From finite lambda calculus to infinite lambda calculi*

Richard Kennaway[1], Jan Willem Klop[2], Ronan Sleep[3] and Fer-Jan de Vries[4]

[1,3] School of Information Systems
University of East Anglia, Norwich NR4 7TJ, UK

[2] Department of Software Technology, CWI
P.O. Box 4079, 1009 AB Amsterdam, the Netherlands

[4] NTT, Communication Sciences Laboratories
Hikaridai,Seika-cho, Soraku-gun,Kyoto619-02, Japan
ferjan@progn.kecl.ntt.jp

13 October 1994

Abstract

In a previous paper we have established the theory of transfinite reduction for orthogonal term rewriting systems. In this paper we perform the same task for the lambda calculus. There happen to be several candidates depending on the metric completion of the finite lambda calculus. The infinite Church-Rosser property does not hold for them, however the slightly weaker normal form property does hold.

要旨

以前の論文で，我々は直交項書き換え系の超限的な簡約の理論を提示した．本論文では，同様のことをラムダ計算に対して行なう．その際，有限ラムダ計算から如何に距離に基づく完備化をおこない無限ラムダ計算を得るかは幾通りかの方法がある．その何れも，Church-Rosser 性は持たないが，やや弱い正規形性をもつことを示す．

Keywords
Transfinite rewriting, lambda calculus, functional language,
Church-Rosser property, normal form property, Böhm tree.

*This paper is an extended abstract of a longer manuscript with the title *The infinite lambda calculus* Λ^{∞}. We have omitted most of the proofs

1 Introduction

In [KKSdV] we used the notion of strongly convergent sequences of length greater than ω to build a theory of transfinite reduction for orthogonal term rewrite systems. In this paper we perform the same task for the lambda calculus. In contrast to term rewriting there happen to be several different notions of depth from which (non-discrete) metrics can be constructed to make the set of finite lambda terms into a metric space. Each depth brings its own completion. And each completion brings its own concepts of (strongly converging) reductions. In this abstract we will concentrate on the infinite lambda calculus related to applicative depth. In tree representation the infinite terms of the related completion $\Lambda^{\infty a}$ a re infinite lambda trees in which on infinite branches eventually all nodes are right sons of their father.

These infinite lambda calculi differ from orthogonal infinitary term rewriting in the following: the parallel moves lemma does not generalise to strongly converging reductions. Hence the infinitary Church-Rosser property fails for them.

However it is possible to prove the infinite Church Rosser property for Böhm reduction along the lines of [AKK+94]. (Böhm reduction is ordinary reduction extended with the option to rewrite a subterm without head normal form to $\perp$, a symbol denoting undefinedness.) As a corollary we obtain—instead of the infinite Church-Rosser property—the infinite normal form property NF^{∞} for $\Lambda^{\infty a}$.

Apart from its theoretical interest, infinite lambda calculi arise from the use of lambda calculus in functional programming. One can write expressions whose normal form is, intuitively speaking, an infinite term - a list of Fibonacci numbers for instance. Such infinite normal forms can be viewed as the limits of infinite reduction sequences. Infinite terms and reductions also arise in the correspondence between lambda graph rewriting and lambda calculus. Lambda graph rewriting extends lambda calculus with sharing and is an important implementation technique for functional languages. Such implementations might use cyclic graphs in order to make certain optimisations. Cyclic graphs correspond to certain infinite lambda terms; rewriting cyclic lambda terms corresponds to infinite computations on lambda terms. A study of the soundness of such implementation techniques requires a study of infinite lambda calculus.

2 Basic definitions

2.1 Finitary lambda calculus

We assume familiarity with the lambda calculus, or as we shall refer to it here, the finitary lambda calculus. [Bar84] is a standard reference. The syntax is simple: there is a set Var of variables; an expression or term E is either a variable, an abstraction $\lambda x.E$ (where x is called the bound variable and E the body), or an application E_1E_2 (where E_1 is called the rator and E_2 the rand). This is the pure lambda calculus — we do not have any built-in constants nor any type system.

As customary, we identify α-equivalent terms with each other, and consider bound variables to be silently renamed when necessary to avoid name clashes.

The following particular terms will be frequently used.

