X. Another Approach to Confluence, Termination, Optimality

1. Pseudoresidual Maps

Certain interesting SRSs are not closed with respect to any residual map satisfying Definition 22 (p. 23). For instance, β-reduction in the λ-calculus may be represented straightforwardly by a SRS (see Section 4, also Rosen [Ro71] (Def. 9.7, p. 4-12) with a natural map for which clause (2) of Definition 22 fails. Such a map is called a pseudoresidual map.

When working with a pseudoresidual map, it is convenient to allow reduction steps which reduce dependent sets of adresses.

Definition 19´

Let $M \varepsilon \mathcal{P}(\text{domain}A)$ be a set of redexes in A, then

$A \xrightarrow[M]{} B$ iff $A \xrightarrow[x_1]{} \xrightarrow[x_2]{} \dots \xrightarrow[x_m]{} B$

where $M=\{x_1,\dots,x_m\}$ and $i<j \Rightarrow \neg x_i \ \underline{\text{anc}}\ x_j$.

Definition 19´ generalizes Definition 19 (p. 21).

$A \xrightarrow[M]{} B$ means that A may be reduced to B by reducing the redexes M from innermost to outermost. See Cadiou for another treatment of reduction at a set of redexes which might not be independent [Cad72] (5.3, pp. 87-89).

Now we may define a pseudoresidual map, similar to Rosen's implicitly defined "pseudoresidue map" [Ro71] (p. 4-16).

Definition 22´

A pseudoresidual map is a function

$r: A \to (\mathcal{P}^* \to \mathcal{P}(\mathcal{P}^*))$ such that

(1) $(x \text{ is a redex in } A \wedge y \varepsilon r\langle A=B\rangle x) \Rightarrow y$ is a redex in B,

(2) $u \neq v \Rightarrow \neg\exists_{w,x \varepsilon r\langle A=B\rangle w,\ y,z \varepsilon r\langle A=B\rangle v}\ w \ \underline{\text{anc}}\ y \wedge z \ \underline{\text{anc}}\ x$,

(3) $r\langle A=B\rangle() = \emptyset$,

(4) $u \ \underline{\text{anc}\neq}\ v \Rightarrow \neg\exists_{w \varepsilon r\langle A=B\rangle u, x \varepsilon r\langle A=B\rangle v}\ x \ \underline{\text{anc}}\ w$.

Note that clauses (1) and (2) are weakened from Definition 22, and residual sets are not required by the functionality of r to be independent sets. Because of the additional clause (4), there are residual maps which are not pseudoresidual maps as well as vice versa.

The extended pseudoresidual map $\hat{r}$ may be defined by modifying Definition 23 (p. 25).

Definition 23′

If r is a pseudoresidual map, then the extended pseudoresidual map $\hat{r}$ is a function

$\hat{r}: F\times(P(P^*))^* \to (P(P^*)\to P(P^*))$ such that

$$\hat{r}[A,()]\{y\}=\hat{r}[A,(\emptyset)]\{y\}=\{y\}$$

$$\hat{r}[A,(\{x\})]\{y\}=\begin{cases}\{x\cdot w \mid w\varepsilon r<A/x=C>z\} & \text{if } <A/x=C>\varepsilon A \text{ and } y=x\cdot z\\ \{y\} & \text{otherwise}\end{cases}$$

$$\hat{r}[A,(\{x\})]N=\bigcup_{y\varepsilon N}\hat{r}[A,(\{x\})]\{y\}$$

$$\hat{r}[A,(M_1,\ldots,M_{n+1})]N=\hat{r}[B,(M_{n+1})](\hat{r}[A,(M_1,\ldots,M_n)]N)$$
$$\text{where } A\xrightarrow[M_1]{}\ldots\xrightarrow[M_n]{}B$$

$$\hat{r}[A,(M)]N=\hat{r}[A,(\{x_1\},\ldots,\{x_m\})]N$$
$$\text{where } M=\{x_1,\ldots,x_n\} \text{ and } i<j \Rightarrow \neg x_i \ \underline{\text{anc}}\ x_j.$$

Only the last clause and the functionality of $\hat{r}$ differ from Definition 23.

Lemmas 5, 6 and 7 (pp. 21, 25, 27) still hold for the more general reduction steps of Definition 19′ (p. 89) and pseudoresidual maps. Lemma 8 must be weakened slightly.

Lemma 8′

Let $\hat{r}$ be an extended pseudoresidual map.

Then (1) $\hat{r}[A,(M_1,\ldots,M_n)]N_1 \cup \hat{r}[A,(M_1,\ldots,M_n)]N_2 = \hat{r}[A,(M_1,\ldots,M_n)](N_1 \cup N_2)$

(2) $N_1 \cap N_2=\emptyset \Rightarrow \hat{r}[A,(M_1,\ldots,M_n)]N_1 \cap \hat{r}[A,(M_1,\ldots,M_n)]N_2=\emptyset$

(3) $N\subseteq M \Rightarrow \hat{r}[A,(M)]N=\emptyset$.

Proof: Analogous to proof of Lemma 8 (p. 27). □

2. The General Pseudoresidual Lemma

In order to extend the results of Chapters IV, V and VI to SRSs with pseudoresidual maps, we need an analog to the General Residual Lemma (Lemma 12, p. 35). A modified form of closure, along with commutativity, provides a sufficient condition.

Definition 25´

A SRS $\langle \Sigma, \mathcal{F}, \rightarrow, \mathcal{A} \rangle$ is pseudoclosed with respect to pseudoresidual map r iff the following holds.

