Effective Reduction and Conversion Strategies for Combinators

by

Rick Statman
Department of Mathematical Sciences
Carnegie Mellon University
Pittsburgh, PA 15213

Research Report No. 96-195 2
October, 1996

Effective Reduction and Conversion Strategies for Combinators

by
Rick Statman
Carnegie Mellon University
October 1996
NSF CCR-9624681

Abstract:

We shall prove the following results concerning effective reduction and conversion strategies for combinators

(1)There is an effective one-step cofinal reduction strategy (answering a question of Barendregt [2] 13.6.6).

(2)There is no effective confluence function but there is an effective one-step confluence strategy (answering a question of Isles reported in [1]).

(3)There is an effective one-step enumeration strategy (answering an obvious question).

(4)There is an effective one-step Church-Rosser conversion strategy ("almost" answering a question of Bergstra and Klop [3])

1.Preliminaries

We work in the set Γ of applicative combinations (below called combinators) of S and K.

'=', '->>'. '<->', and '->' denote resp. conversion, reduction, one-step conversion, and one-step reduction, in all cases (weak) beta, of combinators. In [2] Barendregt defines a reduction strategy as a map $\phi : \Gamma \to \Gamma$ such that, for all combinators M, M ->> ϕ(M). $\phi : \Gamma \to \Gamma$ is a conversion strategy if , for all combinators M, M = ϕ(M). In each case, we say ϕ is one-step if we have M -> ϕ(M) resp. M <-> ϕ(M), and ϕ is effective if ϕ is a total recursive function, after encoding.

Although we shall state our principal results for combinations of S and K ,they remain true for other choices of bases. Our main results do not automatically carry over to lambda calculus; this is because we shall make heavy use of the principle

Residuals of redexes are disjoint .

Below $\equiv$ is used for syntactic identity.

We define the depth, d(M), of a combinator M recursively as follows:

d(S) = d(K) = 1

d(MN) = 1+ Max{d(M),d(N)}.

We shall assume that the combinators are linearly ordered by -< so that d(M) < d(N) => M -< N. Let **D** be the digraph of the one step reduction relation, and **D**(m) the subdigraph induced by all combinators of depth $\leq$ m. Let **D**(M) be the weakly connected component of **D**(d(M)) containing M (weakly connected = connected in the undirected sense). Similarly,
define **D***(M) to be the weakly connected component containing M of the subdigraph of **D**
induced by all combinators -≤ M In addition, let μ(M) be the -< least element of **D**(M). If σ is a reduction from M to N we write σ : M ->> N. We also consider pairs (M,*F*) where *F* is a set of disjoint (non-overlapping) redexes of M and we write *F*/σ for the set of residuals of *F* under σ. *F*/σ is also a set of disjoint redexes and we write σ : (M,*F*) ->> (N,*F*/σ) to show the action of σ on *F* . We shall adopt for the most part the notations of [2], especially those of chapters 6 and 12. (adapted to combinations of S and K). In particular, n̲ is the combinatory integer representing n, and the combinatory integer representing the Godel number of X is 'X'.
[x] is the usual abstraction (of x) algorithm for combinators.

2.An Effective One-step Cofinal Reduction Strategy

We say that the pair (N,*F*) is in **D**(M) if N belongs to **D**(M). The pair (N,*F*) in **D**(M) is said
to be active for M if there is a σ contained in **D**(M) such that σ : (N,*F*) ->> (M,*F*/σ) but there
is no reduction τ contained in **D**(M) with τ : (N,*F*) ->> (M,ϕ). A σ of this sort, with *F*/σ as small as possible, is said to be minimal witness to the activity of (N,*F*).
Remark: If there is no pair (N,*F*) active for M then M is recurrent ([4]). This is because if
for each disjoint set of M redexes *F* there is a reduction τ : (M,*F*) ->> (M,ϕ) then by induction M ->> N => N ->> M.

The pairs (N,*F*) can be ordered in type ω* (the type of the non-positive integers). We
refer to the non-positive integer corresponding to (N*F*). as its priority.

We shall now describe an algorithm for one-step reduction which will generate a cofinal reduction sequence.

The Algorithm 𝔸

Input; a combinator M

Output; a combinator 𝔸(M) such that M -> 𝔸(M) unless M is normal in which case 𝔸(M) ≡ M.

Begin;

(1) Decide whether there is a pair in **D**(M) active for M.