DEFINITION 2.1

$$\begin{aligned} I &= \lambda x.x \\ K &= \lambda x.\lambda y.x \\ Y &= \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)) \\ Y_T &= (\lambda x.\lambda f.f(xxf))(\lambda x.\lambda f.f(xxf)) \end{aligned}$$

Y_T (called Turing's fixed point operator) has the property that $Y_Tf \to^* f(Y_Tf)$, which makes it sometimes more convenient than the more usual fixed point operator Y, for which we only have that for any f, Yf reduces (in one step) to a term Y_f having the property that $Y_f \to^* f(Y_ff)$.

2.2 What is an infinite term?

Drawing lambda expressions as syntax trees gives an immediate and intuitive notion of infinite terms: they are just infinite trees. Formally, we can define this set as the metric completion of the space of finite trees with a well-known (ultra-)metric. The larger the common prefix of two trees, the more similar they are, and the closer together they may be considered to be. First, some terminology. A *position* or *occurrence* is a finite string of positive integers. Given a term M and a position u, the term $M|u$, when it exists, is a subterm of M defined inductively thus:

$$\begin{aligned} M|\langle\rangle &= M \\ (\lambda x.M)|1\cdot u &= M|u \\ (MN)|1\cdot u &= M|u \\ (MN)|2\cdot u &= N|u \end{aligned}$$

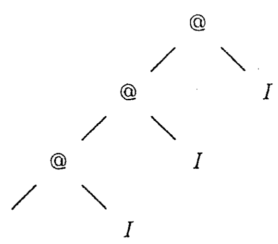

Figure 1:

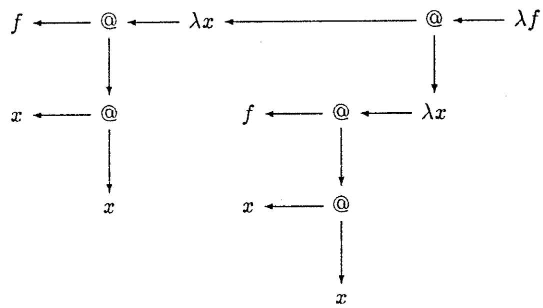

Figure 2:

$M|u$ is called the subterm of M at u, and when this is defined, u is called a position of M. The *depth* of u is its length.

Two positions u and v are *disjoint* if neither neither position is a prefix of the other. A set of positions or redexes is disjoint if every two distinct members are.

Given two distinct terms M and N, let d be the length of the shortest position u such that $M|u$ and $N|u$ are both defined, and are either of different syntactic types or are distinct variables. Then the larger d is, the more similar are M and N. The distance between M and N is defined to be 2^{-d}. Denote this measure by $d^s(M,N)$. (The superscript s will be explained later.) $d^s(M,M)$ is defined to be 0. It is easily proved that this is a metric on the set of finite terms. In fact, it is an ultrametric, i.e. $d^s(M,N) \leq max(d^s(M,P), d^s(P,N))$, although this fact will not be important. The completion of this metric space adds the infinite terms. We call this set $\Lambda^{\infty s}$.

The above is the definition of infinite terms which we used in our study of transfinite term rewriting, but for lambda calculus the situation is a little more complicated. Certain members of the class defined above have problematic properties, and we will be excluding them from our study. Consider the term $(((\ldots I)I)I)I$, illustrated in Figure 1. This term has a combination of properties which is rather strange from the point of view of finitary lambda calculus. By the usual definition of head normal form — being of the form $\lambda x_1 \ldots \lambda x_n.yt_1 \ldots t_m$ — it is not in head normal form. By an alternative formulation, trivially equivalent in the finitary case, it is in head normal form — it has no head redex. It is also a normal form, yet it is unsolvable (that is, there are no terms $N_1, \ldots, N_n$ such that $MN_1 \ldots N_n$ reduces to $\lambda x.x$). The problem is that application is strict in its first argument, and so an infinitely left-branching chain of applications has no obvious meaning. We can say much the same for an infinite chain of abstractions $\lambda x_1.\lambda x_2.\lambda x_3.\ldots$.

Another reason for wishing to exclude terms with infinitely left-branching chains of applications arises from analogy with term rewriting. In a term such as $F(x,y,z)$, the function symbol F is at depth 0. If it is curried, that is, represented as $Fxyz$, or explicitly $@(@(@(F,x),y),z)$ (as it would be if we were to translate the term rewrite system into lambda calculus), the symbol F now occurs at depth 3. We could instead consider it to be at depth zero; more generally, we may adopt a new measure of depth which deems the left argument of an application to be at the same depth as the application itself.