Let $\langle A=B \rangle \varepsilon \mathcal{A}$, $x \neq ()$ and $A \xrightarrow[x]{} A'$.

Then there is a B' such that

$\langle A'=B' \rangle \varepsilon \mathcal{A}$ and $B \xrightarrow[r\langle A,B \rangle x]{} B'$.

Definition 25´ is just the first clause of Definition 25 (p. 30).

Definition 38´

Let $\mathcal{E}=\langle \Sigma, \mathcal{F}, \rightarrow, \mathcal{A} \rangle$ be pseudoclosed with respect to r.

$\mathcal{E}$ commutes with respect to r iff:

for all $\langle A=B \rangle \varepsilon \mathcal{A}$, x and y redexes in A,

$\hat{r}[A,(\{x\},\{()\})]\{y\}=\hat{r}[A,(\{()\},r\langle A=B \rangle x)]\{y\}$.

Definition 38´ is the same as Definition 38 (p. 59) except that $\mathcal{E}$ is pseudoclosed and r is a pseudoresidual map. Notice that if r is both a residual map and a pseudoresidual map, then $\mathcal{E}$ is closed and commutative with respect to r iff $\mathcal{E}$ is pseudoclosed and commutative.

As much as possible, we will extend the work of Chapters IV through VI, replacing independent reduction steps by arbitrary (in the sense of Def. 19´) reduction steps, residual maps by pseudoresidual maps, and closure by pseudoclosure and commutativity. Lemmas 9 through 12 extend as follows.

Lemma 9´

Let $\langle \Sigma, \mathcal{F}, \rightarrow, \mathcal{A} \rangle$ be commutative with respect to pseudoresidual map r.
Assume that $A \xrightarrow[N]{} A'$ and $A \xrightarrow[M_1]{} \dots \xrightarrow[M_n]{} B$.
Then each $y \varepsilon \hat{r}[A,(M_1,\dots,M_n)]N$ is a redex in B.

Proof: Analogous to Lemma 9 (p. 83). □

Lemma 10´

Let $<\Sigma, F, \rightarrow, A>$ commute with respect to r.

Let $A \xrightarrow[z]{} B$ and $A \xrightarrow[M]{} A'$, where $\forall_{y \varepsilon M}\ \neg(y\ \underline{anc}\ z)$.

Then there is a B' such that

$A \xrightarrow[z]{} B'$ and $B \xrightarrow[\hat{r}[A,(\{z\})]M]{} B'$

Proof: Order M as $\{x_1,\ldots,x_m\}$ so that $i<j \Rightarrow$ (1) $\neg x_i\ \underline{anc}\ x_j$ and (2) $\neg\exists_{u\varepsilon\hat{r}[A,(\{z\})]x_i,\ v\varepsilon\hat{r}[A,(\{z\})]x_j}\ u\ \underline{anc}\ v$.

Such an ordering is possible because of Clauses (2) and (4) of Definition 22´ (p.). $A \xrightarrow[x_1]{} \ldots \xrightarrow[x_m]{} B_1$ by Definition 19´ (p. 89) and (1) above.

Let $A_0=A$, $A_m=B$ and $A_0 \xrightarrow[x_1]{} A_1 \ldots A_{m-1} \xrightarrow[x_m]{} A_m$.

Now, by m repeated applications of pseudoclosure, there is a B such that

$A' \xrightarrow[\hat{r}[A_0,(\{z\})]\{x_1\}]{} \cdots \xrightarrow[\hat{r}[A_{m-1},(\{z\})]\{x_m\}]{} B'$

The steps above justify Figure 31.

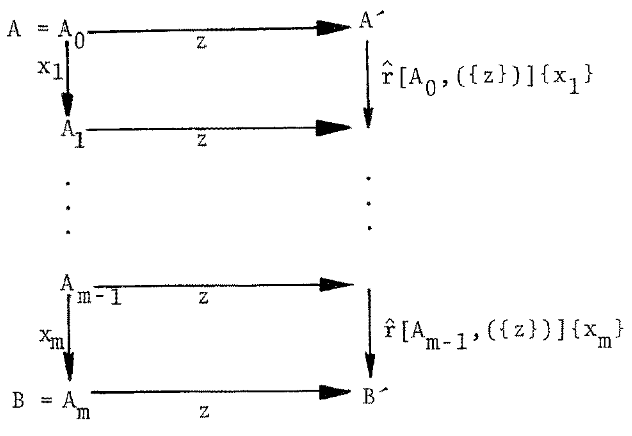


Figure 31

By commutativity and (2) above, for all i.

$\hat{r}[A_{i-1},(\{z\})]\{x_i\} = \hat{r}[A_0,(\{x_1\},\ldots,\{x_{i-1}\},\{z\})]\{x_i\}$

$=\hat{r}[A_0(\{z\},\hat{r}[A_0,(\{z\})]\{x_1\},\ldots,\hat{r}[A_{i-2},(\{z\})]\{x_{i-1}\})]\{x_i\}$

$=\hat{r}[A_0,(\{z\})]\{x_i\} = \hat{r}[A,(\{z\})]\{x_i\}$.

By (2), $A' \xrightarrow[\hat{r}[A,(\{z\})]\{x_1\}\ldots\ \hat{r}[A,\{z\}]\{x_m\}]{} B'$, so

$A' \xrightarrow[\hat{r}[A,(\{z\})]M]{} B'$. □

Lemma 11´

Let $<\Sigma,F,\to,A>$ commute with respect to pseudoresidual map r, and let M and N be sets of redexes such that

$\forall_{x\varepsilon M,\ y\varepsilon N}\ \neg(y\ \underline{\text{anc}}\ x)$.