(2) If the answer to (1) is yes then find a pair (N,*F*) of highest priority in **D**(M) which is
active for M and a minimal witness σ to the activity of (N,*F*) contained in **D**(M). If

the
answer to (1) is no then go to (4).
(3)If F/σ is a singleton [R} then set 𝔸(M) ≡ Cpl(M,{R}) (see [2] pg 292). Otherwise select R
in F/σ so that d(M) ≤ d(Cpl(M,{R})) (such R exists since residuals of redexes are disjoint)
and set 𝔸(M) ≡ Cpl(M,{R}).
(4)If M is normal then set 𝔸(M) ≡ M else set 𝔸(M) ≡ any immediate reduct of M.
End.

Proposition: 𝔸 is an effective one-step cofinal reduction strategy.

Proof: We must show cofinality. Consider the itertions M -> 𝔸(M) -> 𝔸^2(M) ->... of the 𝔸 on M.If this reduction sequence does not leave **D**(M) then it must cycle ; in other words it contains a segment of the form N -> 𝔸(N) -> ... -> 𝔸^n(N) -> N. Let k be the largest ≤ n so that d(𝔸^k(N)) = max { d(𝔸^i(N)) : i = 0,1,...,n } and set N(i) ≡ 𝔸^k+i (mod n) (N). Thus the reduction sequence N(0) -> N(1) ->...-> N(n) -> N(0) is contained in **D**(N(0)). By our previous remark, if instruction (4) in 𝔸 is executed for any of the transitions N(i) -> N(i+1) then the coresponding N(i) is recurrent and the sequence of iterations of 𝔸 on M is certainly cofinal. Otherwise, for each i = 0,1,...,n, we can find a pair (P(i),F (i)) in **D**(N(i)) and a σ(i) : (P(i), F (i)) ->> (N(i),F (i)/σ(i)) contained in **D**(N(i)), obtained in the execution of instrucxtion (2) in 𝔸 on N(i). We make the following observations.

(i) d(N(0)) = d(N(i)) and for the reduction τ(i) ≡ N(i) -> N(i+1) -> ...-> N(0) we have (F (i)/σ(i))/τ(i) is not empty.

This is proved by induction on i. For i = 0, F (0)/σ(0) has residuals in N(0) since (P(0), F (0)) was active for N(0). For i > 0, since F (i-1)/σ(i-1) has residuals in N(0) we have d(N(i)) ≥ d(N(i-1)). In particular d(N(i)) = d(N(i-1)) = ... = d(N(0)). If F (i)/σ(i) has no residuals in N(0) then the reduction σ(i) -> N(i+1) -> ... -> N(0) -> ... -> N(i) is contained in **D**(N(i)) and contradicts the activity of (P(i),F (i)).

(ii) For each i = 0,1,...,n, (P(i), F (i)) ≡ (P(0),F (0)).

For let k be smallest so that (P(k),F (k)) and (P(0),F (0)) are distinct. Then (P(k),F

(k))
has higher priority than (P(0),F (0)) since (P(0),F (0)) is active for N(k). But then (P(k),F (k))
cannot be active for N(0) and there is a reduction τ : (P(k), F (k)) ->> (N(0),ϕ) contained in
D(N(0)) witnessing the inactivity of (P(k),F (k)). But then the reduction τ -> N(1) -> N(2) ->
... -> N(k) is contained in **D**(N(k)) contradicting the activity of (P(k),F (k)) for N(k).

(iii) For each i = 0,1,...,n-1, | F (i+1)/σ(i+1) | < | F (i)/σ(i) |.

This is easily seen.

Thus the reduction σ(n) -> N(0) leaves fewer residuals of F (0) in N(0) than σ(0) does and
this contradicts the minimality of σ(0). Thus we conclude that the sequence of iterations of
$\mathbb{A}$ on M is cofinal by cycling in **D**(M), or it exits from **D**(M) at some stage $\mathbb{A}$^m(M) with d($\mathbb{A}$^m(M)) > d(M).

Now let us set M(i) $\equiv$ $\mathbb{A}$^i(M) and suppose that the sequence M(0) -> M(1) -> ... M(m) -> ...
never cycles. Say that M(i) is well situated if M(0) -> M(1) -> ... ->M(i-1) is contained in **D**(M(i)). The subsequence of well situated M(i) is infinite.

(iv) Any given (P,F) is active for at most finitely many well situated M(i).

