Postponement, Conservation and Preservation of Strong Normalisation for Generalised Reduction

Fairouz Kamareddine*

Abstract

Postponement of β_K-contractions and the conservation theorem do not hold for ordinary β but have been established by de Groote for a mixture of β with another reduction relation. In this paper, de Groote's results are generalised for a single reduction relation β_e which generalises β. We show morever, that β_e has the Preservation of Strong Normalisation property.

Keywords: Generalised β-reduction, Postponement of K-contractions, Generalised Conservation, Preservation of Strong Normalisation.

1 The λ-calculus with generalized reduction

In the term $((\lambda_x.\lambda_y.N)P)Q$, the abstraction starting with λ_x and the argument P form the redex $(\lambda_x.\lambda_y.N)P$. When this redex is contracted, the abstraction starting with λ_y and Q will in turn form a redex. It is important to note that Q (or some residual of Q) is the only argument that the abstraction (or some residual of the abstraction) starting with λ_y can ever have. This fact has been exploited by many researchers. Reduction has been extended so that the implicit redex based on the matching λ_y and Q is given the same priority as the intervening redex.

An initial attempt to generalize the notion of redex might be to define a rule like the following:

$$(\lambda_x.\lambda_y.N)PQ \to (\lambda_x.N[y{:=}Q])P$$

It quickly becomes evident that this is not sufficient as the following example shows:

Example 1 The proposed rule does not allow directly reducing the binding of y to Q in the term $A \equiv (\lambda_z.(\lambda_x.\lambda_y.N)P)RQ$.

We shall exploit the notion of a *well balanced segment* (sometimes known as a β-*chain*), which is the special case of one-hole contexts given by this grammar:

$$S ::= [\bullet] \mid (S[\lambda_x.[\bullet]])M \mid S[S]$$

*Department of Computing and Electrical Engineering, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, Scotland, *email*: fairouz@cee.hw.ac.uk, fax + 44 141 451 3327.

Using balanced segments, *generalized reduction* is then given by this rule:

$$S[\lambda_x.M]N \to S[M[x{:=}N]]$$

We find the above definition of well-balanced segments and generalised reduction rather cumbersome and believe that a more elegant definition can be given. In order to do so, we change from the classical notation to the *item notation*. Instead of writing $\lambda_x.M$, we write $[x]M$ and instead of MN we write $(N)M$.[1] Item notation has many advantages as shown in [7, 8]. Let us illustrate here with term A of Example 1, which we write in item notation as in Figure 1. We see immediately that the redexes

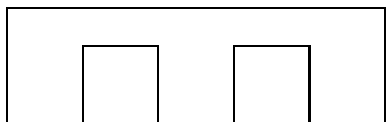


Figure 1: Redexes in item notation in term A

originate from the couples $(Q)[y]$, $(R)[z]$ and $(P)[x]$. Moreover, $(Q)(R)[z](P)[x][y]$ is a well-balanced segment. This natural matching was not present in the classical notation. We call items of the form (P) and $[x]$, application and abstraction items respectively. With item notation, generalised reduction is written as:

$$(M)\overline{s}[x]N \to_{g\beta} \overline{s}\{N[x := M]\} \text{for } \overline{s} \text{ well-balanced.}$$

(Here, { and } are used for grouping purposes so that no confusion arises.) For example:

$$(Q)(R)[z](P)[x][y]N \to_{g\beta} (R)[z](P)[x]\{N[y := Q]\}$$

Surely this is clearer than writing $(\lambda_z.(\lambda_x.\lambda_y.N)P)RQ \to_{g\beta} (\lambda_z.(\lambda_x.N[y := Q]P)R$.

2 An overview of generalised reduction in the literature

Generalized reduction was first introduced by Nederpelt in 1973 to aid in proving the strong normalization of AUTOMATH [19]. Kamareddine and Nederpelt have shown how generalised reduction makes more redexes visible, allowing flexibility in reducing a term [7]. Bloo, Kamareddine, and Nederpelt show that with generalised reduction one may indeed avoid size explosion without the cost of a longer reduction path and that simultaneously the λ-calculus can be elegantly extended with definitions which result in shorter type derivations [5]. Generalised reduction is strongly normalising [5] for all systems of the λ-cube [3].

An alternative approach to generalized reduction which has been followed by many researchers is to use one of these two local transformations:

$$(\theta) \quad (\lambda_x.N)PQ \to (\lambda_x.NQ)P$$
$$(\gamma) \quad (\lambda_x.\lambda_y.N)P \to \lambda_y.(\lambda_x.N)P$$

[1] Note that putting the argument before the function was first introduced by de Bruijn in his Automath project [20] and has been used by many researchers since. For example, Krivine in [17] also puts the argument before the function.

These rules transform terms to make more redexes visible to the ordinary notion of β-reduction. For example, both the γ and θ rules make sure that λ_y and Q in the term A of Example 1 can form a redex before the redex based on λ_x and P is contracted. That is:

$$(\theta_C) \quad (\lambda_x.(\lambda_y.N))PQ \to (\lambda_x.(\lambda_y.N)Q)P$$
$$(\gamma_C) \quad (\lambda_x.\lambda_y.N)PQ \to (\lambda_y.(\lambda_x.N)P)Q$$

Hence both θ and γ put λ_y next to its matching argument. The θ rule moves the argument next to its matching λ whereas γ moves the λ next to its matching argument.