DEFINITION 2.2 Given a term M and a position u of M, the *applicative depth* of the subterm of M at u, if it exists, is defined by:

$$\begin{aligned}
ADepth(M, \langle\rangle) &= 0 \\
ADepth(\lambda x.M, 1\cdot u) &= ADepth(M,u) \\
ADepth(MN, 1\cdot u) &= ADepth(M,u) \\
ADepth(MN, 2\cdot u) &= 1 + ADepth(N,u)
\end{aligned}$$

The associated space of finite and infinite terms is denoted $\Lambda^{\infty a}$, and the measure of distance d^a.

The applicative depth of the subterms of the term Y is shown visually in Figure 2. Note that this is just the usual syntax tree, rotated clockwise through 45°.

There are several other notions of depth we might consider, depending on whether the depth is or is not considered to increase when passing from an abstraction to its body, or from an application to either of its arguments. These three contexts give eight different notions of depth. On of these—*discrete* depth, which considers every subterm at depth zero—generates the finitary lambda calculus in which there are no infinite terms and no strongly converging infinite reductions. The original notion of depth, which increase in all three contexts, we will call ***syntactic*** depth. In this extended abstract we shall henceforth only consider the applicative depth. The full paper will show that most of our results do not depend on which notion of depth is used.

The chosen notion of depth gives a metric on the space of finite terms, from which we obtain a complete mnetric space of finite and infinite terms and a notion of convergent sequence. We shall write d for applicative depth and Λ^∞.

2.3 What is an infinite reduction sequence?

We have spoken informally of convergent reduction sequences but not yet defined them. The obvious definition is that a reduction sequence of length ω converges if the sequence of terms converges with respect to the metric. However, this proves to be an unsatisfactory definition, for the same reasons as in [KKSdV]. There are two problems. Firstly, a certain property which is important for attaching computational meaning to reduction sequences longer than ω fails.

DEFINITION 2.3 A reduction system admitting transfinite sequences satisfies the *Compression Property* if for every reduction sequence from a term s to a term t, there is a reduction sequence from s to t of length at most ω.

A counterexample to the Compression Property is easily found in $\Lambda^{\infty s}$. Let $A_n = (\lambda x.A_{n+1})(B^n(x))$ and $B = \lambda x.y$. Then $A_0 \rightarrow^\omega C$ where $C = (\lambda x.C)$ (B^ω), and $C \rightarrow (\lambda x.C)y$. There is no reduction of A_0 to $(\lambda x.C)y$ in ω or fewer steps.

The second difficulty with this notion of convergence is that taking the limit of a sequence loses certain information about the relationship between subterms of different terms in the sequence. Consider the term I^ω of Λ^∞, and the infinite reduction sequence starting from this term which at each stage reduces the outermost redex: $I^\omega \rightarrow I^\omega \rightarrow I^\omega \rightarrow \ldots$ All the terms of this sequence are identical, so the limit is I^ω. However, each of the infinitely many redexes contained in the original term is eventually reduced, yet the limit appears to still have all of them. It is not possible to say that any redex in the limit term arises from any of the redexes in the previous terms in the sequence.

A third difficulty arises when we consider translations of term rewriting systems into the lambda calculus. Even when such a translation preserves finitary reduction, it may not preserve Cauchy convergent reduction. Consider the term rewrite rule $A(x) \rightarrow A(B(x))$. This gives a Cauchy convergent term rewrite sequence $A(C) \rightarrow A(B(C)) \rightarrow A(B(B(C)))\ldots$ If one tries to translate this by defining $A_\lambda = Y(\lambda f.\lambda x.f(Bx))$ (for some λ-term B), the resulting sequence will have an accumulation point corresponding to the term $A(B^\omega)$, but will not be Cauchy convergent. The reason is that what is a single reduction step in the term rewrite system becomes a sequence of several steps in the lambda calculus, and while the first and last terms of that sequence may be very similar, the intermediate terms are not, destroying convergence.

The remedy for all these problems is the same as in [KKSdV]: besides requiring that the sequence of terms converges, we also require that the depths of the redexes which the sequence reduces must tend to infinity.