We prove this by induction on | priority(P,F) |. Suppose that this is true for all pair of
priority higher than (P,F). Find a well situated M(i) so that all pairs of higher priority are
inactive for any well situated combinator past M(i) If (P,F) is active for the next well situated combinator M(j), then it is of the highest priority and the number of residuals in
F is reduced by the transition M(j) -> M(j+1). Moreover, d(M(j)) $\leq$ d(M(j+1)) if any residuals
reman after the transition and M(j+1) is the next well situated combinator after M(j). This
remark can be repeated for M(j+1), M(j+2),...,etc. and thus (P,F) will be inactive for any well
situated combinator past M(j+ the number of residuals of F in M(j)).

We can now prove cofinality.

(v) Suppose that σ : M ->> N then, for some m, N ->> M(m).

This is proved by induction on σ. Suppose that M ->> P -> N. By induction hypothesis there
exists m such that P ->> M(m). Suppose that

Δ

P -> N

and let F be the residuals of Δ in M(m). By (v) we can find a well situated M(k) with k > m such that (M(m),F) is not active for M(k). Thus by Barendregt's strip lemma N ->> M(k), and we are done.

This completes the proof.

3.Confluence Functions and Strategies

A map φ : Γ x Γ -> Γ is said to be a confluence function if whenever M = N we have M ->> φ(M,N) <<- N. Ψ : Γ x Γ -> Γ x Γ is a one-step confluence strategy if whenever (P,Q) ≡ Ψ(M,N) we have either M -> P and N ≡ Q or M ≡ P and N -> Q, and, whenever we have M = N we have, for some m, Ψ^m(M,N) has the form (P,P). In 1980 David Isles asked whether there is a effective confluence function.

Proposition: There is no effective confluence function but there is an effective one-step confluence strategy.

Remark: the first part of this was proved by us in 1987 but not published

Proof: First suppose that φ is an effective confluence function. Recall the following definitions;

Ω ≡ [x](xx) [x](xx) , and Θ ≡ [xy](y(xxy)) [xy](y(xxy)). For each natural number e we can construct a combinator P(e) such that

P(e) = { n̲ if 0 appears for the first time in the enumeration of the eth RE set at stage n; a term with no head normal form otherwise.

This can be done in the usual way. Set M(e) ≡ P(e)Ω (ΘΩ) and N ≡ ΘΩ. The following facts are easy to verify.

(1) It is decidable whether for a given combinator P we have N ->> P.

(2) If N ->> P the for some n we have P ->> Ω^n(ΘΩ).

(3) If both P(e) = n̲ and M(e) ->> Ω^m(ΘΩ) then n ≤ m.

For (2) Klop's theorem can be used to show the cofinality of the reduction sequence ... ->> Ω^n (ΘΩ) ->>..., and for (3) routine underlining and standardization suffice. Now consider the following algorithm.

algorithm 𝔻

Input: a natural number e

Output: either "yes" or "no".

Begin;

(1) Compute φ(M(e),N).

(2) If N ->> φ(M(e),N) then go to (3) else return "no"

(3) Find n such that ϕ(M(e),N) ->> Ω^n($\Theta\Omega$).
(4) Compute n steps in the enumeration of the eth RE set.
(5) If 0 is found by the nth step then return "yes" else return "no".
End.

We claim 0 belongs to the eth RE set <=> $\mathbb{D}$(e) = "yes". The direction <= is obvious. The direction => follows from (3) above. Thus we have a contradiction and ϕ cannot exist.

On the otherhand we can construct a confluence strategy easily from $\mathbb{A}$. Define an algorithm $\mathbb{B}$
as follows.

Input; a pair (M,N) of combinators
Output; a pair (P,Q) of combinators
Begin;
(1) If there is a reduction M -> N1 -> N2 ->.....-> Nn $\equiv$ N such that for i = 1,2,...,n-1 Ni -< N
pick one which is lexicographically least and set P $\equiv$ N1 and Q $\equiv$ N else
(2) Put P $\equiv$ M and Q $\equiv$ $\mathbb{A}$(N).

It is easy to see that $\mathbb{B}$ has the desired properties. This completes the proof.

4.Enumeration Strategies

An effective one-step enumeration strategy is a total recursive function $\chi : \Gamma \to \Gamma$ such that for each M we have M <-> χ(M) and there exists an N = M such that the sequence
N, χ(N), $\chi(\chi$(N)),...,χ^n(N),... exhausts all the combinators convertible to M. Obviously, if such
a χ exists then N does not depend on M, but only on the weak beta convertibility class of M.