Also let $A\xrightarrow[M]{}B$ and $A\xrightarrow[N]{}C$.

Then there is a D such that $B\xrightarrow[\hat{r}[A,(M)]N]{}D$ and $C\xrightarrow[M]{}D$.

Proof: Almost the same as the proof of Lemma 11 (p. 33). In the third line, replace "Let zεM" by "Let zεM have no proper ancestors in M." □

Lemma 12´ (General Pseudoresidual Lemma)

Let $E=<\Sigma,F,\to,A>$ commute with respect to pseudoresidual map r.

Assume $A\xrightarrow[M]{}B$ and $A\xrightarrow[N]{}C$.

Then there is a D such that

$$B\xrightarrow[\hat{r}[A,(M)]N]{}D \text{ and } C\xrightarrow[\hat{r}[A,(N)]M]{}D.$$

Figure 32 sketches the Lemma.

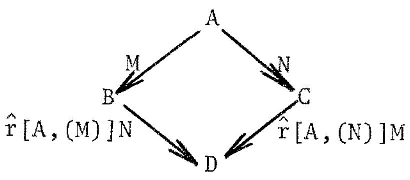


Figure 32

Proof: By induction on $|M|+|N|$.

Basis step: If $|M|+|N|=0$, the lemma follows directly with A=B=C=D.

Induction step: Assume that the lemma holds for smaller sets M and N.

Partition M into two pieces:

$M_{out}=\{x|\ x\varepsilon M \wedge \forall_{y\varepsilon M \cup N}\ \neg(y\ \underline{anc\neq}\ x)\}$

$M_{in}=M-M_{out}=\{x|\ x\varepsilon M \wedge \exists_{y\varepsilon M \cup N}\ y\ \underline{anc\neq}\ y\}$

Similarly, partition N into

$N_{out}=\{y|\ y\varepsilon N \wedge \forall_{x\varepsilon M \cup N}\ \neg(x\ \underline{anc\neq}\ y)\}$

$N_{in}=N-N_{out}=\{y|\ y\varepsilon N \wedge \exists_{x\varepsilon M \cup N}\ x\ \underline{anc\neq}\ y\}$.

Figure 33 summarizes the induction step.

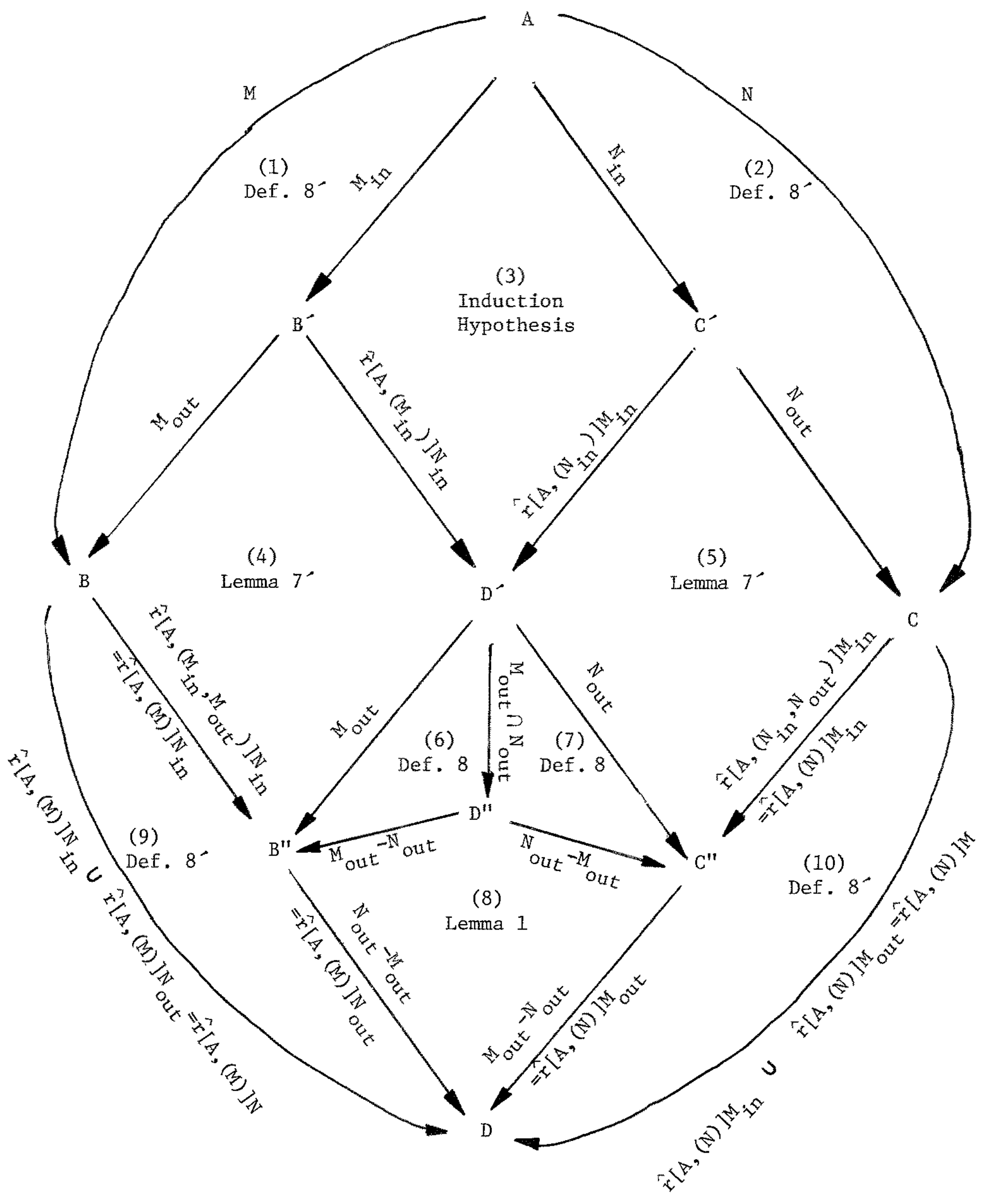

A
M
N
(1)
Def. 8´
(2)
Def. 8´
(3)
Induction
Hypothesis
B´
C´
B
D´
C
(4)
Lemma 7´
(5)
Lemma 7´
(6)
Def. 8
(7)
Def. 8
D"
B"
C"
(8)
Lemma 1
(9)
Def. 8´
(10)
Def. 8´
D

