IBM Research

A CHURCH-ROSSER PROPERTY OF CLOSED APPLICATIVE LANGUAGES

Paul McJones

May 23, 1975

RJ 1589

Yorktown Heights, New York

San Jose, California

Zurich, Switzerland

Copies may be requested from:
IBM Thomas J. Watson Research Center
Post Office Box 218
Yorktown Heights, New York 10598

RJ 1589 (#23713)
May 23, 1975
Computer Science

A CHURCH-ROSSER PROPERTY OF CLOSED APPLICATIVE LANGUAGES

by

Paul McJones

IBM Research Laboratory
San Jose, California 95193

ABSTRACT: In "Programming Language Semantics and Closed Applicative Languages" [IBM Research Report RJ1245], John Backus defines the class of closed applicative languages (of which his Red is a member), and states some of their properties. One of the most interesting is a Church-Rosser property: if an expression has a meaning, then every terminating sequence of reductions on it yields that meaning. The purpose of this report is to show how to prove this and other properties.

The approach will be to first construct the meaning function (mapping expressions into values) for a closed applicative language as the least fixed point of a continuous functional, thus establishing the "mathematical" semantics of the language. Then the "operational" notion of reduction will be defined in terms of the meaning function, and some of its properties verified using structural and computational induction.

1. Some machinery from the fixed point theory is necessary. [See for example: Manna, Z., Ness, S., and Vuilleman, J. Inductive methods for proving properties of programs. CACM 16, 8 (August 1973), 491-502.] A partial ordering $\leq_S$ on a set S is a relation with the following three properties:

$$\left.\begin{array}{ll} \text{reflexivity:} & a\leq_S a \\ \text{transitivity:} & a\leq_S b,\ b\leq_S c \text{ implies } a\leq_S c \\ \text{antisymmetry:} & a\leq_S b,\ b\leq_S a \text{ implies } a=b \end{array}\right\} \quad \text{for all } a,b,c,\in S$$

A least upper bound for a subset X of a partially ordered set [poset] S is an element $a\in S$ such that:

i) $x\leq_S a$ for all $x\in X$

and ii) $a\leq_S b$ for all $b\in S$ such that for all $x\in X$, $x\leq_S b$.

If X has a least upper bound, we denote it by $\text{lub}_S X$. A poset S is complete if it has a least element [usually denoted $\perp_S$ or, somewhat ambiguously, just $\perp$] and if $\text{lub}_S L$ exists for every nonempty totally ordered $L\subseteq S$.

If $f:S_1\to S_2$, for complete posets S_1,S_2, then f is continuous if $\text{lub}_{S_2}\{f\ell:\ell\in L\}$ exists and equals $f(\text{lub}_{S_1}L)$ for every nonempty totally ordered $L\subseteq S_1$. The set of all continuous functions from S_1 to S_2, for given complete posets S_1 and S_2, can itself be given a complete poset structure. We let $f\leq g$ iff $f(a)\leq_{S_2}g(a)$, for all $a\in S_1$. The least element is Ω, defined by $\Omega(a)=\perp_{S_2}$ for all $a\in S_1$.

A set S is a trivial complete poset if there exists an element $\perp_s\in S$ and a relation $\leq_S$ satisfying $a\leq_S a'$ iff $a=\perp_s$ or $a=a'$, for all $a,a'\in S$. Suppose $f:S_1^n\to S_2$ for some trivial complete poset S_1, any complete poset

S_2, and some $n \geq 0$. Then a condition sufficient to ensure f continuous is that f be <u>⊥-preserving</u>, that is, $f(s_1, s_2, \cdots, s_n) = \perp_{S_2}$ whenever any $s_i = \perp_{S_1}$ ($s_i \in S_1$, $1 \leq i \leq n$).

2. Let E be a fixed set of <u>expressions</u>. Both "programs" and "meanings" of our closed applicative language [CAL] will lie in this set. We assume E is a trivial complete poset; intuitively $\perp_E$ represents the result of a nonterminating evaluation.

