, \psi\rangle$ is a function pair, we write $[p]_\phi$ to denote ϕ, and $[p]_\psi$ to denote ψ. A function pair $\langle\phi, \psi\rangle \in D \leftrightarrow D'$ is said to be a

$$\begin{bmatrix} \text{retraction} \\ \text{projection} \\ \text{isomorphism} \end{bmatrix} \text{ pair iff } \begin{bmatrix} \psi \cdot \phi = I_D \\ \psi \cdot \phi = I_D \text{ and } \phi \cdot \psi \sqsubseteq I_{D'} \\ \psi \cdot \phi = I_D \text{ and } \phi \cdot \psi = I_{D'} \end{bmatrix} .$$

Corollary 1 For $p \in D \leftrightarrow D'$, $q \in D' \leftrightarrow D''$, and $r \in D'' \leftrightarrow D'''$:

(1) The expressions $p^\dagger$ and $q \cdot p$ are continuous in p and q.

(2) $r \cdot (q \cdot p) = (r \cdot q) \cdot p$.

(3) $p \cdot I_D = I_{D'} \cdot p = p$.

(4) $(p^\dagger)^\dagger = p$ and $(I_D)^\dagger = I_D$.

(5) $(q \cdot p)^\dagger = p^\dagger \cdot q^\dagger$.

(6) $\bot^\dagger = \bot$, and $\bot \cdot \bot = \bot$

(7)
$$p \text{ is a } \begin{bmatrix} \text{retraction} \\ \text{projection} \\ \text{isomorphism} \end{bmatrix} \text{ pair iff } \begin{bmatrix} p^\dagger \cdot p = I_D \\ p^\dagger \cdot p = I_D \text{ and } p \cdot p^\dagger \sqsubseteq I_{D'} \\ p^\dagger \cdot p = I_D \text{ and } p \cdot p^\dagger = I_{D'} \end{bmatrix} .$$

It is evident that there is a category $\mathcal{D}$ in which the objects are domains and the morphisms from D to D' are the function pairs between D and D', with identity and composition as defined above. Reflection is a contravariant functor from $\mathcal{D}$ to $\mathcal{D}$. Subcategories can be formed by restricting the function pairs to be retraction, projection, or isomorphism pairs.

Definition A domain functor is a function T which maps domains into domains and function pairs into function pairs and satisfies, for $p \in D \leftrightarrow D'$ and $q \in D' \leftrightarrow D''$:

(1) $T(p) \in T(D) \leftrightarrow T(D')$.

(2) $T(p)$ is continuous in p.

(3) $T(I_D) = I_{T(D)}$.

(4) $T(q) \cdot T(p) = T(q \cdot p)$.

(5) $(T(p))^\dagger = T(p^\dagger)$.

In category-theoretic terms, a domain functor is a functor from $\mathcal{D}$ to $\mathcal{D}$ which acts continuously on morphisms and commutes with reflection. The generalization to domain functors of several arguments is straightforward.

The following corollary provides a variety of means for constructing domain functors:

Corollary 2

(1) For any domain D_0, there is a n-ary domain functor T such that

$$T(D_1, \ldots, D_n) = D_0$$
$$T(p_1, \ldots, p_n) = I_{D_0}$$

(2) For any $1 \leq i \leq n$, there is an n-ary domain functor T such that

$$T(D_1, \ldots, D_n) = D_i$$
$$T(p_1, \ldots, p_n) = p_i$$

(3) If T_0 is an m-ary domain functor, and $T_1, \ldots, T_m$ are n-ary domain functors, there is an n-ary domain functor T such that

$$T(D_1, \ldots, D_n) = T_0(T_1(D_1, \ldots, D_n), \ldots, T_m(D_1, \ldots, D_n))$$
$$T(p_1, \ldots, p_n) = T_0(T_1(p_1, \ldots, p_n), \ldots, T_m(p_1, \ldots, p_n))$$

(4) There is a binary domain functor $\rightarrow$ such that $D_1 \rightarrow D_2$ is the domain of continuous functions from D_1 to D_2, with the partial ordering $f \sqsubseteq g$ iff $(\forall x)\ f(x) \sqsubseteq g(x)$, and for $\langle\phi_1, \psi_1\rangle \in D_1 \leftrightarrow D_1'$ and $\langle\phi_2, \psi_2\rangle \in D_2 \leftrightarrow D_2'$,

$$\langle\phi, \psi_1\rangle \rightarrow \langle\phi_2, \psi_2\rangle =$$
$$\langle\lambda f \in D_1 \rightarrow D_2.\ \phi_2\cdot f\cdot\psi_1,\quad \lambda f' \in D_1' \rightarrow D_2'.\ \psi_2\cdot f'\cdot\phi_1\rangle$$