Figure 33

By Definition 19´ (p. 89), there are B' and C' such that:

(1) $A \xrightarrow[M_{in}]{} B' \xrightarrow[M_{out}]{} B$

(2) $A \xrightarrow[N_{in}]{} C' \xrightarrow[N_{out}]{} C$

(3) M_{out} and N_{out} cannot both be empty, so

$|M_{in}| + |N_{in}| < |M| + |N|$.

Now, by the induction hypothesis, there is a D' such that

$B' \xrightarrow[\hat{r}[A,(M_{in})]N_{in}]{} D'$ and $C' \xrightarrow[\hat{r}[A,(N_{in})]M_{in}]{} D'$.

(4) By Lemma 6 (p.   ), since M_{out} has no ancestors in $M_{in} \cup N_{in}$, it has no ancestors in $\hat{r}[A,(M_{in})]N_{in}$.

So, by Lemma 11´ (p.   ) applied to

$B' \xrightarrow[M_{out}]{} B$ and $B' \xrightarrow[\hat{r}[A,(M_{in})]N_{in}]{} D'$,

there is a B" such that

$B \xrightarrow[\hat{r}[A,(M_{in},M_{out})]N_{in}]{} B''$ and $D' \xrightarrow[M_{out}]{} B''$.

(5) Similarly, there is a C" such that

$C' \xrightarrow[\hat{r}[A,(N_{in},N_{out})]M_{in}]{} C''$ and $D' \xrightarrow[N_{out}]{} C''$.

(6) $\perp M_{out}$ and $\perp N_{out}$, so, by Definition 19 (p. 21),

there is a D" such that

$D \xrightarrow[M_{out} \cap N_{out}]{} D'' \xrightarrow[M_{out}-N_{out}]{} B''$,

(7) $D' \xrightarrow[M_{out} \cap N_{out}]{} D'' \xrightarrow[N_{out}-M_{out}]{} C''$.

(8) Now, $(M_{out}-N_{out}) \perp (N_{out}-M_{out})$, so,

by Lemma 1, there is a D such that

$B'' \xrightarrow[N_{out}-M_{out}]{} D$ and $C'' \xrightarrow[M_{out}-N_{out}]{} D$

(9) By Definition 23′ (p. 90),

$\hat{r}[A,(M_{in},M_{out})]N_{in}=\hat{r}[A,(M)]N_{in}$

and $N_{out}-M_{out}=\hat{r}[A,(M)]N_{out}$.

By Definition 19′,

$B \xrightarrow[\hat{r}[A,(M)]N_{in} \cup \hat{r}[A,(M)]N_{out}]{} D$.

By Definition 23′,

$\hat{r}[A,(M)]N_{in} \cup \hat{r}[A,(M)]N_{out}=\hat{r}[A,(M)]N$, so

$B \xrightarrow[\hat{r}[A,(M)]N]{} D$

(10) By a symmetric argument, $C \xrightarrow[\hat{r}[A,(N)]M]{} D$. □

3. Confluence, Termination, Optimality

Given the General Pseudoresidual Lemma, all of the work of Chapter IV on the confluence property, that of Chapter V on complete sequences and strictly innermost sequences, and all of Chapter VI on costs of noncopying sequences extends directly.

Theorems 6′, 8′, 11′, 12′, 13′, 15′

Let $\mathcal{E}=\langle\Sigma,\mathcal{F},\rightarrow,\mathcal{R}\rangle$ commute with respect to pseudoresidual map r. Then:

(6′) $\mathcal{E}$ has the confluence property (Def. 24, p. 28).

(8′) The set of complete sequences, $\mathcal{C}$, is terminating for $\mathcal{E}$ (Defs. 26, 28, pp. 40, 42).

(11′) If r is innermost preserving, then strictly innermost sequences (members of $\mathcal{I}^s$) are infinite whenever possible (Defs. 34, 35, pp. 52).

(12′) If r is innermost preserving, then noncopying strictly innermost sequences are maximally costly (Defs. 34, 35, 37, pp. 52, 55).

(13′, 15′) If $\mathcal{E}$ is d-outer, then some strictly d-outermost noncopying sequence is optimal (Defs. 34, 37, 41, 44, pp. 52, 55, 63, 65).

Proof: Analagous to proofs of Theorems 6, 8, 11, 12, 13, 15 (pp. 38, 42, 53, 60, 63, 65). □

Theorems 12´, 13´ and 15´ must be applied with care, since SRSs with pseudoresidual maps will generally fail to satisfy the uniform cost assumptions at the beginning of Chapter VI (p. 54).