3. The pair <A,K> is a <u>constructor syntax</u> for E if the following hold [the idea is to give an "abstract syntax", postulating only those properties relevant to our subsequent discussion without specifying a particular representation]:

CS0) $\perp \in E$

CS1) $A \subseteq E$ [$a \in A$ is called an <u>atom</u>]

CS2) For each $k \in K$, there is an integer $n \geq 0$ such that k is a ⊥-preserving function from E^n into E. [$k \in K$ is called an n-place <u>constructor</u>]

CS3) For every $e \in E$, <u>exactly one</u> of the following holds:

- i) $e = \perp$ [in which case e has no <u>components</u>]
- ii) $e \in A$ [in which case e has no <u>components</u>]
- iii) there is a unique $k \in K$ [hence, by CS2, a unique $n \geq 0$] and unique $e_1, \cdots, e_n \in E - \{\perp\}$ such that $e = k(e_1, \cdots, e_n)$ [in which case the <u>components</u> of e are $e_1, \cdots, e_n$]

CS4) No expression has an infinite chain of components [so repeated applications of CS3 eventually results in atoms and 0-place constructors].

CS4 assures that we can make proofs by structural induction: if a property P holds for every expression e whenever it holds for all components of e, then P must hold for all expressions in E. [Since $\perp$ and atoms have no components, P must hold for them.]

4. A specific closed applicative language is determined by two things:

   i) a specific constructor syntax <A,K> with a distinguished 2-place constructor $ap \varepsilon K$ [called the application constructor]

   ii) a representation function ρ which is a function from E, to functions from E to E. To assure continuity, a legitimate ρ must satisfy:

   $\rho 1$) $(\rho\perp_E)e=\perp_E$ for all $e\varepsilon E$ [i.e., $\rho\perp_E=\Omega$, the everywhere-$\perp$ function]

   $\rho 2$) $(\rho e)\perp_E=\perp_E$ for all $e\varepsilon E$ [i.e., ρe is always $\perp$-preserving].

5. The meaning function for CALs is best given recursively, defining the meaning of a composite expression in terms of the meanings of its components. The fixed point theory provides a sound basis for such definitions: every continuous function f from a complete poset S into itself has a unique least fixed point $p=lub_S\{\perp_S, f\perp_S, f(f\perp_S), \cdots\}$ [that is, $fp=p$, and if $fq=q$, then $p\leq_S q$, all $q\varepsilon S$]. To see why this is useful, consider the following definition of a functional τ mapping the set of continuous E to E functions into itself. [Notational convention: function applications associate to the right, so $fghx=f(g(hx))$.] For every continuous E to E function f and for every $e\varepsilon E$, let

$$(\tau f)e \overset{\text{def}}{=} \begin{cases} \perp_E \text{ if } e=\perp_E \\ e \text{ if } e \in A \\ k(fe_1,\cdots,fe_n) \text{ if } e=k(e_1,\cdots,e_n) \text{ and } k \neq ap \\ f(\rho fe_1)fe_2 \text{ if } e=ap(e_1,e_2) \end{cases}$$

It is tedious but not impossible to show that τ is indeed continuous. [As discussed in Manna et al., this follows from its composition of known continuous functions and the function parameter f.] Now we define $\mu=f_\tau$=the unique least fixed point of τ to be the meaning function of our CAL, and claim that it has the properties stated in Backus' report. To substantiate this claim, we will use the principle of <u>computational induction</u>: if a property P is true of Ω [the everywhere-$\perp$ function], and if P being true of f implies that P is true of τf, then P must be true of f_τ, the least fixed point of τ. [Technically, P must be an <u>admissable predicate</u>, as discussed by Manna et al.]

6. The subset C of E constructed without ap we call <u>constants</u> [as justified by proposition 3 below]. We can define C recursively as follows:

C1) $A \subseteq C$

C2) If $k \in K$, $k \neq ap$, and $e_1,\cdots,e_n \in C$, then $k(e_1,\cdots,e_n) \in C$

C3) $e \in C$ only by virtue of C1,C2 above.

7. Based on the definition of μ, we can define a relation R of <u>reducibility</u>. An expression e is [directly] reducible to e', eRe', iff:

R1) $e=ap(c_1,c_2)$ and $e'=(\rho c_1)c_2$, for some $c_1,c_2 \in C$

or

R2) $e=k(e_1,\cdots,e_i,\cdots,e_n)$ and $e'=k(e_1,\cdots,e_i',\cdots,e_n)$ and $e_i R e_i'$, for some $k\epsilon K$ and some i, $1\leq i\leq n$.

Intuitively eRe' means e' results from e by "performing some innermost application".