(5) There is a binary domain functor + such that $D_1 + D_2$ is the separated sum of D_1 and D_2, and $\langle\phi_1, \psi_1\rangle + \langle\phi_2, \psi_2\rangle = \langle\phi_1+\phi_2, \psi_1+\psi_2\rangle$, where

$$(f_1+f_2)(\perp) = \perp$$
$$(f_1+f_2)(\top) = \top$$
$$(f_1+f_2)(\langle 1, x_1\rangle) = \langle 1, f_1(x_1)\rangle$$
$$(f_1+f_2)(\langle 2, x_2\rangle) = \langle 2, f_2(x_2)\rangle \quad .$$

According to this corollary, a domain functor can be defined by any expression built up from variables and constant domains by means of the binary operators $\rightarrow$ and +. For example,

$$T(x) = P + (x \rightarrow x)$$
$$T'(x) = ((P + (x \rightarrow x)) \rightarrow 0) \rightarrow 0$$

It is understood that when such an expression is evaluated to obtain a function pair, any constant domain will stand for the corresponding identity function pair.

The domains D and D' introduced earlier are solutions of the isomorphisms $D \simeq T(D)$ and $D' \simeq T'(D')$, where T and T' are the domain functors defined above. We consider the construction of solutions for arbitrary isomorphisms of this form.

Construction of Recursively Defined Domains

Given a domain functor T, we wish to construct a domain D_∞ which is isomorphic to $T(D_\infty)$. Let D_0 be the one-element lattice $\{\cdot\}$, and let $p_0 \in D_0 \leftrightarrow T(D_0)$ be the projection pair

$$p_0 = \langle\lambda x \in D_0.\ \perp_{T(D_0)},\ \lambda x \in T(D_0).\ \perp_{D_0}\rangle$$

For $n \geq 0$, let $D_{n+1} = T(D_n)$ and $p_{n+1} = T(p_n)$. Then the D_n's and p_n's form a sequence of projections:

$$D_0 \overset{p_0}{\leftrightarrow} D_1 \overset{p_1}{\leftrightarrow} D_2 \overset{p_2}{\leftrightarrow} \quad \cdots$$

Let $t_{mn} \in D_m \leftrightarrow D_n$ be defined by

$$\text{If} \quad \begin{bmatrix} m < n \\ m = n \\ m > n \end{bmatrix} \quad \text{then } t_{mn} = \quad \begin{bmatrix} p_{n-1}\cdot \ \cdots \ \cdot p_m \\ I_{D_m} \\ t_{nm}^{\dagger} \end{bmatrix}$$

It can be shown that the t_{mn}'s satisfy:[9]

(1) $t_{mn} = \perp$ when $m = 0$ or $n = 0$.

(2) $t_{m+1,n+1} = T(t_{mn})$

(3) t_{mn} is a projection pair when $m \leq n$.

(4) $t_{mn}\cdot t_{km} \sqsubseteq t_{kn}$

(5) $t_{mn} \cdot t_{km} = t_{kn}$ when $m \geq k$ or $m \geq n$.

(6) $t_{kn} = \bigsqcup_{m=o}^{\infty} t_{mn} \cdot t_{km}$, which is the limit of a directed sequence.

Next, we define D_∞ to be the inverse limit of the D_n's, i.e., the domain

$$D_\infty = \{<x_0, x_1, x_2, \ldots > \mid x_n \in D_n \text{ and } x_n = [p_n]_\psi(x_{n+1}) \}$$

with the partial ordering $x \sqsubseteq y$ iff $[x]_n \sqsubseteq [y]_n$ for all n. Then let $t_{n\infty} \in D_n \leftrightarrow D_\infty$ be the function pair

$$t_{n\infty} = <\lambda x_n \in D_n.\ <[t_{n0}]_\phi(x_n), [t_{n1}]_\phi(x_n), [t_{n2}]_\phi(x_n), \ldots >,$$
$$\lambda x \in D_\infty.\ [x]_n>$$

and let $t_{\infty n} = t_{n\infty}^{\dagger}$ and $t_{\infty\infty} = I_{D_\infty}$. Then it can be shown that the above properties of the t_{mn}'s continue to hold when ∞ is permitted as a subscript.

Finally, let i be the limit of the directed sequence

$$i = \bigsqcup_{n=0}^{\infty} T(t_{n\infty}) \cdot t_{\infty n+1} \in D_\infty \leftrightarrow T(D_\infty) .$$

Then i is an isomorphism, so that $D_\infty \simeq T(D_\infty)$. It can also be shown that D_∞ is minimal, in the sense that whenever $D' \simeq T(D')$ there is a projection pair from D_∞ to D'.