Obviously, θ and γ are related to generalised reduction. In fact, θ and γ transform terms in order to make more potential redexes visible and then conventional β-reduction can be used to contract those newly visible redexes. Generalised reduction on the other hand, performs reduction on the potential redexes without having to bother to make them into classical redexes. The following example illustrates:

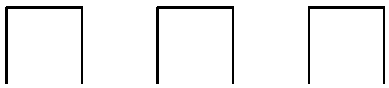
Example 2 Take again the term $A \equiv (\lambda_z.(\lambda_x.\lambda_y.N)P)RQ$. With generalised reduction we got: $(\lambda_z.(\lambda_x.\lambda_y.N)P)RQ \to_{g\beta} (\lambda_z.(\lambda_x.N[y := Q])P)R$. We illustrate how θ and γ work:

θ
$$(\lambda_z.(\lambda_x.\lambda_y.N)P)RQ \to_\theta (\lambda_z.(\lambda_x.\lambda_y.N)PQ)R \to_\theta$$
$$(\lambda_z.(\lambda_x.(\lambda_y.N)Q)P)R \to_\beta (\lambda_z.(\lambda_x.N[y := Q])P)R$$

γ
$$(\lambda_z.(\lambda_x.\lambda_y.N)P)RQ \to_\gamma (\lambda_z.\lambda_y(\lambda_x.N)P)RQ \to_\gamma$$
$$(\lambda_y(\lambda_z.(\lambda_x.N)P)R)Q \to_\beta (\lambda_z.(\lambda_x.N[y := Q])P)R$$

Note that in item notation it is easier to describe θ and γ. We illustrate with θ.

Example 3 We can reshuffle $(Q)(R)[z](P)[x][y]N$ to $(R)[z](P)[x](Q)[y]N$ in order to transform the bracketing structure {{ }{ }} into { }{ }{ }, where all the redexes correspond to adjacent '{' and '}'. Figure 1 can be redrawn using the θ-reduction twice in Figure 2.



(R) $[z]$ (P) $[x]$ (Q) $[y]$ N

Figure 2: θ-normal forms in item notation for term A

The θ rule can be applied to both explicitly and implicitly typed systems. However, the transfer of γ to explicitly typed systems is not straightforward, since in these systems the type of y in the term A may be affected by the reducible pair of λ_x and P. For example, it is fine to write $((\lambda_{x:*}.\lambda_{y:x}.y)z)u \to_\theta (\lambda_{x:*}.(\lambda_{y:x}.y)u)z$ but not to write $((\lambda_{x:*}.\lambda_{y:x}.y)z)u \to_\gamma (\lambda_{y:x}.(\lambda_{x:*}.y)z)u$.[2]

Local transformations like γ and θ began to appear in the literature during the eighties. (See [15] for a summary). Regnier [21] introduces the notion of a *premier redex* which is similar to the redex based on λ_y and Q above (which we call a *generalised redex*). Later, he uses θ and γ (and calls the combination σ) to show that

[2] An alternative is to apply γ to the *type erasure* of the term, which may be quite complicated to express in terms of the type-annotated term.

the perpetual reduction strategy finds the longest reduction path when the term is Strongly Normalising (SN) [22]. Vidal also introduces similar reductions [25]. Kfoury, Tiuryn, and Urzyczyn use θ (and other reductions) to show that typability in ML is equivalent to acyclic semi-unification [12]. Sabry and Felleisen describe a relationship between a reduction similar to θ and a particular style of CPS [23]. De Groote [6] uses θ and Kfoury and Wells [14] use γ to reduce the problem of β-strong normalisation to the problem of weak normalisation (WN) for related reductions. Kfoury and Wells use θ and γ to reduce typability in the rank-2 restriction of system F to the problem of acyclic semi-unification [13]. Klop, Sørensen, and Xi [16, 26, 24] use related reductions to reduce SN to WN. Finally, Ariola, Felleisen, Maraist, Odersky and Wadler use θ (called "let-C") in [1], as a part of an analysis of how to represent sharing in a call-by-need language implementation in a formal calculus.

All the research mentioned above is a living proof for the importance and usefulness of generalised reduction (from now on, β_e). For this reason, properties of this reduction must be studied. Confluence of β_e is a direct consequence of the fact that $M =_\beta N \Leftrightarrow M =_{\beta_e} N$. Subject reduction for β_e has been established in [5] (with the condition that *explicit definitions* must be added for some systems of the cube). And, as we mentioned earlier, Strong Normalisation of β_e has been established for the whole Cube and type derivation paths have been analysed. Other important properties of β_e have however remained unanswered. Those properties are:

1. *Preservation of Strong Normalisation PSN.* This property is: if M is strongly normalising for ordinary β-reduction (written M is β-SN), then M remains strongly normalising for generalised reduction β_e (i.e. M is also β_e-SN). PSN makes β_e a useful extension of β. This parallels the work on extending λ-calculi with explicit substitutions which satisfy the PSN property.

2. *Conservation of β_e-reduction.* This property is: if a term is β_{eI}-normalisable (i.e. β_e-normalisable reducing only redexes that don't erase their arguments, so called I-redexes, or strict redexes), then it is strongly normalisable. This is interesting in view of the ongoing interest of showing that strong normalisation can be reduced to weak normalisation [16, 24, 26].

3. *Postponement of K-reduction.* Generalised reduction allows the postponement of K-reduction (which discards their arguments) after I-reductions (which use their arguments in at least one place). Hence, generalised reduction allows unnecessary K-redexes to be bypassed. From the implementation point of view, this results in flexibility in work. Unnecessary work can be delayed or even avoided completely.

In this paper, we show these three properties for the generalised reduction β_e. We use item notation to be able to write generalised reduction in a *really general way* and to be able to describe proofs and proof objects elegantly. We believe that if this paper was written in classical notation, then the proofs would have been cumbersome to present.

3 Contributions of this paper and related work

Because we still have not introduced all the machinery of item notation, we shall use classical notation in this section.