8. We will need an additional function δ mapping expressions into the trivial complete poset $N=\{\perp_N,0,1,2,\cdots\}$ [with the obvious $\leq_N$]. Intuitively, δe gives the number of reductions needed to evaluate e. Using the fixed point theory, we take δ to be f_σ, the unique least fixed point of the [reputedly] continuous functional σ defined as follows: For every continuous E to E function f and for every $e\epsilon E$, let

$$(\sigma f)e \overset{\text{def}}{=} \begin{cases} \perp_N \text{ if } e=\perp_E \\ 0 \text{ if } e\epsilon A \\ f(e_1)+\cdots+f(e_n) \text{ if } e=k(e_1,\cdots,e_n),\ k\neq ap \\ 1+f(e_1)+f(e_2)+f(\rho\mu e_1)\mu e_2 \text{ if } e=ap(e_1,e_2) \end{cases}$$

The propositions below give various properties of μ, R, and δ, culminating in the "Church-Rosser property" for CALs.

Prop. 1. For all $e\varepsilon E$, $\mu e\varepsilon C\cup\{\bot\}$.

Proof. We use computational induction.

<u>Basis</u>: $\Omega e=\bot$ for all $e\varepsilon E$, by definition of Ω.

<u>Induction</u>: Assume that $fe\varepsilon C\cup\{\bot\}$ for all $e\varepsilon E$. We show that $(\tau f)e\varepsilon C\cup\{\bot\}$ for all $e\varepsilon E$. By definition CS3, there are three distinct cases.

<u>Case 1</u>: $e=\bot$. Then $(\tau f)e=\bot$ by definition of τ.

<u>Case 2</u>: $e\varepsilon A$. Then $(\tau f)e=e$, and $e\varepsilon C$ by definition C1.

<u>Case 3</u>: $e=k(e_1,\cdots,e_n)$ for a unique n-place $k\varepsilon K$ and unique $e_1,\cdots,e_n\varepsilon E$. Two subcases arise.

i) $k\neq ap$. Then $(\tau f)e=k(fe_1,\cdots,fe_n)$ by definition of τ, and by induction $fe_i\varepsilon C\cup\{\bot\}$. Thus by definitions C2 and CS2, $(\tau f)e\varepsilon C\cup\{\bot\}$.

ii) $k=ap$, so $e=ap(e_1,e_2)$. Then $(\tau f)e=f(\rho fe_1)fe_2$ by definition of τ. By induction, $f(\rho fe_1)fe_2\varepsilon C\cup\{\bot\}$.

By computational induction $\mu e\varepsilon C\cup\{\bot\}$, where μ=least fixed point of τ.

Prop. 2. a) $\mu\bot=\bot$

b) $\mu a=a$, all $a\varepsilon A$

c) $\mu k(e_1,\cdots,e_n)=k(\mu e_1,\cdots,\mu e_n)$, $k\neq ap$

d) $\mu ap(e_1,e_2)=\mu(\rho\mu e_1)\mu e_2$

Proof. Use the fact that $\mu=\tau\mu$ [μ is a fixed point of τ], together with the definition of τ.

Prop. 3. For all $e \varepsilon C \cup \{\perp\}$, $\mu e = e$.

Proof. We show that if for every component e' of e, $e' \varepsilon C \cup \{\perp\}$ implies $\mu e' = e'$, then $\mu e = e$ for all $e \varepsilon C \cup \{\perp\}$. The proposition follows by structural induction. Definition CS3 leads to three distinct cases.

<u>Case 1</u>: $e = \perp$. Then $\mu e = e$ by proposition 2a.

<u>Case 2</u>: $e \varepsilon A$. Then $\mu e = e$ by proposition 2b.

<u>Case 3</u>: $e = k(e_1, \cdots, e_n)$ for a unique n-place $k \varepsilon K$ and unique $e_1, \cdots, e_n \varepsilon E$, and we assume the proposition holds for each e_i, $1 \le i \le n$. If $k = ap$ then $e \not\varepsilon C$, so assume $k \neq ap$. By definition C1-C3, if $e \varepsilon C$ then each $e_i \varepsilon C$, $1 \le i \le n$. Then by induction $\mu e_i = e_i$, $1 \le i \le n$. Thus $\mu e = k(\mu e_1, \cdots, \mu e_n) = k(e_1, \cdots, e_n) = e$.