DEFINITION 2.4 A *pre-reduction* sequence of length α is a function ϕ from an ordinal α to reduction steps of Λ^∞, and a function τ from $\alpha + 1$ to terms of Λ^∞, such that if $\phi(\beta)$ is $a \rightarrow^r b$ then $a = \tau(\beta)$ and $b = \tau(\beta+1)$. Note that in a pre-reduction sequence, there need be no relation between the term $\phi(\beta)$ and any of its predecessors when β is a limit ordinal.

A pre-reduction sequence is a *Cauchy convergent reduction sequence* if τ is continuous with respect to the usual topology on ordinals and the metric on Λ^∞.

It is a *strongly convergent reduction sequence* if it is Cauchy convergent and if, for every limit ordinal $\lambda \leq \alpha$, $\lim_{\beta\rightarrow\lambda} d_\beta = \infty$, where d_β is the depth of the redex reduces by the step $\phi(\beta)$. (The measure of depth is the one appropriate to each version of Λ^∞.)

If α is a limit ordinal, then an *open* pre-reduction sequence is defined as above, except that the domain of τ is α. If τ is continuous, the sequence is *Cauchy continuous*, and if the condition of strong convergence is satisfied at each limit ordinal less than α, it is *strongly continuous*.

When we speak of a reduction sequence, we will mean a strongly convergent reduction sequence unless otherwise stated.

COUNTEREXAMPLE 2.5 Strongly convergent reduction in Λ^∞ is not Church-Rosser. Let $D = \lambda x.xx$, $A = \lambda x.I(xx)$ Then $DA \rightarrow AA \rightarrow I(AA) \rightarrow^\omega I^\omega$. But $DA \rightarrow DD$. Both DD and I^ω reduce only to themselves, hence they have no common reduct.

Note that this is also a counterexample to a special case of the Church-Rosser property called the

Strip Lemma: one of the two sequences is only a single step.

3 Descendants and residuals

3.1 Descendants

When a reduction $M \to N$ is performed, each subterm of M gives rise to certain subterms of N — its descendants — in an intuitively obvious way. Everything works in almost exactly the same way as for finitary lambda calculus.

DEFINITION 3.1 Let u be a position of t, and let there be a redex $(\lambda x.M)N$ of t at v, reduction of which gives a term t'. The set of *descendants* of u by this reduction, u/v, is defined by cases.

- If $u \not\geq v$ then $u/v = \{u\}$.
- If $u = v$ or $u = v \cdot 1$ then $u/v = \emptyset$.
- If $u = v \cdot 2 \cdot w$ then $u/v = \{v \cdot y \cdot w \mid y \text{ is a free occurrence of } x \text{ in } M\}$. If $u = v \cdot 1 \cdot w$ then $u/v = \{v \cdot w\}$.

The *trace* of u by the reduction at v, $u /\!/ v$, is defined in the same way, except for the second case: if $u = v$ or $u = v \cdot 1$ then $u /\!/ v = \{v\}$.

For a set of positions U, $U/v = \bigcup\{u/v \mid u \in U\}$ and $U /\!/ v = \bigcup\{u /\!/ v \mid u \in U\}$.

The notions of descendant and trace can be extended to reductions of arbitrary length.

DEFINITION 3.2 Let U be a set of positions of t, and let S be a reduction sequence from t to t'. For a reduction sequence of the form $S \cdot r$ where r is a single step, $U/(S \cdot r) = (U/S) \cdot r$. If the length of S is a limit ordinal α then $U/S = \bigcup_{\beta<\alpha} \bigcap_{\beta<\gamma<\alpha} U/S_\gamma$.

$U /\!/ S$ is defined similarly.

Strong convergence of S ensures that the above limit exists.

LEMMA 3.3 *Let U be a set of positions of redexes of t, and let S be a reduction from t to t'. Then there is a redex at every member of U/S.* □

DEFINITION 3.4 The redexes at U/S in the preceding lemma are the *residuals* of the redexes at U.

DEFINITION 3.5 Let u and v be positions in the initial and final terms respectively of a sequence S. If $v \in u /\!/ S$, we also say that u *contributes* to v (via S). If there is a redex at v, then u *contributes* to that redex if u contributes to v or $v \cdot 1$.

Note that descendants, traces, residuals, and contribution are not defined for Cauchy convergent reductions, which is not surprising given the examples of section 2.3.