We say that M is special if it has the form KP('**D***(Q)') . We really do not care much about the details of the Godel numbering except for the fact that coding and decoding should be effective, that" special " should be distinct from the case KP('N') for any N, and that P -< 'P', '**D***(P)'.

We shall now describe an algorithm for one-step conversion which yields an enumeration strategy. The algorithm is defined by recursion on -<

The Algorithm $\mathbb{E}$
Input; a combinator M
Output; a combinator $\mathbb{E}$(M) such that $\mathbb{E}$(M) <-> M.
Begin;
(1)Determine whether M is special. If it is go to (5) else to (2)
(2)For each element N of **D***(M), besides M, compute $\mathbb{E}$(N). If $\mathbb{E}$ fails to be injective then exit.
Otherwise each such element N lies in a maximal chain of $\mathbb{E}$ iterates beginning with

some
v(N); make a list of these v(N).
(3)If there is some N in **D***(M) with v(N) >- v(M) then set **E**(M) ≡ KM('**D***(M)') else go on to (4)
(4) Set **E**(M) ≡ KM('M')
(5)If M ≡ KP('**D***(Q)') execute **E** on Q and determine whether **E**(Q) is set by executing (*).
If the answer is no go to (2) for M else find a -< smallest v(N) >- v(Q) in **D***(Q), and a
lexicographically -< shortest path **R** in **D***(Q) from Q to v(N).
If P lies on **R** then,

if P ≡ v(N) then
(*) set **E**(M) ≡ P
else
(**) set **E**(M) ≡ K(the next element of **R** after P)('**D***(Q)')

else if P fails to lie on **R** then go to (2) for M.

End.

Proposition: **E** is an effective one-step enumeration strategy.

Proof: First we prove by induction on -< that **E** is total and defines a 1-1 function. If either of
these fails for M , by induction hypothesis, the failure must be in 1-1 ness. This can only happen, by choice of Godel numbering, when there are M1, M2 -≤ M, M ε {M1,M2}, **E**(M1) ≡ P ≡ **E**(M2), and **E**(M1) is set by (*). We distinguish two cases.

Case1; **E**(M2) is set by (*).

In case M1 -< M2 we have that v(P) ≡ v(M1) -< P after **E**(M1) is defined ,so (*) cannot be
executed w.r.t. P for M2 to set **E**. Similarly for the case that M2 -< M1.

Case 2; **E**(M2) is not set by (*).

We have M1 ≡ KP('**D***(Q)') and P ≡ KUV for some U and V. Indeed either M2 ≡ U or there exist L, T such that V ≡ '**D***(T)' , L-< T -< V , and M2 ≡ KLV -< M1. However, if M2 -< M1 then P cannot be a v(N) for M1 and **E**(M1) is not set by (*). This completes the proof of totality and 1-1 ness.

Now suppose that N is the -< least element in the convertibility equivalence class of M. Let Q be any combinator convertible to N but not in the sequence N, **E**(N), **E**(**E**(N)),...,**E**^n(N),....and let P be the -< least element such that Q ≡ **E**^m(P) for some m
Since **E** is 1-1, N is not in the range of **E**. Thus the above sequence is infinite. Say that an element L of this sequence is well situated if we have N, **E**(N),..., **E**^(n-1)(N) -<

$\mathbb{E}$^n(N) ≡ L
and $\mathbb{E}$(L) is not set by (5). The $\mathbb{E}$ iterate of N which first has depth > n is well situated thus
the sequence of well situated combinators is infinite. Let P <-> P1<-> ...<-> Pp <-> N be any conversion from P to N. There are infinitely many well situated L such that P,P1,...,Pp, $\mathbb{E}$(P), $\mathbb{E}$^2(P),..., $\mathbb{E}$^m(P) -< L. Pick one so that no U -< P appears after it in the iterates of $\mathbb{E}$ on N.
Now P cannot be in the range of $\mathbb{E}$ for if $\mathbb{E}$(T) ≡ P we have T ≡ KU('**D***(V)') and $\mathbb{E}$(T) is set by
(5). If $\mathbb{E}$(T) is set by (*) then ν(V) -< P and P ε { $\mathbb{E}$^k(V) : k = 0,1,2,... } contradicting the choice
of P. Similarly if $\mathbb{E}$(T) is set by (**) then V -< P and P ε { $\mathbb{E}$^k(V) : k = 0,1,2,... } again contra-
dicting the choice of P. Thus when $\mathbb{E}$(L) is computed it is set by (3) to KL('**D***(L)') which,
after some number of iterations of $\mathbb{E}$ set by (**), is set to P by (*). This contradicts the choice
of P and Q.This completes the proof.