Prop. 4. $C \cup \{\perp\}$ is the set of fixed points of μ, so $\mu = \mu \circ \mu$ [μ is idempotent].

Proof. By proposition 3, $\perp$ and the elements of C are all fixed points of μ. Conversely, if $e = \mu e$ is a fixed point of μ, then by proposition 1, $e \varepsilon C \cup \{\perp\}$. Thus μ, whose range is equal to the set of its own fixed points, is idempotent.

Prop. 5. e is reducible [for some e', eRe'] if and only if $e \not\varepsilon C \cup \{\perp\}$.

Proof. The proof is by structural induction. We must show that the proposition holds for each expression, provided that it holds for all components of the given expression. By definition CS3, there are three distinct cases to consider.

<u>Case 1</u>: $e = \perp$. By inspection of definition R1-R2, e is not

reducible. Further, $e \in C \cup \{\perp\}$. Thus e satisfies the proposition.

Case 2: $e \in A$. [Same as case 1 above.]

Case 3: $e = k(e_1, \cdots, e_n)$, for a unique n-place $k \in K$ and unique $e_1, \cdots, e_n \in E$. There are two subcases of interest.

i) $k \neq ap$. Suppose e is reducible [we show $e \notin C \cup \{\perp\}$]. By definition R1-R2, some component e_i of e is also reducible. By induction, $e_i \notin C \cup \{\perp\}$. Thus by definition C1-C3, $e \notin C$ [and of course $e \neq \perp$ by assumption].

Conversely, suppose $e \notin C \cup \{\perp\}$ [we show e is reducible]. Inspection of definition C1-C3 shows we must have $e_i \notin C$ for some component e_i of e. Further, $e_i \neq \perp$ since k is $\perp$-preserving [definition CS2]. By induction, $e_i R e_i'$ for some $e_i' \in E$, so $eRk(e_1, \cdots, e_i', \cdots, e_n)$ follows from definition R2.

ii) $k = ap$, so $e = ap(e_1, e_2) \neq \perp$. Then $e \notin C \cup \{\perp\}$, so we must find $e' \in E$ with eRe'. If both $e_1 \in C$ and $e_2 \in C$, definition R1 gives us $eR(\rho e_1)e_2$. So suppose $e_1 \notin C$. By definition CS2, $e_1 = \perp$ is impossible, so by induction there exists $e_1' \in E$ with $e_1 R e_1'$. Thus $eRap(e_1', e_2)$ by definition R2. [The final possibility, $e_1 \in C$ but $e_2 \notin C$, is treated the same way.]

Prop. 6. If e is reducible to e', then $\mu e=\mu e'$ [R preserves μ].

Proof. We show the proposition holds for every expression whose components all satisfy the proposition. Then we can conclude by structural induction that the proposition holds for all $e\varepsilon E$. Definition CS3 partitions E into three classes: $\{\perp\}$, A, and the "constructed expressions".

<u>Case 1</u>: $e=\perp$. Then by proposition 5, e is irreducible.

<u>Case 2</u>: $e\varepsilon A$. Then $e\varepsilon C$, so again by proposition 5, e is irreducible.

<u>Case 3</u>: $e=k(e_1,\cdots,e_n)$ for a unique n-place $k\varepsilon K$ and unique $e_1,\cdots,e_n\varepsilon E$. We consider two subcases.

i) $k\neq ap$. Suppose eRe'. Then by definition R1-R2, we must have e_iRe_i' for some component of e_i of e, and $e'=k(e_1,\cdots,e_i',\cdots,e_n)$. Now $\mu e_i=\mu e_i'$ by induction, so $\mu e=k(\mu e_1,\cdots,\mu e_i,\cdots,\mu e_n)=k(\mu e_1,\cdots,\mu e_i',\cdots,\mu e_n)=\mu e'$.

ii) $k=ap$, so $e=ap(e_1,e_2)$. If both $e_1\varepsilon C$ and $e_2\varepsilon C$, then $eR(\rho e_1)e_2$ by definition R1. Further, $\mu e_1=e_1$ and $\mu e_2=e_2$ [proposition 3], so $\mu e=\mu(\rho\mu e_1)\mu e_2=\mu(\rho e_1)e_2$. On the other hand, if one of the components e_1,e_2 is nonconstant then it must be reducible. Reasoning similar to case i above shows that e is reducible to an e' with $\mu e=\mu e'$.