THEOREM 3.6 *For any strongly convergent sequence $t_0 \to^\alpha t_\alpha$ and any position u of t_α, the set of all positions of all terms in the sequence which contribute to u is finite, and the set of all reduction steps contributing to u is finite.*

3.2 Developments

DEFINITION 3.7 A *development* of a set of redexes R of a term M is a sequence in which every step reduces some residual of some member of R by the previous steps of the sequence. It is *complete* if the final term contains no residual of any member of R.

Not every set of redexes has a complete development. An example is provided by the term $I^\omega = (\lambda x.x)((\lambda x.x)((\lambda x.x)(\ldots)))$. Every attempt to reduce all the redexes in this term must give a reduction sequence containing infinitely many reduction steps at the root of the term. Note that the set consisting of every redex at odd depth has a complete development, as does the set consisting of every redex at even depth, but their union does not.

Let $M = (\lambda x.y)(I^\omega)$. Let R_1 be the set of redexes in the subterm I^ω at odd depth. Let S_1 be a complete development of R_1, followed by reduction of the outermost redex. Let R_2 be the set of redexes in the subterm I^ω at even depth. Let S_2 be a complete development of R_2, followed by reduction of the outermost redex. Then neither of the sets R_1/S_2 and R_2/S_1 have complete developments.

THEOREM 3.8 *Complete developments of the same set of redexes end at the same term.*

PROOF. In the finitary case one proves this by showing that (1) this is true for a pair of redexes, and (2) all developments are finite, and invoking Newman's Lemma.

In the infinitary case a more refined argument is required. □

4 The truncation theorem

Some results about the finitary lambda calculus can be transferred to the infinitary setting by using finite approximations to infinite terms.

DEFINITION 4.1 A $\Lambda_\bot$ term is a term of the version of lambda calculus obtained by adding $\bot$ as a new symbol. $\Lambda_\bot^\infty$ is defined from $\Lambda_\bot$ as Λ^∞ is from Λ, and similarly for the other versions of Λ^∞.

The terms of $\Lambda_\bot^\infty$ have a natural partial ordering, defined by stipulating that $\bot < t$ for all t, and that application and abstraction are monotonic.

A *truncation* of a term t is any term t' such that $t' \leq t$. We may also say that t' is weaker than t, or t is stronger than t'.

THEOREM 4.2 *Let $t_0 \to^\alpha t_\alpha$ be a reduction sequence. Let s_α be a prefix of t_α, and for $\beta < \alpha$, let s_β be the prefix of t_β contributing to t_α. Then for any term r_0 such that $s_0 \leq r_0$ there is a reduction sequence $r_0 \to^{\leq \alpha} r_\alpha$ such that:*

1. *For all β, s_β is a prefix of r_β.*
2. *If $t_\beta \to t_{\beta+1}$ is performed at position u and contributes to s_α, then $r_\beta \to r_{\beta+1}$ by reduction at u.*
3. *If $t_\beta \to t_{\beta+1}$ is performed at position u and does not contribute to s_α, then $r_\beta = r_{\beta+1}$.*

As an example of the use of this theorem, we demonstrate that Λ^∞ is conservative over the finitary calculus, for terms having finite normal forms.

COROLLARY 4.3
1. *If $t \to^\infty s$ and s' is a finite prefix of s, then t is reducible in finitely many steps to a term having s' as a prefix. In particular, if t is reducible to a finite term, it is reducible to that term in finitely many steps.*
2. *If a finite term is reducible to a finite normal form, it is reducible to that normal form in the finitary lambda calculus.* □

5 The Compressing Lemma

We have justified the interest of infinite terms and sequences by seeing them as limits of finite terms and sequences. In this light, the computational meaning of a sequence of length longer than ω may be obscure — it is difficult to imagine performing an infinite amount of work and then doing some more work. We therefore wish to be assured that every reduction sequence of length greater than ω is equivalent to one of length no more than ω, in the same of having the same initial and final term. This allows us to freely use sequences longer than ω without losing computational relevance.

THEOREM 5.1 (Compressing Lemma.) *In Λ^∞, for every strongly convergent sequence there is a strongly convergent sequence with the same endpoints whose length is at most ω.*

PROOF. The corresponding theorem of [KKSdV] shows that the case of a sequence of length $\omega + 1$ implies the whole theorem, and the proof is not dependent on the details of rewriting — it is valid for any abstract transfinite reduction system (as defined in [Ken92]). □

REMARK 5.2 The Compressing Lemma is false for $\beta\eta$reduction.