Directed Complete Relations

In order to construct relations between recursively defined domains, we must impose a rather weak kind of continuity condition:

Definition A relation θ between domains D and D' is said to be directed complete iff $\theta: x \mapsto x'$ whenever x and x' are the least upper bounds of two directed sequences $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots$ and $x_0' \sqsubseteq x_1' \sqsubseteq x_2' \sqsubseteq \ldots$ such that $\theta: x_n \mapsto x_n'$ for all n.

It is easily seen that universally true and false relations, equality, and the partial ordering $\sqsubseteq$ are all directed complete. On the other hand, the topological relation $\prec$ is not directed complete. Moreover,

Corollary 3

(1) A continuous function is a directed complete relation.

(2) A directed complete relation which is a monotonic function is a continuous function.

(3) The converse of a directed complete relation is directed complete.

(4) If θ is a directed complete relation and f is a continuous function then the relational composition $\theta \cdot f$ is directed complete.

(5) The intersection of a set of directed complete relations is directed complete.

(6) The union of a finite set of directed complete relations is directed complete.

We leave the proof to the reader, except the heart of (6): Given directed complete relations θ and η, and a pair of directed sequences such that $\theta \cup \eta: x_i \mapsto x'_i$, we have $\theta: x_i \mapsto x'_i$ or $\eta: x_i \mapsto x'_i$ for each i. But at least one of these relations must hold for an infinite number of i's, and therefore for a pair of directed subsequences with the same limits as the original sequences.

Unfortunately, directed completeness is not preserved under relational composition, so that directed complete relations do not form a subcategory of the category of relations among domains.

<u>Diagrams</u>

<u>Definition</u> A <u>diagram</u> is a collection of four domains, two function pairs, and two directed complete relations with the following form,

$$\begin{array}{ccc} D & \overset{p = \langle \phi, \psi \rangle}{\leftrightarrow} & \bar{D} \\ \theta \downarrow & & \downarrow \bar{\theta} \\ D' & \underset{\leftrightarrow}{p' = \langle \phi', \psi' \rangle} & \bar{D}' \end{array},$$

whose components satisfy the properties

For all $x \in D$ and $x' \in D'$, $\theta: x \mapsto x'$ implies $\bar{\theta}: \phi(x) \mapsto \phi'(x')$

For all $\bar{x} \in \bar{D}$ and $\bar{x}' \in \bar{D}'$, $\bar{\theta}: \bar{x} \mapsto \bar{x}'$ implies $\theta: \psi(\bar{x}) \mapsto \psi'(\bar{x}')$

These properties can be stated more succinctly using relational composition,

$$\theta \subseteq \phi'^{-1} \cdot \bar{\theta} \cdot \phi \qquad \text{where } f^{-1} \text{ denotes the converse of } f,$$
$$\bar{\theta} \subseteq \psi'^{-1} \cdot \theta \cdot \psi$$

and can be recast into a variety of forms such as

$$\phi' \cdot \theta \subseteq \bar{\theta} \cdot \phi$$
$$\psi' \cdot \bar{\theta} \subseteq \theta \cdot \psi$$

by using the fact that for any function $f \in D \to \bar{D}$, $I_D \subseteq f^{-1} \cdot f$ and $f \cdot f^{-1} \subseteq I_{\bar{D}}$.

It is easy to see that the <u>vertical</u> <u>and</u> <u>horizontal</u> <u>reflections</u>,

$$\begin{array}{ccc} D' & \overset{p'}{\leftrightarrow} & \bar{D}' \\ \theta^{-1} \downarrow & & \downarrow \bar{\theta}^{-1} \\ D & \overset{p}{\leftrightarrow} & \bar{D} \end{array} \qquad \begin{array}{ccc} \bar{D} & \overset{p^{\dagger}}{\leftrightarrow} & D \\ \bar{\theta} \downarrow & & \downarrow \theta \\ \bar{D}' & \overset{p'^{\dagger}}{\leftrightarrow} & D' \end{array},$$

of a diagram are diagrams, that two diagrams of the form

$$\begin{array}{ccc} D & \overset{p}{\leftrightarrow} & \bar{D} \\ \theta \downarrow & & \downarrow \bar{\theta} \\ D' & \overset{p'}{\leftrightarrow} & \bar{D}' \end{array} \qquad \begin{array}{ccc} \bar{D} & \overset{q}{\leftrightarrow} & \bar{\bar{D}} \\ \bar{\theta} \downarrow & & \downarrow \bar{\bar{\theta}} \\ \bar{D}' & \overset{q'}{\leftrightarrow} & \bar{\bar{D}}' \end{array}$$