Let us recall the three basic reduction rules of the λ-calculus ($FV(M)$ stands for the free variables of M):

(β)	$(\lambda_x.M)N \to M[x := N]$	
(β_I)	$(\lambda_x.M)N \to M[x := N]$	if $x \in FV(M)$
(β_K)	$(\lambda_x.M)N \to M$	if $x \notin FV(M)$

Redexes based on the β_I rule are called β_I- or I-redexes. Similarly, those based on the β_K rule are called β_K- or K-redexes. For any relation r, we write r_K and r_I for the corresponding K- and I-reductions.

In this paper, we show that the generalised reduction β_e satisfies PSN, the *postponement of K-contractions* and *conservation*. Of course the latter two properties fail for ordinary β as shown by the following example:

Example 4 $(\lambda_y.(\lambda_x.x))MN \to_{\beta_K} (\lambda_x.x)N \to_{\beta_I} N$ and it is impossible to β_I-reduce $(\lambda_y.(\lambda_x.x))MN$. Moreover, $((\lambda_x.\lambda_y.y(\lambda_z.zz))u)\lambda_z.zz$ is β_I-normalising but not strongly β-normalising.

Attempts have been made at establishing some reduction relations for which postponement of K-contractions and conservation hold ([2] and [6]). The picture is as follows (-N stands for normalising and $r \in \{\beta_I, \theta_K\}$ where (θ) was defined earlier):

(β_K-postponement for r)	If $M \to_{\beta_K} N \to_r O$ then $\exists P$ such that $M \twoheadrightarrow^+_{\beta_I\theta_K} P \twoheadrightarrow_{\beta_K} O$
(Conservation for β_I)	If M is β_I-N then M is β_I-SN
(Conservation for $\beta + \theta$)	If M is $\beta_I\theta_K$-N then M is β-SN

Conservation for β_I is found in [2]. Conservation for $\beta + \theta$ and β_K-postponement for $r \in \{\beta_I, \theta_K\}$ are established by de Groote in [6]. However, de Groote does not produce these results for a single reduction relation, but for β in which another relation (θ) is used. This paper establishes β_K-postponement and conservation for a single relation β_e and is hence the first to do so. Moreover, the relation θ is more restrictive than the generalised reduction of this paper.

Let us now list the postponement and conservation properties for β_e:

(β_{eK}-postponement for β_e)	If $M \to_{\beta_{eK}} N \to_{\beta_{eI}} O$ then $\exists P$ such that $M \to_{\beta_{eI}} P \twoheadrightarrow^+_{\beta_{eK}} O$
(Conservation for β_e)	If M is β_{eI}-N then M is β_e-SN

These two properties are important because here we have the first reduction relation which generalises β (yet $M =_\beta N \Leftrightarrow M =_{\beta_e} N$) and which satisfies them.

Now we come to the PSN property which is as follows:

$$\text{(PSN for } \beta_e) \quad M \text{ is } \beta\text{-SN} \Leftrightarrow M \text{ is } \beta_e\text{-SN}.$$

PSN not only means that β_e does not change the set of β-SN terms, but also that we can actually use β_e with explicit substitution. In fact, explicit substitution is an important topic of research and PSN is an important property for any λ-calculus

extended with explicit substitution. In fact, lately, much research has been carried out ([4, 9]) in order to find systems of explicit substitution which are both confluent and have the PSN property (if M is β-SN then M is λ_s-SN where λ_s is the lambda calculus extended with explicit substitution). This is the reason for our interest in PSN of β_e (which is confluent by the way). After all, generalised reductions à la β_e have been extensively used as we saw in Section 2 for both theoretical and practical reasons. Furthermore, systems of explicit substitution have been the subject of much recent research. Both generalised reduction and explicit substitution are of practical importance and combining them both in one system may turn out to be very useful. The main benefits of these concepts are similar: both emphasize flexibility in the ordering of operations. In particular, both generalized reduction and explicit substitution allow the postponement of work, but in different, complementary ways. On one side, generalized reduction always allows unnecessary K-redexes to be bypassed. Explicit substitution will not in general allow this, since reducing the K-redex might be necessary to expose an essential I-redex. Similarly, on the other side, explicit substitution allows bypassing any work inside a subterm that will be discarded later. However, generalized reduction does not provide any means for performing only those parts of a substitution that will be used later. Thus, we can see that their benefits are complementary.

We claim that a system with the combination of generalized reduction and explicit substitution is more advantageous than a system with each concept separately. Obviously, if the benefits of both are desired simultaneously, it is important to study the combination, a task which this paper performs. Before the combination can be safely used, it must be checked that this combination is sound and safe exactly like it has been checked that each of explicit substitutions and generalised reductions separately are sound and safe.

Once PSN is established we can study extending the λ-calculus with both explicit substitution and generalised reduction. This means that we can combine the advantages of the two different extensions in one system [10, 11].

We had established in [9] property (1) below, and in [10, 11] property (2) below ($\lambda_{\beta_e s}$ stands for the lambda calculus extended with explicit substitution and generalised reduction and for reasons of uniformity, we write λ-SN for β-SN and λ_{β_e}-SN for β_e-SN):

$$\begin{array}{lcl} (1) & M \text{ is } \lambda\text{-SN} & \Leftrightarrow \quad M \text{ is } \lambda_s\text{-SN} \\ (2) & M \text{ is } \lambda_s\text{-SN} & \Leftrightarrow \quad M \text{ is } \lambda_{\beta_e s}\text{-SN} \end{array}$$

The proofs for (1) and (2) are similar. Now with PSN, we get (3) below and then (4) comes for free.

$$\begin{array}{lcl} (3) & M \text{ is } \lambda\text{-SN} & \Leftrightarrow \quad M \text{ is } \lambda_{\beta_e}\text{-SN} \\ (4) & M \text{ is } \lambda_{\beta_e}\text{-SN} & \Leftrightarrow \quad M \text{ is } \lambda_{\beta_e s}\text{-SN} \end{array}$$

Hence, one gets: M is λ-SN $\Leftrightarrow$ M is λ_{β_e}-SN $\Leftrightarrow$ M is $\lambda_{\beta_e s}$-SN $\Leftrightarrow$ M is λ_s-SN.