For a counterexample, let $M = Y(\lambda f.\lambda x.I(fx))$ where $Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ and $I = \lambda x.x$. Then $\lambda x.Mxx \to^\omega \lambda x.I(I(I(...)))x \to_\eta I(I(I(...)))$. However, $\lambda x.Mxx$ is not reducible in ω steps or fewer to $I(I(I(...)))$.

This is not really surprising. The η-rule requires testing for the absence of the bound variable in the body of the abstraction; if the abstraction is infinite, this is an infinite task, and such discontinuities are to be expected.

6 Head normal forms and Böhm trees

DEFINITION 6.1 A *head normal form* is a term of the form $\lambda x_1 \ldots \lambda x_n.yM_1 \ldots M_k$, where y may be any of $x_1, \ldots, x_n$ or any other variable.

A term *has* a head normal form if it can be reduced to one.

A term not in head normal form having a finite left spine must have the form $\lambda x_1 \ldots \lambda x_n.(\lambda y.M)N$ $M_1 \ldots M_k$. The redex $(\lambda y.M)N$ is the *head redex* of the term.

A term M is *solvable* if there are terms $N_1, \ldots, N_n$ such that $MN_1 \ldots N_n \to^\infty I$.

In $\Lambda_\bot^\infty$, the same definitions apply. This means that $\bot$ is considered to be not a head normal form, and is unsolvable, as are terms of the form $\bot\ M$ and $\lambda x.\bot$.

DEFINITION 6.2 The head normal forms $\lambda x_1 \ldots \lambda x_n.yM_1 \ldots M_k$, and $\lambda x'_1 \ldots \lambda x'_n.y'M'_1 \ldots M'_k$ are *equivalent* if $n = n'$, $k = k'$, and either y and y' are both free and $y = y'$, or for some i, $y = x_i$ and $y' = x'_i$.

DEFINITION 6.3 The *left spine* of a term is the set of all of its positions of the form 1^n for all n. A *spine redex* of a term is a redex whose position is on the left spine.

THEOREM 6.4 *A term M is solvable if and only if there are terms $N_1, \ldots, N_n$ such that $MN_1 \ldots N_n \to^* I$.*

PROOF. This is immediate from the Compressing Lemma and the fact that a finite term such as I cannot be the limit of a reduction sequence whose length is a limit ordinal. □

We can prove many relationships between head reduction, solvability, head reductions and finite reduction sequences.

THEOREM 6.5

1. *A term M is solvable if and only if there are terms $N_1, \ldots, N_n$ such that $MN_1 \ldots N_n \to^* I$.*
2. *A term is solvable if and only if it has a head normal form.*
3. *If a term has a head normal form, then it is reducible to head normal form in finitely many steps.*
4. *A term has a head normal form if and only if the head reduction sequence starting from that term terminates in finitely many steps.*
5. *All head normal forms of the same term are equivalent.*
6. *A term has no head normal form if and only if there is a reduction sequence starting from that term which contains infinitely many head reductions.*
7. *A term has no head normal form if and only if there is a reduction sequence starting from that term which contains infinitely many spine reductions.*
8. *The set of terms having head normal forms, and the complement of that set, are both closed under reduction.*

Reduction sequences which perform no reductions in unsolvable subterms are closely related to strong convergence.

LEMMA 6.6 *Every Cauchy continuous reduction sequence which performs no reductions inside any unsolvable subterm is strongly convergent.*

PROOF. Suppose a reduction sequence is Cauchy convergent but not strongly convergent. Then there must be some position u such that the sequence performs infinitely many reductions at u or descendants of u at the same depth, and only finitely many reductions at any proper prefix of u. Consider a final segment of the sequence starting after the last reduction at any proper prefix of u. The condition on the metric implies that these reductions must all be performed on the left spine of the subterm at u. Let t be the subterm at u of the term at the beginning of such a segment. The segment gives an infinite reduction of t containing infinitely many reductions on the left spine of u. By Theorem 6, t must be unsolvable, contrary to the hypothesis that no reduction is performed in unsolvable subterms. □

THEOREM 6.7 *A reduction sequence which performs no reductions inside any unsolvable subterm, and has maximal length with respect to that property, strongly converges to a term in which every redex is contained in an unsolvable subterm.*

PROOF. Immediate from Lemma 6.6, and the fact that all strongly convergent reduction sequences have countable length. □

DEFINITION 6.8 A *reduction strategy* is a function which maps each term t to a set of strongly convergent sequences which each start from t.