5.Church-Rosser Strategies

In [3] Bergstra and Klop asked whether there is an effective one-step Church-Rossser reduction strategy. This question remains unanswered. However, here we will construct an effective one-step conversion strategy which is Church-Rosser. This is, we will construct a total recursive function ψ such that
(1)For any M, M <-> ψ(M)
(2)For any M = N there exist m and n such that M ->> ψ^m(M) ≡ ψ^n(N) <<- N
by defining an algorithm $\mathbb{C}$ below.

We need to recall from 1. some of the properties of the algorithm $\mathbb{A}$. Either there are
infinitely many m such that M -> $\mathbb{A}$(M) -> ... -> $\mathbb{A}$^(m-1)(M) is contained in **D**($\mathbb{A}$^m(M)) or there is a single m such that for any n > m we have $\mathbb{A}$^n(M) belongs to **D**($\mathbb{A}$^m(M)). In the first case we call these $\mathbb{A}$^m(M) the well situated reducts of M and in the second case we call $\mathbb{A}$^m(M) a sink for M. If there is a sink in **D**(M) we let ν(M) be the -< least such sink. Given a finite reduction sequence *R* = M1 -> M2 -> ... -> Mm we make the following definitions; lh(*R*) = m , df(R) = Σ i=1,...,n max{ d(M(i+1)) - d(Mi), 0}, wk(*R*) =
| {Mi : μ(Mi) >- μ(M1) } |. Now we order the triples trip(*R*) = (df(*R*),wk(*R*),lh(*R*)) lexico-
graphically and observe that among all the reduction sequences from M1 to Mm there are only finitely many paths *P* with df(*P*) < k for any fixed k. This is because any

term
in such a path has depth at most d(M1) + k. We shall assume that all of these paths have
been well ordered by << so that trip(P 1) < trip(P 2) => P 1 << P 2. Given M let ρ(M) be the
<< least reduction path from M to a well situated reduct of μ(M) or a sink of μ(M) which ever
exists. Also let γ(M) be the << least reduction path from M to ν(M) if this term exists. It should be clear that ρ(M) and γ(M) can be effectively constructed from M using 𝔸. Now
let
A1 ≡ S(KK)(S(SKK)(SKK))(S(KK)(S(SKK)(SKK))) (this is just a combinatory fixed point of K)
A2 ≡ KK(S(KK)(S(SKK)(SKK))(A1).
We now define the algorithm ℂ.
Input; a combinator M
Output; a combinator ℂ(M) such that M <-> ℂ(M)
Begin;
(1) If ν(M) exists then
if M ≡ K(ν(M))(K^n(A1)) then ℂ(M) ≡ K(ν(M))(K^n(A2)) else
if M ≡ K(ν(M))(K^n(A2)) then ℂ(M) ≡ K(ν(M))(K^(n+1)(A1)) else
if M ≡ ν(M) then ℂ(M) ≡ KM(A1) else
set M+ ≡ the next point on γ(M) and if ν(M+) exists and is -< ν(M)
then ℂ(M) ≡ M+ else ℂ(M) ≡ KM('M')
else go to (2)
(2) If M ≡ K(𝔸^n(N)('N') where N is a well situated reduct of μ(M) and none of the 𝔸^j(N)
for j = 1,...,n are well situated reducts of μ(M) then ℂ(M) ≡ K(𝔸^(n+1)(N))('N') else go to (3).
(3) If lh(ρ(M)) = 1 then
if 𝔸(M) is a well situated reduct of μ(M) then ℂ(M) ≡ 𝔸(M)
else ℂ(M) ≡ KM)'M')
else let M+ be the next point on ρ(M) and go to (4)
(4) If μ(M+) -≤ μ(M) then
if M ≡ K(M+)('M+') then ℂ(M) ≡ K(𝔸(M+))('M+')
else ℂ(M) ≡ M+
else ℂ(M) ≡ KM('M')
End.
Proposition: ℂ is an effective one-step Church-Rosser conversion strategy

Proof: First consider the sequence of iterations of ℂ on a combinator M ; viz, M , ℂ(M) , ℂ(ℂ(M)) , ... , ℂ^m (M) , ... We claim that this sequence is unbounded in depth.