Based on the above discussion, this article shows β_{eK} postponement (Section 5), the generalised conservation for β_e (Section 6), and the PSN property for β_e (Section 7).

4 The formal machinery

We assume the reader familiar with the λ-calculus whose terms are

$$\Lambda ::= \mathcal{V} | (\Lambda\Lambda) | (\lambda_{\mathcal{V}}.\Lambda)$$

We take terms modulo α-conversion and use the variable convention VC (as in [3]) which avoids any clash of variables. We use x, y, z, x_1, x_2, ... and M, N, P, Q, A, B, A_1, ... to range over $\mathcal{V}$ and Λ respectively. We assume the usual definition of substitution and use $FV(M)$ for the set of free variables of M. Because we need to see redexes (ordinary and generalised) we shall write terms in *item notation* (see [8] or [7]). In this notation, λ_x is written as $[x]$ and (MN) is written $(N)M$ (note that following de Bruijn, we put the argument before the function). $[x]$ and (N) are called *items*. A sequence of items is called a *segment*. We use $I, I_1, \ldots$ to range over items and $S, S_1, S_2, \ldots$ to range over segments. A *well-balanced* segment (w.b for short) is defined as the empty segment or $(P)S_1[x]S_2$ where S_1 and S_2 are w.b. Note that the concatenation of w.b segments is a well-balanced segment.

One particular advantage of this notation is that redexes are more clear than in the usual notation. For example, γ_C of Section 2 becomes:

$$(\gamma_C) \qquad (Q)(P)[x][y]N \to (Q)[y](P)[x]N$$

where it is clear that (P) matches $[x]$ and (Q) matches $[y]$. So, an ordinary redex starts with a $()$ adjacent to $[]$. A generalised redex starts with $()S[]$ where S is w.b. When $S = \emptyset$, a generalised redex is an ordinary redex. In $(Q)(P)[x][y]N$, we say that (P), $[x]$, (Q) and $[y]$ are *partnered*, (P) is the *partner* of $[x]$ (or $[x]$ is the partner of (P)) and (Q) is the partner of $[y]$. (P) and $[x]$ are also said to be β-partnered whereas (Q) and $[y]$ are β_e-partnered. In general, we say that (P) (or $[x]$) is partnered in M if:

- $M \equiv (P)S[x]N$ where S is w.b (in this case (P) and $[x]$ are partners), or
- $M \equiv [y]N$ and (P) (or $[x]$) is partnered in N, or
- $M \equiv (N_1)N_2$ and (P) is either partnered in N_1 or in N_2.

We may also talk of β_I-, β_{eI}-, β_K-, β_{eK}-partnered items with the obvious meaning. Note that if $S_1(A)S_2[x]S_3$ is w.b where (A) and $[x]$ are partnered then S_2 and S_1S_3 are w.b.

If an item is not partnered in a term we say that it is *bachelor* (and may talk of β-, β_{eI}-, β_K-, β_{eK}-, β_I- and β_e-bachelor items). A segment consisting of bachelor items only is called bachelor. Note that a term will always be written as $I_1 I_2 \ldots I_n x$. Each I_i is said to be a *main*-item in M. A main item can of course have items inside it but these will not be main in M. For example, $((y)[x]x)[z]z$ has the main items $((y)[x]x)$ and $[z]$. The redex $((y)[x]x)[z]z$ is said to be a main-redex. The other redex $(y)[x]x$ is not main. The *weight* of a segment is defined to be the number of its main items. We write $[x := N]M$ instead of $M[x := N]$ which stands for substituting N for the free occurrences of x in M.

We assume the reader familiar with the basic machinery of reduction ([2], p. 50-59). In particular, if R is a binary relation $\subset \Lambda \times \Lambda$, and $(M, N) \in R$, we call M the R-redex and N the contractum of M. Given $R \subset \Lambda \times \Lambda$, we define $\rightarrow_R$ to be the least compatible relation containing R, $\twoheadrightarrow_R$ to be its reflexive transitive closure and $=_R$ to be its reflexive, symmetric and transitive closure. A term M is said to be in R-normal form (R-nf) iff there is no N such that $M \rightarrow_R N$. M is said to have a R-nf, iff there is N in R-nf such that $M \twoheadrightarrow_R N$. We say M is R-normalising or is R-N iff M has a R-nf. We say that M is strongly R-normalising and write M is R-SN iff there is no infinite R-reduction path starting at M. We may use $M \twoheadrightarrow_R^+ N$ to indicate the existence of one or more steps from M to N and $M \twoheadrightarrow_R^n N$ to mean that there are n reduction steps. Ordinary β-, β_I- and β_K-reduction are defined as the reduction relations generated by the corresponding rules below:

(β)	$(N)[x]M \rightarrow [x := N]M$	
(β_I)	$(N)[x]M \rightarrow [x := N]M$	if $x \in FV(M)$
(β_K)	$(N)[x]M \rightarrow M$	if $x \notin FV(M)$

As explained in Example 4, postponement of K-contractions and conservation do not hold for β. De Groote in [6] introduces different reduction relations for which he establishes these properties. First, [6] uses

$$(\theta_K) \qquad (O)(N)[x]M \rightarrow (N)[x](O)M \qquad \text{if } x \notin FV(M)$$

Note that by VC, in θ_K, $x \notin FV(O)$. Then, de Groote moves (O) to the right of $(N)[x]$ so that it can eventually occur adjacent to its partner in M if it exists. De Groote establishes the following two results ($r \in \{\beta_I, \theta_K\}$):

(β_K-postponement for r)	If $M \rightarrow_{\beta_K} N \rightarrow_r O$ then $\exists P$ such that $M \twoheadrightarrow^+_{\beta_I\theta_K} P \twoheadrightarrow_{\beta_K} O$
(Conservation for $\beta + \theta$)	If M is $\beta_I\theta_K$-N then M is β-SN.