A strongly continuous sequence is *generated* by a reduction strategy F from a term t_0 if it consists of a concatenation of segments $\{S_\beta : t_\beta \to t_{\beta+1} \mid \beta < \alpha\}$ such that $S_\beta \in F(t_\beta)$ for all β.

A reduction strategy is *strongly normalising* if every reduction sequence which it generates from a term having a normal form reaches that normal form.

A reduction strategy is *strongly convergent* if every reduction sequence which it generates is strongly convergent.

THEOREM 6.9 *Every strongly convergent reduction strategy is strongly normalising. In particular, a reduction sequence which performs no reductions inside any unsolvable subterm, and has maximal length with respect to this property, strongly converges to a term in which every redex is contained in an unsolvable subterm.*

PROOF. The first statement follows from the fact that all strongly converging reductions are of countable length. The second follows from that and lemma 6.6

DEFINITION 6.10 *Böhm reduction* is reduction in $\Lambda^\infty_\perp$ by the β rule and the Böhm rule, viz. $M \to \perp$ if M is unsolvable and not $\perp$.

For technical convenience, we also define *strong* Böhm reduction: this is Böhm reduction subject to the restriction that the β rule may not be applied to any subterm of an unsolvable subterm, nor may the Böhm rule be applied to any proper subterm of an unsolvable subterm.

A *Böhm tree* is a normal form of $\Lambda_{\perp}^{\infty}$ with respect to Böhm reduction.

From the previous theorem it now follows that:

THEOREM 6.11 *Every term has a normal form with respect to Böhm reduction.*

LEMMA 6.12 *The Böhm rule (that is, the strategy mapping each term to the set of one-step reductions by the Böhm rule) is strongly convergent.*

Reductions by the Böhm rule are Church-Rosser.

DEFINITION 6.13 Write $B(t)$ for the unique normal form of t with respect to the Böhm rule.

LEMMA 6.14 *Let R be a set of redexes of a term t. If no member of R is contained in an unsolvable subterm of t, then every development of R is strongly convergent. In particular, R has a complete development.*

THEOREM 6.15 *Every term has a unique normal form with respect to Böhm reduction.*

PROOF. Theorem 6.5 provides almost all ingredients for a proof as we have given in [AKK+94]. It only remains to be proved that the following Church-Rosser property holds: $\leftarrow^{*} \circ \rightarrow^{*} \;\subseteq\; \rightarrow^{\infty} \circ \leftarrow^{\infty}$. □

LEMMA 6.16 *If $\perp$ does not occur in the Böhm tree of a term, its Böhm tree is its β-normal form.*

PROOF. If the construction of Theorem 6.11 produces a Böhm normal form not containing $\perp$, then it consists entirely of β-reductions, hence yields the β-normal form. □

THEOREM 6.17 *Böhm reduction is Church-Rosser and transfinitely normalising. Strong Böhm reduction is transfinitely strongly normalising.*

PROOF. The first is immediate from Theorem 6.15. The second is immediate from Lemma 6.9 and the fact that strong Böhm reduction is strongly convergent. □

The last theorem enables us to prove the infinite normal form property, a property which is slightly weaker than the (falsified) Church-Rosser property.

THEOREM 6.18 *Λ^{∞} has the transfinite NF property.*

PROOF. β-reduction is contained in Böhm reduction, and a β-normal form is also a Böhm normal form. The theorem then follows from Theorem 6.15. □

References

[AKK+94] Z.M. Ariola, J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Syntactic definitions of undefined: On defining the undefined. In *Proc. Theoretical Aspects of Computer Software, TACS'94*, pages 543–554. Sendai, Japan, LNCS 789, Springer Verlag, 1994.

[Bar84] H.P. Barendregt. *The Lambda Calculus, its Syntax and Semantics.* North-Holland, 2nd edition, 1984.

[Ken92] J.R. Kennaway. On transfinite abstract reduction systems. Technical Report CS-R9205, CWI, Amsterdam, 1992.

[KKSdV] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Transfinite reductions in orthogonal term rewriting systems. *To apear in Information and Computation*, ?? Available by ftp from ftp::/ftp.sys.uea.ac.uk/pub/kennaway/transfinite.{dvi,ps}.Z.