Indeed if ν(N) is defined for any N ≡ ℂ^n(M) then let N be such a member of the sequence with ν(N) -< smallest. We distinguish two cases

(i) N is ν(N), K(ν(N))(K^m(A1)), or K(ν(N))(K^m(A2))

Then ℂ(N) ≡ K(ν(N))(A1), K(ν(N))(K^m(A2)), or K(ν(N))(K^(m+1)(A1)) and ν(ℂ(N)) ≡ ν(N).

(ii) Otherwise

Then ν(ℂ(N)) ≡ ν(N) and γ(ℂ(N)) << γ(N)

Thus the first case eventually comes up and once it is established it persists forever. Otherwise, ν(ℂ^m(M)) is never defined. Now, let P ≡ ℂ^p(M) be such that μ(P) is -< smallest and from among these such that ρ(P) is << smallest. We claim that some well situated reduct of μ(P) is in the original iterative sequence. Let P1 ≡ P and ρ(P) ≡ P1 -> P2 -> ... -> Pk.If for any i > 1 we have μ(P(i+1)) >- μ(P1) then for a smallest such i we have ℂ(P1) ≡ P2 , ℂ(P2) ≡ P3, ... , ℂ(P(i-1)) ≡ Pi, ℂ(Pi) ≡ K(Pi)('Pi'), and K(Pi)('Pi') -> K(P(i+1))('Pi') -> ...->K(Pk)('Pi') -> Pk. Thus ρ(ℂ(Pi)) << ρ(P1) contradicting the choice of P1. Thus for j = 1,...,k-2 we have ℂ(Pj) ≡ P(j+1). We distinguish two cases here.

(a) ℂ(P(k-1)) ≡ Pk

Then the well situated reduct of μ(P) in the original iterative sequence is Pk.

(b) Otherwise.

Then we have ℂ(P(k-1)) ≡ K(P(k-1))('P(k-1)'). In this case the next well situated reduct of μ(P) is in the original iterative sequence. For if this reduct is 𝔸^r(Pk) we have ℂ(P(k-1)) ≡ K(P(k-1))('P(k-1)'), ℂ^2(P(k-1)) ≡K(Pk)('P(k-1)'), ..., ℂ^(r+2)(P(k-1)) ≡ 𝔸^r(Pk).

Once a well situated reduct of μ(P) is found in the original iterative sequence , the sequence of interates alternates between instruction (2) and instruction (3) forever. Since ν(N) cannot exist the sequence must grow unbounded in depth and this proves the claim. Say that ℂ^m(M) is a well situated convert of M if the elements M , ℂ(M) , ... , ℂ^(m-1)(M) belong to **D**(ℂ^m(M)).

Since the sequence of iterations grows unbounded in depth there are infinitely many well situated converts of M. Let the well situated converts of M be N1,N2,...,Nn,....Now if there is a sink for M then there is some Nn such that that sink belongs to **D**(Nn) and hence ν(Nn) exists for all but finitely many n. Thus for all but finitely many n, ν(ℂ^n(M)) exists and is the -< least sink beta convertible to M. Similarly, for all but finitely many n, μ(ℂ^n(M)) is the -< least combinator beta convertible to M. Finally if there is some sink for M then for all but finitely many n, ℂ^n(M) alternates between KP(K^m(A1)) and KP(K^m(A2)) where P is the -< least such sink and if there is no such sink then for all but finitely many n, ℂ^n(M) alternates between the well situated reducts P of the -< least combinator beta convertible to M and the terms K(𝔸^m(P))('P') for all but finitely many such P. It follows that ℂ is a Church-Rosser strategy.

This completes the proof.

6.References

[1] Open Problems — Bull. Euro. Assoc. Theor. Comp. Sci. 10 (1980) pps 136-140

[2] Barendregt — The Lambda Calculus, North Holland ,1984

[3] Bergstra & Klop — Church-Rosser strategies in the lambda calculus, Preprint 62 , Dept. of Math., Univ. of Utrecht, July 1977 (appeared in TCS 9 1979 pps 27-38)

[4] Jacopini & Venturini-Zilli — Equating for recurrent terms of lambda calculus, Pubblicazioni dell' Istituto per le Applicazioni del Calcolo Ser.III 85, CNR, Rome, 1978