In this paper, we will improve both results. We will define a β_e-reduction relation (see Definition 5) whose β_{eI} and β_{eK} stand for its I and K-reductions. We shall show that:

(β_{eK}-postponement for β_e)	If $M \rightarrow_{\beta_{eK}} N \rightarrow_{\beta_{eI}} O$ then $\exists P$ such that $M \rightarrow_{\beta_{eI}} P \twoheadrightarrow^+_{\beta_{eK}} O$
(Conservation for β_e)	If M is β_{eI}-N then M is β_e-SN.

Definition 5 (Generalised β-reduction β_e) We generalise β, β_I and β_K to the reduction relations generated by the corresponding rules of what follows:

(β_e)	$(N)S[x]M \rightarrow S[x := N]M$	if S is w.b
(β_{eI})	$(N)S[x]M \rightarrow S[x := N]M$	if S is w.b and $x \in FV(M)$
(β_{eK})	$(N)S[x]M \rightarrow SM$	if S is w.b and $x \notin FV(M)$

Note that β_e is more generalised than the reduction relation introduced by combining de Groote's $\beta + \theta_K$. In fact, β_e is not restricted to K-redexes and one unique step can do the work of many in Groote's sense. For example, if $S \equiv (A_1)[x_1](A_2)[x_2]\ldots(A_n)[x_n]$ and all the redexes starting with $(A_1), (A_2), \ldots (A_n)$ are K-redexes in $S[x]M$, then $(N)S[x]M \rightarrow_{\beta_e} S[x := N]M$ iff $(N)S[x]M \twoheadrightarrow^n_{\theta K} S(N)[x]M \rightarrow_\beta S[x := N]M$.

Now, here is a basic lemma about terms:

Lemma 6

1. Let $r \in \{\beta_e, \beta_{eI}, \beta_{eK}\}$. If (A) is r-bachelor in $(A)M$ then (B) is also r-bachelor in $(B)(A)M$.
2. If M is in β-nf, then $M \equiv [x_1][x_2]\ldots[x_n](A_1)(A_2)\ldots(A_m)z$ where $n \geq 0$, $m \geq 0$ and $\forall i,\ 1 \leq i \leq m \Rightarrow A_i$ is in β-nf.
3. If $A \rightarrow_r A'$ then $SA \rightarrow_r SA'$ for any segment S and any reduction relation $r \in \{\beta, \beta_I, \beta_K, \beta_e, \beta_{eI}, \beta_{eK}\}$.

Proof

1. If (B) was r-partnered, then $(B)(A)M \equiv (B)(A)S[x]N$ where $(A)S$ is w.b (and hence $(A)S \equiv (A)S_1[y]S_2$ where S_1, S_2 are w.b) contradicting the fact that (A) is r-bachelor.
2. By induction on the structure of M.
3. By induction on the weight of S.

In order to show the Preservation of Strong Normalisation for β_e, we need a reduction strategy where a β_K-redex $(M)[x]N$ is contracted only if M is in β-nf. This strategy is actually the perpetual strategy (see [2] and [22]):

Definition 7 We define the perpetual strategy F as follows:

$$
\begin{array}{lcll}
F([x]M) & = & F(M) & \\
F((M)N) & = & F(N) & \text{if } N \not\equiv [x]P \text{ and } N \text{ is not in } \beta\text{-nf} \\
F((M)N) & = & F(M) & \text{if } N \not\equiv [x]P \text{ and } N \text{ is in } \beta\text{-nf} \\
F((M)[x]N) & = & (M)[x]N & \text{if } x \in FV(N) \text{ or } M \text{ is in } \beta\text{-nf} \\
F((M)[x]N) & = & F(M) & \text{if } x \notin FV(N) \text{ and } M \text{ is not in } \beta\text{-nf}
\end{array}
$$

We call perpetual reduction the reduction associated with this strategy. When M β-reduces to N by contracting $F(M)$, we write, $M \rightarrow_F N$. This strategy has been shown in [22] to give the longest path for a SN term. It was moreover, shown in [2] that M is β-SN iff its perpetual reduction terminates. With the result of this paper, it will also be the case that M is β_e-SN iff its perpetual path terminates.

The following lemma is informative about where F-reduction takes place in a term in the case of K-redexes:

Lemma 8 If $M \rightarrow_F N$ where $F(M)$ is a β_K-redex, then one of the following holds:

1. $M \equiv [x_1][x_2]\ldots[x_m](A_1)(A_2)\ldots(A_n)(A)[x]P$ and
   $N \equiv [x_1][x_2]\ldots[x_m](A_1)(A_2)\ldots(A_n)P$
   where $x \notin FV(P)$, A is in β-nf, $n \geq 0$ and $m \geq 0$.
2. $M \equiv [x_1][x_2]\ldots[x_m](A_1)(A_2)\ldots(A_n)(A)[x]P$ and
   $N \equiv [x_1][x_2]\ldots[x_m](A_1)(A_2)\ldots(A_n)(A')[x]P$
   where $x \notin FV(P)$, A is not in β-nf, $A \rightarrow_F A'$, $n \geq 0$ and $m \geq 0$.