4. β-Reduction in the λ-Calculus

Let $\Sigma=\{\lambda,\underline{AP}\} \cup V$ where V is an infinite set of variable symbols. Let F' contain Σ-trees which represent λ-expressions, that is,

$$
\begin{aligned}
F' = \{A \mid\ & A \text{ is a } \Sigma\text{-tree and} \\
& Ax\in V \Rightarrow x\cdot(1)\notin \underline{\text{domain}}A \\
& Ax=\underline{AP} \Rightarrow x\cdot(1),x\cdot(2)\in \underline{\text{domain}}A \\
& \qquad\qquad x\cdot(3)\notin \underline{\text{domain}}A \\
& Ax=\lambda \Rightarrow A(x\cdot(1))\in V \\
& \qquad\qquad x\cdot(2)\in \underline{\text{domain}}A \\
& \qquad\qquad x\cdot(3)\notin \underline{\text{domain}}A\}.
\end{aligned}
$$

The correspondence between λ-terms [Ste72] ((i)-(iv), p. 41) and trees in F' is straightforward. The notions of free and bound occurrences of variables carry over directly from λ-terms [Ste72] (p. 41).

To avoid problems of renamings of bound variables, as well as "collisions" and "captures" of variables [CF58] (2, pp. 89-91), let $V=\{v_i \mid i\in P\} \cup \{w_x \mid x\in P^*\}$ where the v_i will represent free variables and the w_x will represent bound variables. Let $F\subseteq F'$ contain all the trees A in which free occurrences of variables are all in $\{v_i \mid i\in P\}$ and

$$Ax=\lambda \Rightarrow A(x\cdot(1))=w_x.$$

F contains exactly one expression from each α-equivalence class [CF58] ((α), p. 91) (two λ-terms are α-equivalent when they are the same up to renaming of bound variables). Let $C\in F'$ be an expression such that, for all descendants z of x, w_z is not free in C (i.e., C is a subexpression of some expression in F). Define <u>rename</u>(C,x) to be the result of renaming bound variables in C to produce an expression which may be used as the subtree at x of a tree in F:

$\underline{\text{rename}}(v_i,x)=v_i$ and $\underline{\text{rename}}(w_y,x)=w_y$

$\underline{\text{rename}}(\underline{\text{AP}}(A,B),x)=\underline{\text{AP}}(\underline{\text{rename}}(A,x\cdot(1)),\ \underline{\text{rename}}(B,x\cdot(2)))$

$\underline{\text{rename}}(\lambda(w_y,A),x)=$

$\qquad\lambda(w_x,\underline{\text{rename}}(A(z\leftarrow w_x|\ z\varepsilon A^{-1}w_y),x\cdot(2)))$.

Now define $\mathcal{A}$ to mimic β-reduction of expressions in $\mathcal{F}$ [Ste72] (≥, p. 44):

$\mathcal{A}=\{<\underline{\text{AP}}(\lambda(w_{t\cdot(1)},A),B)=\underline{\text{rename}}(A(u\leftarrow B|\ u\varepsilon A^{-1}w_{t\cdot(1)}),t)>|$
$\qquad\underline{\text{AP}}(\lambda(w_{t\cdot(1)},A),B)$ is a subtree of a tree in $\mathcal{F}\}$.

A pseudoresidual map r may be defined straightforwardly:

$$r<\underline{\text{AP}}(\lambda(w_{t\cdot(1)},C),D)=\underline{\text{rename}}(C(u\leftarrow D|\ u\varepsilon C^{-1}w_{t\cdot(1)}),t)>x=\begin{cases}\{z\} \text{ if } u=(1,2)\cdot z\\ \{u\cdot z|\ u\varepsilon C^{-1}w_{t\cdot(1)}\} \text{ if } x=(2)\cdot z\\ \emptyset \text{ otherwise.}\end{cases}$$

r gives exactly the residuals defined by Church [Ch41] (pp. 18-19) [CF58] (Def. 1, pp. 115-116).

Lemma 25

Let $\mathcal{E}=<\Sigma,\mathcal{F},\rightarrow,\mathcal{A}>$ be a SRS representing β-reduction, with Σ, $\mathcal{F}$, $\mathcal{A}$ defined above. Let r be the pseudoresidual map defined above.

Then $\mathcal{E}$ commutes with respect to r.

Proof: To show that $\mathcal{E}$ is pseudoclosed with respect to r (Def. 25′, p. 91), let $<A=B>\varepsilon\mathcal{A}$ where

$A=\underline{\text{AP}}(\lambda(w_{t\cdot(1)},C),D)$ and

$B=\underline{\text{rename}}(C(u\leftarrow D|\ u\varepsilon C^{-1}w_{t\cdot(1)}),t)$.

Let $x\neq()$ and $A\xrightarrow[x]{}A'$.

Since $x\neq()$, we have $A=\underline{\text{AP}}(\lambda(w_{t\cdot(1)},C'),D')$.

Consider three cases:

(1) $x=(1,2)\cdot z$ (x is an address in the subtree C).
Then $D=D'$, $C \xrightarrow[z]{} C'$ and $r\langle A=B\rangle x=\{z\}$.
Since x is a redex in A, $A_x=C_z \notin V$.
So, $B=\underline{\text{rename}}(C(u\leftarrow D \mid u\varepsilon C^{-1}w_{t\cdot(1)}),t) \xrightarrow[r\langle A=B\rangle x]{}$
$\underline{\text{rename}}(C'(u\leftarrow D \mid u\varepsilon C'^{-1}w_{t\cdot(1)}),t)=B'$
and $\langle A'=B'\rangle \varepsilon A$ by straightforward calculations on tree addresses.