have a <u>horizontal</u> <u>composition</u>

$$\begin{array}{ccc} D & \overset{q \cdot p}{\leftrightarrow} & \bar{\bar{D}} \\ \theta \downarrow & & \downarrow \bar{\bar{\theta}} \\ D' & \overset{q' \cdot p'}{\leftrightarrow} & \bar{\bar{D}}' \end{array}$$

which is a diagram, and that diagrams of the form

$$\begin{array}{ccc} D & \overset{I_D}{\leftrightarrow} & D \\ \theta \big\downarrow & & \big\downarrow \theta \\ D' & \overset{I_{D'}}{\leftrightarrow} & D' \end{array}$$

are identity diagrams for horizontal composition. (Diagrams cannot be composed vertically since directed completeness is not preserved by relational composition.)

In brief, diagrams under horizontal composition from a category $\mathcal{R}$ in which the objects are triples of the form $\langle D, D', \theta\rangle$ where $\theta \in D \nrightarrow D'$, and the morphisms from $\langle D, D', \theta\rangle$ to $\langle \bar{D}, \bar{D}', \bar{\theta}\rangle$ are the pairs $\langle p, p'\rangle \in (D\leftrightarrow\bar{D})\times(D'\leftrightarrow\bar{D}')$ of function pairs satisfying the above-stated properties. Vertical and horizontal reflection are covariant and contravariant functors from $\mathcal{R}$ to $\mathcal{R}$. There are also two functors, top and bot, from $\mathcal{R}$ to $\mathcal{D}$, defined by

$$\text{top}(\langle D, D', \theta\rangle) = D \qquad \text{top}(\langle p, p'\rangle) = p$$
$$\text{bot}(\langle D, D', \theta\rangle) = D' \qquad \text{bot}(\langle p, p'\rangle) = p' \quad .$$

Directed completeness and the nature of isomorphisms have the following consequences for diagrams:

Corollary 4 If p and p' are the least upper bounds of two directed sequences $p_0 \sqsubseteq p_1 \sqsubseteq p_2 \sqsubseteq \ldots$ and $p'_0 \sqsubseteq p'_1 \sqsubseteq p'_2 \sqsubseteq \ldots$ such that

$$\begin{array}{ccc} D & \overset{p_n}{\leftrightarrow} & \bar{D} \\ \theta \big\downarrow & & \big\downarrow \bar{\theta} \\ D' & \overset{p'_n}{\leftrightarrow} & \bar{D}' \end{array}$$

is a diagram for all n, then

$$\begin{array}{ccc} D & \overset{p}{\leftrightarrow} & D \\ \theta \big\downarrow & & \big\downarrow \bar{\theta} \\ D' & \overset{p'}{\leftrightarrow} & \bar{D}' \end{array}$$

is a diagram.

If

$$\begin{array}{ccc} D & \overset{i}{\leftrightarrow} & \bar{D} \\ \theta \big\downarrow & & \big\downarrow \bar{\theta} \\ D' & \overset{i'}{\leftrightarrow} & \bar{D}' \end{array}$$

is a diagram in which $i = \langle\phi, \psi\rangle$ and $i' = \langle\phi', \psi'\rangle$ are isomorphisms, then $\theta = \psi'\cdot\bar{\theta}\cdot\phi$ and $\bar{\theta} = \phi'\cdot\theta\cdot\psi$.

Relational Functors

Definition Given domain functors T and T', a function B on directed complete relations is called a relational functor between T and T' iff (1) If $\theta \in D \nrightarrow D'$, then $B(\theta) \in T(D) \nrightarrow T'(D')$, and (2) If

$$\begin{array}{ccc} D & \overset{p}{\leftrightarrow} & \bar{D} \\ \theta \big\downarrow & & \big\downarrow \bar{\theta} \\ D' & \overset{p'}{\leftrightarrow} & \bar{D}' \end{array}$$

is a diagram, then

$$\begin{array}{ccc} T(D) & \overset{T(p)}{\leftrightarrow} & T(\bar{D}) \\ B(\theta)\downarrow\!\!\!\!\times & & \downarrow\!\!\!\!\times\, B(\bar{\theta}) \\ T'(D') & \overset{T'(p')}{\leftrightarrow} & T'(\bar{D}') \end{array}$$

is a diagram. We write $T \Rightarrow\!\!\!\!\times T'$ for the set of relational functors between T and T'.