Prop. 7. If $\mu e\neq\perp$, then $\delta e\neq\perp$.

Proof. The proof is by "parallel computational induction": if $P(\Omega,\Omega)$, and $P(f,g)$ implies $P(\tau f,\sigma g)$ for an admissible predicate P, we can deduce $P(f_\tau,g_\sigma)$, where f_τ,g_σ are the least fixed points of τ, σ respectively. Our predicate $P(f,g)$ is: if $fe\neq\perp$, then $fe=\mu e$ and $ge\neq\perp$. [The clause relating f and μ is needed in the induction step.] τ and σ are the defining functionals of μ and δ, respectively.

<u>Basis</u>: $[P(\Omega,\Omega)]$. $fe=\Omega e=\perp$ for all $e\varepsilon E$, by definition of Ω.

<u>Induction</u>: Assume $P(f,g)$ holds; we must show $P(\tau f,\sigma g)$, that is, if $(\tau f)e\neq\perp$, then $(\tau f)e=\mu e$ and $(\sigma g)e\neq\perp$. By definition CS3, there are three distinct cases to consider.

<u>Case 1</u>: $e=\perp$. Then $(\tau f)e=\perp$ by definition of τ.

<u>Case 2</u>: $e\varepsilon A$. Then $(\tau f)e=e$ by definition of τ, and $e=\mu e$, by proposition 2b. Also $(\sigma g)e=0\neq\perp$, by definition of σ.

<u>Case 3</u>: $e=k(e_1,\cdots,e_n)$ for a unique n-place $k\varepsilon K$ and unique $e_1,\cdots,e_n\varepsilon E$. Two subcases arise.

i) $k\neq ap$. Assume $(\tau f)e\neq\perp$. Then

$(\tau f)e=k(fe_1\cdots,fe_n)$ [def. of τ; note $fe_i\neq\perp$ by CS2]

$=k(\mu e_1,\cdots,\mu e_n)$ [by the inductive assumption, $P(f,g)$]

$= \mu e$ [by proposition 2c].

Additionally, $(\sigma g)e\neq\perp$, for $(\sigma g)e=ge_1+\cdots+ge_n$ by definition of τ, and $ge_i\neq\perp$ by inductive assumption.

ii) $k=ap$, so $e=ap(e_1,e_2)$. Note that f must be $\perp$-preserving. [By inductive assumption, either $f\perp=\perp$ or $f\perp=\mu\perp$. But $\mu\perp=\perp$, by proposition 2a.] Thus if we assume $(\tau f)e$ $=f(\rho fe_1)fe_2\neq\perp$, then we must have $(\rho fe_1)fe_2\neq\perp$. By our general assumptions on ρ, $fe_1 \neq \perp$ and $fe_2 \neq \perp$. $fe_1 = \mu e_1$, $fe_2 = \mu e_2$, and $f(\rho fe_1)fe_2 = \mu(\rho fe_1)fe_2 = \mu(\rho\mu e_1)\mu e_2$ follow by induction and substitution. This proves $(\tau f)e = \mu e$.

We must still show that $(\sigma g)e = 1 + ge_1 + ge_2 + g(\rho\mu e_1)\mu e_2 \neq \perp$. Since $fe_1 \neq \perp$ and $fe_2 \neq \perp$, $ge_1 \neq \perp$ and $ge_2 \neq \perp$ follow by induction. Also $g(\rho\mu e_1)\mu e_2 \neq \perp$, because $f(\rho fe_1)fe_2 \neq \perp$, $fe_1 = \mu e_1$, and $fe_2 = \mu e_2$ [as was shown above]. Thus $(\sigma g)e \neq \perp$.

Prop. 8. For all $e \in C$, $\delta e = 0$

Proof. By structural induction on e. There are, by CS3, three distinct cases.

<u>Case 1</u>: $e = \perp$. Then $e \notin C$, by definition C1-C3.

<u>Case 2</u>: $e \in A$. Then $\delta e = (\sigma\delta)e$, since δ is a fixed point of σ. But $(\sigma\delta)e = 0$ for all $e \in A$, by definition of σ.

<u>Case 3</u>: $e = k(e_1, \cdots, e_n)$, for unique n-place $k \in K$ and $e_1, \cdots, e_n \in E$. Two subcases arise.

i) $k \neq ap$. If $e \in C$ then by definition C1-C3, $e_i \in C$ for $1 \leq i \leq n$. By induction $\delta e_i = 0$, $1 \leq i \leq n$. So $\delta e = \delta e_1 + \cdots + \delta e_n = 0$.

ii) $k = ap$. Then $e = ap(e_1 e_2) \notin C$.