(2) $x=(2)\cdot z$ (x is an address in the subtree D).
Then $C=C'$, $D \xrightarrow[z]{} D'$ and $r\langle C=D\rangle x=\{u\cdot z \mid u\varepsilon C^{-1}w_{t\cdot(1)}\}$.
So, $B=\underline{\text{rename}}(C(u\leftarrow D \mid u\varepsilon C^{-1}w_{t\cdot(1)}),t) \xrightarrow[r\langle A=B\rangle x]{}$
$\underline{\text{rename}}(C(u\leftarrow D \mid u\varepsilon C^{-1}w_{t\cdot(1)}),t)=B$
and $\langle A'=B'\rangle \varepsilon A$.

(3) $\neg\exists_z\ x=(1,2)\cdot z \wedge x=(2)\cdot z$.
Then $x=(1)$ or $(1,1)$ and $Ax=\lambda$ or $w_{y\cdot(1)}$.
In either case, x is not a redex in A, contradicting the assumption $A \xrightarrow[x]{} A'$.

Now, to show commutativity, let A, B, C, D, A', B', C', D' be as above, and let $x,y \ \varepsilon$ <u>domain</u>A, x a redex in A. Suppose first that $Ay=v_i$, where v_i appears only at the address y in A. By Definition 25′ (p. 91) and the definition of r above, if

$A \xrightarrow[M_1]{} \cdots \xrightarrow[M_n]{} E$, then $Es=v_i$ iff $s\varepsilon\hat{r}[A,(M_1,\ldots,M_n)]\{y\}$.
In particular,

$A \xrightarrow[\{x\}]{} A' \xrightarrow[\{()\}]{} B'$, so
$B\ s=v_i$ iff $s\varepsilon\hat{r}[A,(\{x\},\{()\})]\{y\}$.

Similarly, $A \xrightarrow[\{()\}]{} B \xrightarrow[r\langle A=B\rangle x]{} B'$ (by pseudoclosure), so
$B's=v_i$ iff $s\varepsilon\hat{r}[A,(\{()\},r\langle A=B\rangle x)]\{y\}$.

Therefore, $\hat{r}[A,(\{x\},\{()\})]\{y\}=\hat{r}[A,(\{()\}, r\langle A=B\rangle x)]\{y\}$. Since the residuals of y are the same whether or not $A_y=v_i$, the above equation holds in general. □

<u>Lemma 26</u>

E defined in Lemma 25 is d-outer (Def. 41, p. 63) for preorder d.

Proof: Let A and B be in F, $x,z \in$ <u>domain</u>A,
$x(dA)z$, $x \neq z$, x not a redex in A, $A \xrightarrow[z]{} B$.
If $Ax \neq \underline{AP}$, then $Bx \neq \underline{AP}$ and x is not a redex in B.
If $Ax = \underline{AP}$, then $A(x\cdot(1)) \neq \lambda$, and x will be a redex in B iff $B(x\cdot(1)) = \lambda$.

The only way for $B(x\cdot(1))$ to become λ is to have $z = x\cdot(1)$. But $x\cdot(1)$ immediately succeeds x in preorder (Def. 42, p. 64), so there can be no redex y in A with $x(dA)y(dA)x\cdot(1)$ and $y \neq x\cdot(1)$. □

<u>Theorem 22</u>

For the λ-calculus SRS $E = \langle \Sigma, F, \rightarrow, A \rangle$ defined above the following hold:

(1) The confluence property.

(2) Leftmost outermost noncopying sequences are optimal.

(3) Leftmost outermost noncopying reduction terminates whenever possible.

Proof:
(1) Theorem 6′ (p. 97), Lemma 25.
(2) Theorem 15′ (p. 97), Lemma 26.
(3) Theorem 15′, Lemma 26 and the uniqueness of leftmost outermost reduction sequences (see p. 65). □

(1) is well known for the usual formulation of the λ-calculus. A result similar to (3) was proved by Curry and Feys [CF58] (Cor. 1.1, p. 142). (2) is new, but must be applied with discretion, since there may not be an efficient implementation of noncopying reduction sequences for the λ-calculus (the pseudoresidual map r above does not satisfy the assumptions of Chapter IV, Section 1). For an application of SRS theory to η-reduction and δ-reduction, and another approach to β-reduction, see Rosen [Ro71] (Ch. 4, pp. 4-1 through 4-43).

Further Research

Further research in reduction strategies divides naturally into four areas: abstract replacement systems, rule schemata, implementation techniques, and applications.

(1) Abstract Replacement Systems

(a) An even more general formalism for studying reduction strategies is the replacement system of Sethi [Se74] (2, p. 673). Sethi and Rosen give theorems [Se74] (Th. 2.3, p. 675) [Ro73] (Th. 3.5, p. 164; Th. 3.8, p. 166) which guarantee the confluence property for unions of replacement systems. Staples [Sta77,1] also studies optimality in a general setting. These results should be extended to show how termination properties and costs of sequences are affected by combining replacement systems. See Newman [Ne42], Curry [Cu52], and Hindley [Hi69] [Hi74] for a topological approach to replacement systems.

(b) Extensions of SRS style results to replacement systems based on structures other than trees might be fruitful. See Ehrig and Rosen [ER77] and Staples [Sta77,2] for recent work on graph replacements.

(c) Within the SRS formalism, more general criteria for confluence, termination, optimality are desirable. Theorems 10 and 14 (pp. 50, 65) are especially suitable for generalization to SRSs with pseudoresidual maps.

(d) Weaker conditions than the confluence property (e.g., uniqueness of normal forms, or of certain normal forms) may be sufficient for some applications. Sufficient conditions weaker than closure might be found.

(e) In SRSs for which $\mathcal{O}^e$ (p. 45) is not terminating, there may be other classes of sequences which are terminating and which are more interesting than $\mathcal{C}$ (p. 42).