In category-theoretic terms, a relational functor between T and T' is essentially a functor from $\mathcal{R}$ to $\mathcal{R}$ which satisfies $top \cdot B = T \cdot top$ and $bot \cdot B = T' \cdot bot$. The generalization to relational functors of several arguments is straightforward.

The following Corollary provides a variety of means for constructing relational functors:

<u>Corollary 5</u>

(1) For any $\theta_0 \in D_0 \to\!\!\!\!\times D_0'$, there is an n-ary $B \in T \Rightarrow\!\!\!\!\times T'$ such that

$$T(x_1, \ldots, x_n) = D_0$$
$$T'(x_1, \ldots, x_n) = D_0'$$
$$B(\theta_1, \ldots, \theta_n) = \theta_0$$

(2) For any $1 \leq i \leq n$, there is an n-ary $B \in T \Rightarrow\!\!\!\!\times T$ such that

$$T(x_1, \ldots, x_n) = x_i$$
$$B(\theta_1, \ldots, \theta_n) = \theta_i$$

(3) If $B_0 \in T_0 \Rightarrow\!\!\!\!\times T_0'$ is an m-ary relational functor, and $B_1 \in T_1 \Rightarrow\!\!\!\!\times T_1'$, $\ldots$, $B_m \in T_m \Rightarrow\!\!\!\!\times T_m'$ are n-ary relational functors, then there is an n-ary $B \in T \Rightarrow\!\!\!\!\times T'$ such that

$$T(x_1, \ldots, x_n) = T_0(T_1(x_1, \ldots, x_n), \ldots, T_m(x_1, \ldots, x_n))$$
$$T'(x_1, \ldots, x_n) = T_0'(T_1'(x_1, \ldots, x_n), \ldots, T_m'(x_1, \ldots, x_n))$$
$$B(\theta_1, \ldots, \theta_n) = B_0(B_1(\theta_1, \ldots, \theta_n), \ldots, B_m(\theta_1, \ldots, \theta_n))$$

(4) If $B_1, B_2 \in T \Rightarrow\!\!\!\!\times T'$ are n-ary relational functors, then there is an n-ary $B \in T \Rightarrow\!\!\!\!\times T'$ such that

$$B(\theta_1, \ldots, \theta_n) = B_1(\theta_1, \ldots, \theta_n) \cup B_2(\theta_1, \ldots, \theta_n)$$

(5) If $B_1 \in T' \Rightarrow\!\!\!\!\times T$ is an n-ary relational functor, then there is an n-ary $B \in T \Rightarrow\!\!\!\!\times T'$ such that

$$B(\theta_1, \ldots, \theta_n) = (B_1(\theta_1^{-1}, \ldots, \theta_n^{-1}))^{-1}$$

(6) Let $T_\cdot$ be the domain functor such that $T_\cdot(x_1, \ldots, x_n) = \{\cdot\}$. If $B_1 \in T_1 \Rightarrow\!\!\!\!\times T'$, $B_2 \in T_2 \Rightarrow\!\!\!\!\times T'$, $B_\perp, B_\top \in T_\cdot \Rightarrow\!\!\!\!\times T'$ are n-ary relational functors then there is an n-ary $B \in T \Rightarrow\!\!\!\!\times T'$ such that

$$T(x_1, \ldots, x_n) = T_1(x_1, \ldots, x_n) + T_2(x_1, \ldots, x_n)$$
$$B(\theta_1, \ldots, \theta_n) =$$
$$+(B_1(\theta_1, \ldots, \theta_n), B_2(\theta_1, \ldots, \theta_n), B_\perp(\theta_1, \ldots, \theta_n), B_\top(\theta_1, \ldots, \theta_n)),$$

where $+(\theta_1, \theta_2, \theta_\perp, \theta_\top)$: $x \mapsto x'$ iff

$x = \perp$ and $\theta_\perp: \cdot \mapsto x'$
or $x = \top$ and $\theta_\top: \cdot \mapsto x'$
or $x = \langle 1, y \rangle$ and $\theta_1: y \mapsto x'$
or $x = \langle 2, y \rangle$ and $\theta_2: y \mapsto x'$

(7) There is a binary relational functor $\rightarrow \in T \Rightarrow\!\!\!\times T$

$T(x_1, x_2) = x_1 \rightarrow x_2$
For $\theta_1 \in D_1 \rightarrow\!\!\!\times D_1'$ and $\theta_2 \in D_2 \rightarrow\!\!\!\times D_2'$, $\theta_1 \rightarrow \theta_2$: $f \mapsto f'$ iff,
for all $x \in D_1$ and $x' \in D_1'$, θ_1: $x \mapsto x'$ implies θ_2: $f(x) \mapsto f'(x')$