3. $M \equiv [x_1][x_2]\ldots[x_m](A_1)(A_2)\ldots(A_n)(A)(B_1)(B_2)\ldots(B_r)z$ and
   $N \equiv [x_1][x_2]\ldots[x_m](A_1)(A_2)\ldots(A_n)(A')(B_1)(B_2)\ldots(B_r)z$ and
   A is not in β-nf, $A \to_F A'$, $n \geq 0$, $m \geq 0$ and $r \geq 0$ and $\forall i, 1 \leq i \leq r, B_i$ is in β-nf.

Proof By induction on $M \to_F N$ where $F(M)$ is a β_K-redex.

5 Postponement of β_{eK}-reduction

In this section, we establish in lemma 10 the postponement of β_{eK}-reduction. The proof of postponement is similar to that of de Groote. For us, however, we can get away with only one step β_{eI} reduction in the postponement lemma (Lemma 10). De Groote, had to have many steps in order to accommodate the slow process of moving an item () next to its matching [] (see for example his proof of Lemma 11, (c) ii). We could also in Lemma 10, replace β_{eK} with ordinary β_K in $P \twoheadrightarrow^+_{\beta_{eK}} O$ but we won't bother doing so in this paper as it is not needed.

The following lemma establishes that substitution preserves β_{eK}-reduction.

Lemma 9 If $M \to_{\beta_{eK}} N$ then the following hold:

1. $[x := M]P \twoheadrightarrow_{\beta_{eK}} [x := N]P$.
2. If $x \in FV(P)$ then $[x := M]P \twoheadrightarrow^+_{\beta_{eK}} [x := N]P$.
3. $[x := P]M \to_{\beta_{eK}} [x := P]N$.

Proof Use induction on the structure of P for 1 and 2, and on the derivation of $M \to_{\beta_{eK}} N$ for 3.

Now we come to the postponement lemma. Note that in this lemma, $P \twoheadrightarrow^+_{\beta_{eK}} O$ and not $P \twoheadrightarrow_{\beta_{eK}} O$ nor $P \to_{\beta_{eK}} O$. This is due to Lemma 9.

Lemma 10 If $M \to_{\beta_{eK}} N \to_{\beta_{eI}} O$ then $\exists P$ such that $M \to_{\beta_{eI}} P \twoheadrightarrow^+_{\beta_{eK}} O$.

Proof By induction on the derivation of $M \to_{\beta_{eK}} N$.

- Case $(A)S[x]B \to_{\beta_{eK}} SB$, S w.b, $x \notin FV(B)$, check where in SB the β_{eI}-redex appears (note that if $S_1S_2S_3$ and S_2 are w.b, then S_1S_3 is w.b).
  We only treat the case where $S \equiv S_1(A_1)S_2[y]S_3$ with S_2 w.b and
  $S_1(A_1)S_2[y]S_3B \to_{\beta_{eI}} S_1S_2\{[y := A_1]S_3\}[y := A_1]B$.
  Then $(A)S_1(A_1)S_2[y]S_3[x]B \to_{\beta_{eI}} (A)S_1S_2\{[y := A_1]S_3\}[x]\{[y := A_1]B\} \to_{\beta_{eK}}$
  $S_1S_2\{[y := A_1]S_3\}\{[y := A_1]B\}$ as $x \notin FV([y := A_1]B)$ due to VC.
- Case $[x]M \to_{\beta_{eK}} [x]N \to_{\beta_{eI}} O$, then $O \equiv [x]Q$. Use IH on $M \to_{\beta_{eK}} N \to_{\beta_{eI}} Q$.
- Case $(A)B \to_{\beta_{eK}} (A')B \to_{\beta_{eI}} O$, investigate how $(A')B \to_{\beta_{eI}} O$. We only treat the case where $(A')B \equiv (A')S[x]B_1 \to_{\beta_{eI}} S[x := A']B_1$. Then
  $(A)B \equiv (A)S[x]B_1 \to_{\beta_{eI}} S[x := A]B_1 \twoheadrightarrow^+_{\beta_{eK}} S[x := A']B_1$ by Lemma 9.

- Case $(A)B \to_{\beta_{eK}} (A)B' \to_{\beta_{eI}} O$, we only treat the case where $B' \equiv S[x]C$, S is w.b, and $O \equiv S[x := A]C$. I.e. $(A)B \to_{\beta_{eK}} (A)S[x]C \to_{\beta_{eI}} S[x := A]C$ (note that it cannot occur that $B \equiv (Q)S[x][y]C \to_{\beta_{eK}} S[x]([y := Q]C)$). Then one of the following holds:
  - Case $B \equiv S[x]C_1$ and $C_1 \to_{\beta_{eK}} C$, then $(A)S[x]C_1 \to_{\beta_{eI}} S[x := A]C_1 \to_{\beta_{eK}} S[x := A]C$ by Lemma 9, case 3.
  - Case $B \equiv S_1[x]C$ and the β_{eK}-redex is in S_1, i.e. $(A)S_1[x]C \to_{\beta_{eK}} (A)S[x]C \to_{\beta_{eI}} S[x := A]C$, then $(A)S_1[x]C \to_{\beta_{eI}} S_1[x := A]C \to_{\beta_{eK}} S[x := A]C$ by VC.

6 The generalised conservation for β_e

In this section, we establish in Theorem 28, the generalised conservation for β_e, the same relation for which we established in the previous section, the postponement of its K-reduction. This is an extension of de Groote's work which established the postponement and conservation properties for two different relations. We start by defining the set of labelled terms which will help us in establishing the generalised conservation. Labels will used as counters to record the number of contracted redexes when reducing a term.