Prop. 9. If $\mu e \neq \perp$ and eRe', then $\delta e = \delta e' + 1$

Proof. Suppose $\mu e \neq \perp$ and eRe'. By proposition 6, $\mu e' = \mu e \neq \perp$; hence by proposition 7, $\delta e \neq \perp$ and $\delta e' \neq \perp$. We prove the relationship $\delta e = \delta e' + 1$ using structural induction on e. By CS3, there are three distinct cases to consider.

Case 1: $e=\perp$. Then $\mu e=\perp$, by proposition 2a.

Case 2: $e\varepsilon A$. Then $e\varepsilon C$, so e is irreducible by proposition 5.

Case 3: $e=k(e_1,\cdots,e_n)$ for unique n-place $k\varepsilon K$ and $e_1,\cdots,e_n\varepsilon E$. There are two subcases.

i) $k\neq ap$. Suppose eRe'. By definition R2, $e'=k(e_1,\cdots,e_i',\cdots,e_n)$ for some $e_i'\varepsilon E$ with e_iRe_i'. Since $\mu e=k(\mu e_1,\cdots,\mu e_i,\cdots,\mu e_n)\neq\perp$ and k is $\perp$-preserving [by CS2], $\mu e_i\neq\perp$. By induction, $\delta e_i=\delta e_i'+1$. Thus

$$\begin{aligned}\delta e&=\delta e_1+\cdots+\delta e_i+\cdots+\delta e_n\\&=\delta e_1+\cdots+(\delta e_i'+1)+\cdots+\delta e_n\\&=(\delta e_1+\cdots+\delta e_i'+\cdots+\delta e_n)+1=\delta e'+1.\end{aligned}$$

ii) $k=ap$, so $e=ap(e_1,e_2)$ for unique $e_1,e_2\varepsilon E$. If $e_1\notin C$ and/or $e_2\notin C$, the reasoning is analogous to case i above. So here assume $e_1,e_2\varepsilon C$ and eRe'. By definition R1 we must have $e'=(\rho e_1)e_2$. Thus

$$\begin{aligned}\delta e&=1+\delta e_1+\delta e_2+\delta(\rho\mu e_1)\mu e_2 && [\text{since } \delta e=(\sigma\delta)e]\\&=1+0+0+\delta(\rho\mu e_1)\mu e_2 && [\text{by proposition 8}]\\&=1+\delta(\rho e_1)e_2 && [\text{by proposition 3}]\\&=1+\delta e'.\end{aligned}$$

Prop. 10. If $\mu e\neq\perp$, then every sequence of reductions on e converges to μe [in exactly δe steps].

Proof. A sequence of reductions on e is a [finite or infinite] sequence of expressions $e_0,e_1,e_2,\cdots$ with $e_0=e$ and, for all $i\geq 0$, e_iRe_{i+1}.

Thus we wish to show that if $\mu e \neq \perp$, the sequence is of length δe [which, by proposition 7, is not $\perp$], and that $e_{\delta e} = \mu e$.

Since $\mu e_0 = \mu e \neq \perp$ and R preserves μ [proposition 6], it follows by induction that the meanings of all expressions in the sequence are identical. If, for some $n \geq 0$, there is no e' with $e_n R e'$, then $e_n \varepsilon C$ [proposition 5], so $\mu e = \mu e_n = e_n$ [proposition 3]. Thus if the sequence terminates, it terminates in μe.

But the sequence must terminate, for $\delta e_0, \delta e_1, \delta e_2, \cdots$ is a decreasing sequence of natural numbers [proposition 9]. Suppose e_n is the last term, that is, $\mu e_n = e_n \varepsilon C$. Then $\delta e_n = 0$ [proposition 8], so $\delta e_0 = \delta e_1 + 1 = (\delta e_2 + 1) + 1 = \cdots = \delta e_n + n = n$, the number of steps in any reduction sequence for e.

ACKNOWLEDGEMENTS

I am indebted to John Backus and Jim Gray for their careful reading and constructive criticism of the manuscript, and to Barry Rosen for much information on complete posets and their application to closed applicative languages.