(8) For any domain O containing two or more elements, there is a unary relational functor $K \in T \Rightarrow\!\!\!\times T'$ such that

$T(x) = x$
$T'(x) = (x \rightarrow O) \rightarrow O$
For $\theta \in D \rightarrow\!\!\!\times D'$, $K(\theta)$: $x \mapsto x''$ iff there is a $x' \in D'$
such that θ: $x \mapsto x'$ and $x'' = \lambda c \in D' \rightarrow O.\ c(x')$

We leave the proof to the reader except for a couple of tricky points about directed completeness. In (6), the directed completeness of $+(\theta_1, \theta_2, \theta_\perp, \theta_\top)$ depends upon the fact that any directed sequence in a separated lattice sum must possess a tail with the same limit which consists entirely of $\perp$, or consists entirely of $\top$, or consists entirely of elements from one of the component lattices. In (8), one must show that if the elements $\lambda c \in D' \rightarrow O.\ c(x_n')$ form a directed sequence in $(D' \rightarrow O) \rightarrow O$, then the x_n' must be a directed sequence in D'. This is equivalent to showing that $\lambda c.\ c(x') \sqsubseteq \lambda c.\ c(y')$ implies $x' \sqsubseteq y'$. For each open subset U' of D', let

$$c_{U'} = \lambda x' \in D'.\ \underline{\text{if}}\ x' \in U'\ \underline{\text{then}}\ \top_O\ \underline{\text{else}}\ \perp_O$$

If $\lambda c.\ c(x') \sqsubseteq \lambda c.\ c(y')$, then $c_{U'}(x') \sqsubseteq c_{U'}(y')$ for every U'. Then since O contains two or more elements, $\top_O$ and $\perp_O$ are distinct, so that every open set containing x' must contain y'. This implies $x' \sqsubseteq y'$.

With regard to the specific problem raised in the first part of this paper, Corollary 5 implies that there is a relational functor $B \in T \Rightarrow\!\!\!\times T'$ such that

$T(x) = P + (x \rightarrow x)$
$T'(x) = ((P + (x \rightarrow x)) \rightarrow O) \rightarrow O$
$B(\theta) = +(K(+(\lambda x \in P.\ x, \text{false}, \text{false}, \text{false})^{-1}),$
$K(+(\text{false}, (\theta \rightarrow \theta)^{-1}, \text{false}, \text{false})^{-1}),$
$\lambda y \in \{\cdot\}.\ \perp_{T'(D')} \cup \lambda y \in \{\cdot\}.\ \lambda c \in (P+(D' \rightarrow D')) \rightarrow O.\ \text{error},$
$\lambda y \in \{\cdot\}.\ \top_{T'(D')})$

where $\theta \in D \rightarrow\!\!\!\times D'$, and false denotes the universally false relation. The property of η given earlier is just that η: $x \mapsto x'$ iff $B(\eta)$: $\Phi(x) \mapsto \Phi'(x')$, where Φ and Φ' are the ϕ-components of the isomorphisms between D_∞ and $T(D_\infty)$, and D_∞' and $T'(D_\infty')$. We now consider the construction of such relations for a variety of relational functors.

<u>Relations between Recursively Defined Domains</u>

Let T and T' be domain functors, and let p_n, t_{mn}, D_n, $i = \langle \Phi, \Psi \rangle$ and p'_n, t'_{mn}, D'_n, $i = \langle \Phi', \Psi' \rangle$ be the entities obtaining by applying the previously described construction to T and T' respectively, so that $D_\infty \simeq T(D_\infty)$ and $D'_\infty \simeq T'(D'_\infty)$ with isomorphisms i and i'.

Suppose B is a relational functor between T and T', with the additional property that, for any $\theta \in D \nrightarrow D'$, $B(\theta)\colon \bot_{T(D)} \mapsto \bot_{T'(D')}$. We wish to construct a unique directed complete relation $\theta_\infty \in D \nrightarrow D'$ such that $\theta_\infty\colon x \mapsto x'$ iff $B(\theta_\infty)\colon \Phi(x) \mapsto \Phi'(x')$.