(f) Optimal sequences other than those in $\mathcal{O}_d^s$ (p. 65) should be developed for certain classes of SRSs. Results on the relative costs of nonoptimal strategies would be interesting in cases such as Lucid (see p. 87) where optimal sequences cannot be generated efficiently.

(g) Cost measures other than that of Definition 37 (p. 55) should be considered.

(h) In general, the problem of deciding equality of expressions is prohibitively difficult. Knuth and Bendix [KB70] consider this problem only in the case where all reduction sequences end in normal form. Significant generalizations could be very useful (see 2b). Kozen [Ko77] considers the special case where the set of equational axioms is finite.

(2) Rule Schemata.

(a) Sets of rule schemata may be very well-behaved even though they overlap. Pseudoresidual maps are probably more appropriate than residual maps for studying overlapping sets of schemata. Knuth and Bendix [KB70] (Cor., p. 275) obtain the confluence property for certain overlapping sets of schemata, but only when all reduction sequences end in normal forms. More general criteria are needed for these overlapping sets of schemata to guarantee confluence, termination and cost properties.

(b) Nontrivial extensions of schemata results to rule schemata with repeated variables on the left (e.g., [if then else (A,B,B)=B]) could be very helpful. Knuth and Bendix [KB70] handle such schemata, but only when all reduction sequences end in normal forms.

(c) Knuth and Bendix's [KB70] (Cor., p. 275; 6, pp. 276-277) methods for testing closure and, in some cases, generating an equivalent closed set of schemata from a nonclosed set should be extended to SRSs where reduction sequences may not terminate.

(3) Implementation Techniques.

(a) Clever programming tricks might lead to implementations of reduction superior to or more general than the pointer strategies sketched in Example 17 (p. 54) and Appendix A. For instance, an efficient implementation of noncopying reduction in the λ-calculus requires a more sophisticated approach. Graph replacement studies such as [ER77] and [Sta77,2] may be helpful. In LISP, a clever implementation may save a large amount of space [FW76,3].

(b) General problems of choosing reduction steps efficiently should be considered. For $\mathcal{O}_d^s$ (p. 65) with preorder d (p. 64) this is easy, but more general cases (see 1e, 1f) should be covered if possible.

(4) Applications.

(a) Other applicative programming languages, such as APL [Iv62] and the Red languages [Ba73] may be susceptible to SRS techniques.

(b) The SRS formalism might be used to study proofs in richer systems than pure equational logic. Good candidates would be a language of equations with implication, or systems of Horn clauses in the predicate calculus (see [Kow74] and [vEK76] for applications of Horn clauses to computing).

(c) Programming techniques for languages with outermost interpreters include some methods not available in normal languages. These methods should be explored. See [FW76,2] to start.

(d) See [76] for an application of equational definition to Hewitt's actors, and a category theoretic approach to equations.

Appendix A: A Problem in Implementing Noncopying Reduction

Example 17 (p. 54) ignores a subtle but crucial problem in implementing noncopying reduction with pointer structures. Consider the rule schema [car(cons(A,B))=A]. This schema might be applied to a pointer structure such as that of Figure 34, representing car(cons(eval(ATOM,NIL),ATOM)).

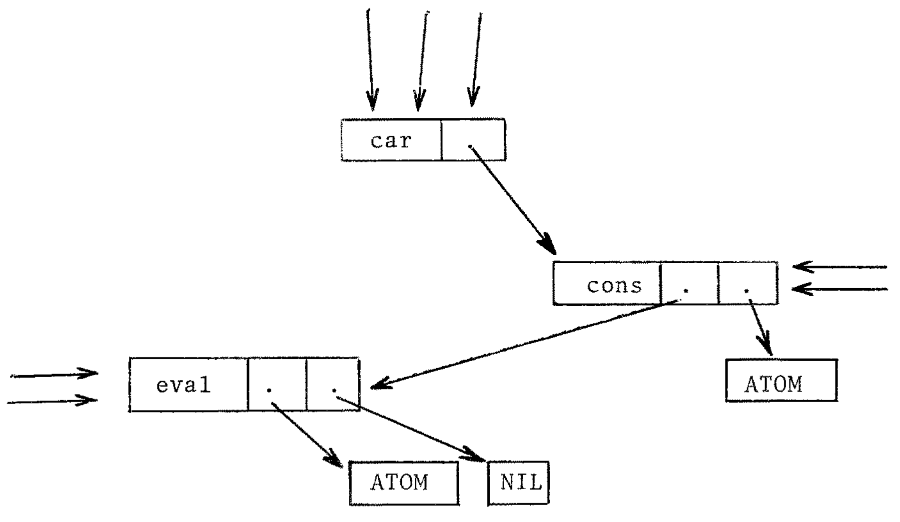


Figure 34

Notice that multiple pointers refer to the nodes | car | . | , | cons | . | . | and | eval | . | . | . A representation of the reduced expression eval(ATOM,NIL) must preserve the meanings of all pointers into this structure. Two natural representations of the reduced expression are shown in Figure 35.