Definition 11 The set ${}^{\mathbb{N}}\Lambda$ of labelled λ-terms is inductively defined as follows:

1. $n \in \mathbb{N}, x \in \mathcal{V} \Rightarrow {}^n x \in {}^{\mathbb{N}}\Lambda$.
2. $n \in \mathbb{N}, x \in \mathcal{V}, M \in {}^{\mathbb{N}}\Lambda \Rightarrow {}^n[x]M \in {}^{\mathbb{N}}\Lambda$.
3. $n \in \mathbb{N}, M, N \in {}^{\mathbb{N}}\Lambda \Rightarrow {}^n(M)N \in {}^{\mathbb{N}}\Lambda$

We take $M, N, O, A, B, \ldots$ to range over labelled λ-terms. We use ${}^n M$ to stress that the outermost label of a λ-term M is n. Hence, M and ${}^n M$ stand for the same labelled λ-term. We write ${}^{+m}M$ for the labelled λ-term obtained by adding m to the outermost label of a labelled λ-term M. Hence if the outermost label of M is n then ${}^{+m}M$ denotes ${}^{n+m}M$.

For $M \in {}^{\mathbb{N}}\Lambda$, we write $|M|$ for the (unlabelled) λ-term in Λ obtained by erasing all labels in M. Moreover, if $M \in \Lambda$, we identify M with M' in ${}^{\mathbb{N}}\Lambda$ such that $|M'| \equiv M$ and all labels in M' are 0. Hence, $\Lambda \subset {}^{\mathbb{N}}\Lambda$.

We use in this section, the notations and techniques of de Groote adapted however to our generalised reduction. Basically the idea is as follows:

1. **Church Rosser CR:** We introduce a labelled reduction relation $\to_{\beta^+_{eI}}$ which we prove Church Rosser. $\to_{\beta^+_{eI}}$ is shown CR by showing that a related reduction relation $\to_1$ is CR. Hence, if a labelled term M has a β^+_{eI}-nf, it must be unique.
2. **Increasing property Inc:** We then introduce the *weight* of a term M, $\Theta[M]$, which is used to limit the length of β^+_{eI}-reductions starting at normalising terms. That is, the length of any sequence of β^+_{eI}-reductions starting at a normalising term M is bounded by $\Theta[M'] - \Theta[M]$ where M' is the (unique) β^+_{eI}-nf of M. This implies that any β^+_{eI}-N term is β^+_{eI}-SN. This will be extended to β_{eI} by showing that any β_{eI}-N term is β_{eI}-SN.

3. **Weak Normalisation ⇒ Strong Normalisation (WN ⇒ SN)** Next we show that if M is β_{eI}-N then it is β_e-SN by using the fact that M is β_{eI}-SN, postponement and that there can only be a finite β_{eK}-redexes. This establishes the generalised conservation.

Note that the structure of our proof can be seen as: CR+Inc+WN ⇒ SN. This is as we said a generalisation of the proof of de Groote. One could however use Corollary 5.19 in Klop's thesis [16] which states that WCR+Inc+WN ⇒ SN & CR where WCR is Weak Church Rosser. We leave this alternative to the reader to establish.

Here is the definition of substitution on labelled terms:

Definition 12 Let $M, N \in {}^{\mathbf{N}}\Lambda$. $[x := N]M$ is defined as follows:

$$\begin{array}{lcll} [x := {}^n N]^m x & \equiv & {}^{n+m}N & \\ [x := {}^n N]^m y & \equiv & {}^m y & \text{if } x \not\equiv y \\ [x := {}^n N]^m (P)Q & \equiv & {}^m([x := {}^n N]P)[x := {}^n N]Q & \\ [x := {}^n N]^m [y]M & \equiv & {}^m[y][x := {}^n N]M & \end{array}$$

Now we define $\rightarrow_{\beta^+_{eI}}$ which will be used to show conservation.

Definition 13 $M \rightarrow_{\beta^+_{eI}} N$ is defined inductively as follows:

1. ${}^n({}^i N)S^o[x]^j M \rightarrow_{\beta^+_{eI}} {}^{+n+o+1}S[x := {}^i N]^j M$ if $x \in FV(M), S$, w.b.
2. If $M \rightarrow_{\beta^+_{eI}} N$ then ${}^n[x]M \rightarrow_{\beta^+_{eI}} {}^n[x]N$, ${}^n(M)P \rightarrow_{\beta^+_{eI}} {}^n(N)P$ and ${}^n(P)M \rightarrow_{\beta^+_{eI}} {}^n(P)N$

$\twoheadrightarrow_{\beta^+_{eI}}$ is defined as the transitive reflexive closure of $\rightarrow_{\beta^+_{eI}}$.

We define $\rightarrow_1$ for which CR is easier to show than for $\rightarrow_{\beta^+_{eI}}$.

Definition 14 $M \rightarrow_1 N$ is defined inductively as follows:

1. $M \rightarrow_1 M$
2. If $M \rightarrow_1 N$ then ${}^n[x]M \rightarrow_1 {}^n[x]N$
3. If $M \rightarrow_1 O$ and $N \rightarrow_1 P$ then ${}^n(M)N \rightarrow_1 {}^n(O)P$
4. If $S^p[x]M \rightarrow_1 S'^q[x]O$, $N \rightarrow_1 P$, S, S' w.b, and $x \in FV(M)$ then ${}^n(N)S^p[x]M \rightarrow_1 {}^{+n+q+1}S'[x := P]O$.

$\twoheadrightarrow_1$ is defined as the transitive reflexive closure of $\rightarrow_1$.

The following lemma shows that labels can be increased for both $\rightarrow_1$ and $\rightarrow_{\beta^+_{eI}}$.

Lemma 15 Let $M, N \in {}^{\mathbf{N}}\Lambda$ and $r \in \{1, \beta^+_{eI}\}$. If $M \rightarrow_r N$ then ${}^{+n}M \rightarrow_r {}^{+n}N$.