We begin by defining θ_0 to be the universally true relation between the one-point domains D_0 and D'_0 and

$$\theta_{n+1} = B(\theta_n) \ .$$

Then consider the sequence

$$\begin{array}{ccccccc} D_0 & \overset{p_0}{\leftrightarrow} & D_1 & \overset{p_1}{\leftrightarrow} & D_2 & \overset{p_2}{\leftrightarrow} & \cdots \\ \theta_0 \downarrow & & \theta_1 \downarrow & & \theta_2 \downarrow & & \\ D'_0 & \overset{p'_0}{\leftrightarrow} & D'_1 & \overset{p'_1}{\leftrightarrow} & D'_2 & \overset{p'_2}{\leftrightarrow} & \cdots \end{array}$$

The additional restriction that $B(\theta)\colon \bot \mapsto \bot$ insures that the first rectangle is a diagram, and the fact that B is a relational functor insures that each successive rectangle is a diagram. Then by horizontal composition and reflection, for all m and n

$$\begin{array}{ccc} D_m & \overset{t_{mn}}{\leftrightarrow} & D_n \\ \theta_m \downarrow & & \downarrow \theta_n \\ D'_m & \overset{t'_{mn}}{\leftrightarrow} & D'_n \end{array}$$

is a diagram.

Now let θ_∞ be the relation between D_∞ and D'_∞ such that $\theta_\infty\colon x \mapsto x'$ iff, for all n, $\theta_n\colon [x]_n \mapsto [x']_n$. This is equivalent to

$$\theta_\infty = \bigcap_{n=0}^{\infty} [t'_{n\infty}]_\psi^{-1} \cdot \theta_n \cdot [t_{n\infty}]_\psi$$

which establishes that θ_∞ is directed complete.

Obviously, $\theta_\infty\colon x \mapsto x'$ implies $\theta_n\colon [t_{n\infty}]_\psi(x) \mapsto [t'_{n\infty}]_\psi(x')$ for all n. On the other hand, from the diagrams for the t_{mn}, $\theta_n\colon x_n \mapsto x'_n$ implies

$$\theta_m\colon [t_{nm}]_\phi(x_n) \mapsto [t'_{nm}]_\phi(x'_n) \text{ for all m}$$

which is equivalent to

$$\theta_m\colon [[t_{n\infty}]_\phi(x_n)]_m \mapsto [[t'_{n\infty}]_\phi(x'_n)]_m \text{ for all m}$$

which is equivalent to

$$\theta_\infty\colon [t_{n\infty}]_\phi(x_n) \mapsto [t'_{n\infty}]_\phi(x'_n) \ .$$

Thus

$$\begin{array}{ccc} D_n & \overset{t_{n\infty}}{\leftrightarrow} & D_\infty \\ \theta_n \downarrow & & \downarrow \theta_\infty \\ D'_n & \overset{t'_{n\infty}}{\leftrightarrow} & D'_\infty \end{array}$$

is a diagram for all n.

By applying B to this diagram, we get

$$\begin{array}{ccc} D_{n+1} & \overset{T(t_{n\infty})}{\leftrightarrow} & T(D_\infty) \\ \theta_{n+1} \big\downarrow\!\!\!\!\times & & \big\downarrow\!\!\!\!\times\, B(\theta_\infty) \\ D'_{n+1} & \overset{T'(t'_{n\infty})}{\leftrightarrow} & T'(D'_\infty) \end{array}$$

Then, by replacing n by n+1 in the diagram for $t_{n\infty}$, and reflecting and composing it with the diagram for $T(t_{n\infty})$, we get

$$\begin{array}{ccc} D_\infty & \overset{T(t_{n\infty})\cdot t_{\infty n+1}}{\leftrightarrow} & T(D_\infty) \\ \theta_\infty \big\downarrow\!\!\!\!\times & & \big\downarrow\!\!\!\!\times\, B(\theta_\infty) \\ D'_\infty & \overset{T'(t'_{n\infty})\cdot t'_{\infty n+1}}{\leftrightarrow} & T'(D'_\infty) \end{array}$$

By Corollary 4, we get

$$\begin{array}{ccc} D_\infty & \overset{i}{\leftrightarrow} & T(D_\infty) \\ \theta_\infty \big\downarrow\!\!\!\!\times & & \big\downarrow\!\!\!\!\times\, B(\theta_\infty) \\ D'_\infty & \overset{i'}{\leftrightarrow} & T'(D'_\infty) \end{array}$$

and since $i = \langle \Phi, \Psi \rangle$ and $i' = \langle \Phi', \Psi' \rangle$ are isomorphisms, we have the desired result:

$$\theta_\infty : x \mapsto x' \text{ iff } B(\theta_\infty) : \Phi(x) \mapsto \Phi'(x') \ .$$