(1)

(2)

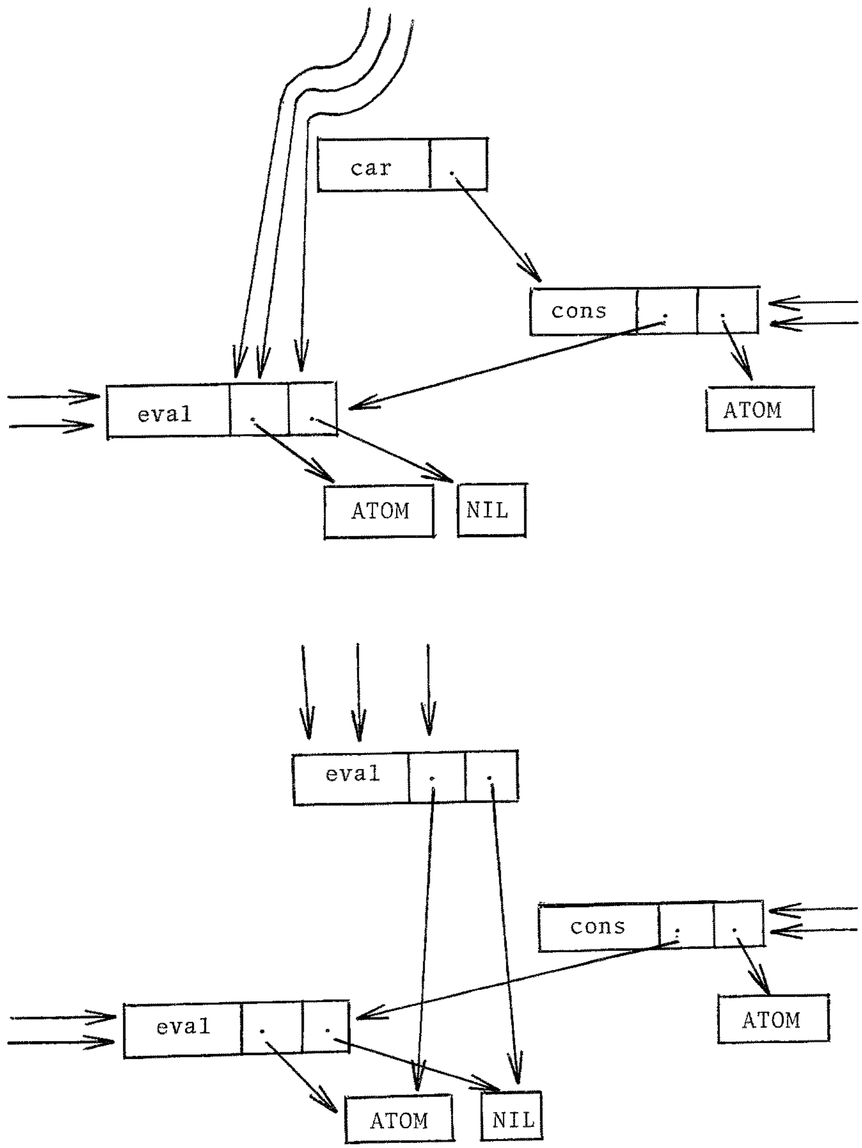


Figure 35

(1) requires a potentially huge amount of effort to find all pointers to | car | . | , some of which might prove irrelevant to the remainder of the reduction sequence. (2) is incorrect, since it creates an extra copy of eval(ATOM,NIL) where there should be only one. A similar problem occurs whenever a rule schema has a single variable symbol on the right-hand side.

Using a new type of node, indirect, with one pointer, we may reduce Figure 34 to the configuration of Figure 36.

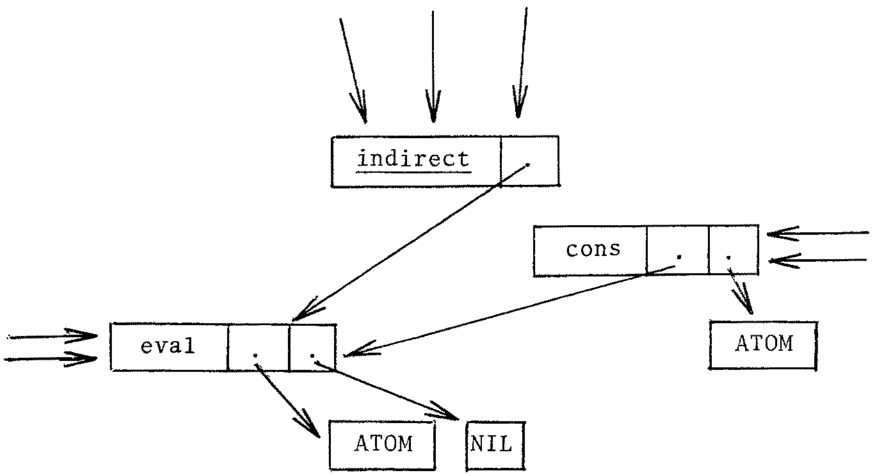
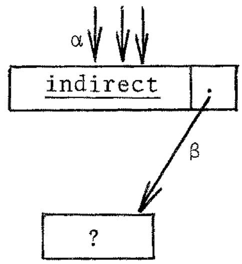


Figure 36

The indirect node is not significant to the meaning of the structure. Whenever the implementation follows a pointer α to a node | indirect | β | , α is changed to point to the node referenced by β as in Figure 37.



is changed to

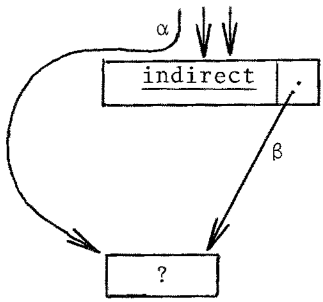


Figure 37

If and when all of the pointers to an indirect node have been followed, the node will be free for garbage collection.

Henderson and Morris [HM76] (first full paragraph, p. 99) give another solution to this problem for outermost noncopying reduction in LISP, but their method does not extend to arbitrary schemata-generated SRSs or to other reduction sequences. Ehrig and Rosen present a formal analysis of indirection [ER77,2] (Section 5).