It remains to show that θ_∞ is unique. Let $\alpha \in D_\infty \rightarrow\!\!\!\!\times D'_\infty$ be any directed complete relation such that

$$\begin{array}{ccc} D_\infty & \overset{i}{\leftrightarrow} & T(D_\infty) \\ \alpha \big\downarrow\!\!\!\!\times & & \big\downarrow\!\!\!\!\times\, B(\alpha) \\ D'_\infty & \overset{i'}{\leftrightarrow} & T'(D'_\infty) \end{array}$$

is a diagram. The restrictions on B insure that $\alpha : \bot \mapsto \bot$, which implies that

$$\begin{array}{ccc} D_0 & \overset{t_{0\infty}}{\leftrightarrow} & D_\infty \\ \theta_0 \big\downarrow\!\!\!\!\times & & \big\downarrow\!\!\!\!\times\, \alpha \\ D'_0 & \overset{t'_{0\infty}}{\leftrightarrow} & D'_\infty \end{array}$$

since $t_{0\infty}$ maps the unique element of D_0 into $\bot_{D_\infty}$, and similarly for $t'_{0\infty}$.

Then by induction on n, we can show that

$$\begin{array}{ccc} D_\infty & \overset{t_{n\infty}\cdot t_{\infty n}}{\leftrightarrow} & D_\infty \\ \theta_\infty \big\downarrow\!\!\!\!\times & & \big\downarrow\!\!\!\!\times\, \alpha \\ D'_\infty & \overset{t'_{n\infty}\cdot t'_{\infty n}}{\leftrightarrow} & D'_\infty \end{array}$$

The n = 0 case follows from the composition of the two diagrams involving $t_{0\infty}$. By applying B to the diagram for the nth case, and composing with the appropriate diagrams involving i and i' on either side, we get

$$\begin{array}{ccccccc}
D_\infty & \overset{i}{\leftrightarrow} & T(D_\infty) & \overset{T(t_{n\infty}\cdot t_{\infty n})}{\leftrightarrow} & T(D_\infty) & \overset{i^\dagger}{\leftrightarrow} & D_\infty \\
\theta_\infty \downarrow & & B(\theta_\infty) \downarrow & & \downarrow B(\alpha) & & \downarrow \alpha \\
D'_\infty & \overset{i'}{\leftrightarrow} & T'(D'_\infty) & \overset{T'(t'_{n\infty}\cdot t'_{\infty n})}{\leftrightarrow} & T'(D'_\infty) & \overset{i'^\dagger}{\leftrightarrow} & D'_\infty
\end{array}$$

which reduces to the diagram for the n+1th case since $t_{n+1,\infty}\cdot t_{\infty,n+1} = i^\dagger\cdot T(t_{n\infty}\cdot t_{\infty n})\cdot i$, and similarly for the primed quantities.

Finally, since the quantities $t_{n\infty}\cdot t_{\infty n}$ form a directed sequence whose limit is I_{D_∞}, and similarly for the primed quantities, Corollary 4 gives $\alpha = \theta_\infty$.

ACKNOWLEDGEMENT

The author wishes to thank Professor Dana Scott for suggesting the notion of a directed complete relation and its relevance to the inverse limit construction. The use of such relations in proving theorems similar to Theorem 1 has been discovered independently by R. Milne of Oxford.

REFERENCES

1. Morris, L., The Next 700 Programming Language Descriptions. Unpublished.

2. Strachey, C. and Wadsworth, C. P., Continuations - A Mathematical Semantics for Handling Full Jumps. Tech. Monograph PRG-11, Programming Research Group Oxford University Computing Laboratory, January 1974.

3. Reynolds, J. C., "Definitional Interpreters for Higher-Order Programming Languages", Proc. 25th National ACM Conference, Boston, August 1972.

4. Tennent, R. D., Mathematical Semantics and Design of Programming Languages. Ph.D. Thesis, University of Toronto, October, 1973.

5. Landin, P. J., "A Correspondence Between ALGOL 60 and Church's Lambda-Notation". Comm ACM 8 (February-March 1965), 89-101 and 158-165.

6. Scott, D., "Lattice-Theoretic Models for Various Type-Free Calculi," Proc. Fourth International Congress for Logic, Methodology, and the Philosophy of Science, Bucharest (1972).

7. Naur, P., et al. Revised Report on the Algorithmic Language ALGOL 60. Comm ACM 6, 1 (January 1963), 1-17.

8. Scott, D., "Continuous Lattices," Proc. 1971 Dalhousie Conf., Springer Lecture Note Series, Springer-Verlag, Heidelberg. Also, Tech. Monograph PRG-7, Programming Research Group, Oxford University Computing Laboratory, August 1971.

9. Reynolds, J. C., "Notes on a Lattice-Theoretic Approach to the Theory of Computation", Systems and Information Science, Syracuse University, October 1972.