Proof By induction on the derivation $M \rightarrow_r N$.

The following lemma shows that $\rightarrow_1$ and $\rightarrow_{\beta^+_{eI}}$ are closed under substitution.

Lemma 16 Let $M, N, P, O \in {}^{\mathbf{N}}\Lambda$. The following hold:

1. If $M \to_1 N$, then $[x := M]^m O \to_1 [x := N]^m O$.
2. If $M \to_{\beta^+_{eI}} N$, then $[x := M]^m O \twoheadrightarrow_{\beta^+_{eI}} [x := N]^m O$.
3. If $M \to_1 N$ and $O \to_1 P$ then $[x := O]M \to_1 [x := P]N$.
4. If $M \to_{\beta^+_{eI}} N$ and $O \to_{\beta^+_{eI}} P$ then $[x := O]M \twoheadrightarrow_{\beta^+_{eI}} [x := P]N$.

Proof 1 and 2 are similar and are by induction on the structure of O. 3 and 4 are by induction on the derivation $M \to_r N$ where r is the corresponding reduction.

Here is the relationship between $\to_1$ and $\to_{\beta^+_{eI}}$:

Lemma 17 $M \twoheadrightarrow_1 N$ iff $M \twoheadrightarrow_{\beta^+_{eI}} N$.

Proof $\Rightarrow$) By induction on the derivation of $M \to_1 N$ show that:
$M \to_1 N \Rightarrow M \twoheadrightarrow_{\beta^+_{eI}} N$.
$\Leftarrow$) By induction on the derivation $M \to_{\beta^+_{eI}} N$, show that $M \to_{\beta^+_{eI}} N \Rightarrow M \to_1 N$.

The following two lemmas enable us to establish that $\to_1$ is CR.

Lemma 18 If S, S' w.b, none of the binding variables of $S[x]$ occurs free in N, none of the binding variables of $S'[x]$ occurs free in P, none of the binding variables of N are free in M and none of the binding variables of P are free in O, $S^p[x]M \to_1 S'^q[x]O$ and $N \to_1 P$ then ${}^{+p}S[x := N]M \to_1 {}^{+q}S'[x := P]O$.

Proof Note that if $\mathtt{weight}(S) = \mathtt{weight}(S')$ then $p = q$, $M \to_1 O$ and if (A_i) and (A'_i) are the ith main application items of S and S' respectively, then $A_i \to_1 A'_i$. Hence the result is shown by Lemmas 15 and 16 and the def. of $\to_1$.
If $\mathtt{weight}(S) > \mathtt{weight}(S')$, then we prove the lemma by induction on $\mathtt{weight}(S)$.

Lemma 19 Let $M, N, O \in {}^{\mathbf{N}}\Lambda$ such that $M \to_1 N$ and $M \to_1 O$ then $\exists P \in {}^{\mathbf{N}}\Lambda$ such that $N \to_1 P$ and $O \to_1 P$.

Proof By induction on the derivation of $M \to_1 N$.

Now we have the Church Rosser property for $\to_1$:

Corollary 20 (Church Rosser of $\to_1$) Let $M, N, O \in {}^{\mathbf{N}}\Lambda$ such that $M \twoheadrightarrow_1 N$ and $M \twoheadrightarrow_1 O$ then $\exists P \in {}^{\mathbf{N}}\Lambda$ such that $N \twoheadrightarrow_1 P$ and $O \twoheadrightarrow_1 P$.

Hence, the first part of this section (CR of $\to_{\beta^+_{eI}}$) is done:

Lemma 21 (Church Rosser of $\to_{\beta^+_{eI}}$) Let $M, N, O \in {}^{\mathbf{N}}\Lambda$ such that $M \twoheadrightarrow_{\beta^+_{eI}} N$ and $M \twoheadrightarrow_{\beta^+_{eI}} O$ then $\exists P \in {}^{\mathbf{N}}\Lambda$ such that $N \twoheadrightarrow_{\beta^+_{eI}} P$ and $O \twoheadrightarrow_{\beta^+_{eI}} P$.

Proof By Corollary 20 and Lemma 17.

In order to show Lemma 25, we introduce the following definition:

Definition 22 The weight $\Theta[M]$ of a labelled λ-term M is defined as follows:

$$\begin{array}{lcl} \Theta[^n x] & = & n \\ \Theta[^n [y]M] & = & n + \Theta[M] \\ \Theta[^n (M)N] & = & n + \Theta[M] + \Theta[N] \end{array}$$

The following two lemmas enable us to establish that the weight as we defined will help us to measure terminating reductions:

Lemma 23 If $x \in FV(M)$ then $\Theta[[x := N]M] \geq \Theta[M] + \Theta[N]$.

Proof By induction on the structure of M showing first that $\Theta[^{+m} M] = m + \Theta[M]$.

Lemma 24 Let $M, N \in {}^{\boldsymbol{N}}\Lambda$ and $M \twoheadrightarrow^{+}_{\beta^{+}_{eI}} N$ then $\Theta[M] < \Theta[N]$.

Proof By induction on the derivation $M \twoheadrightarrow^{+}_{\beta^{+}_{eI}} N$ using Lemma 23.

Now, β^{+}_{eI}-N and β^{+}_{eI}-SN are the same:

Lemma 25 If M is β^{+}_{eI}-N then M is β^{+}_{eI}-SN.

Proof Since M is β^{+}_{eI}-N, and since β^{+}_{eI} is Church Rosser by Lemma 21, then M has a unique β^{+}_{eI}-nf M'. According to Lemma 24, the length of any sequence of β^{+}_{eI}-reduction starting at M is bounded by $\Theta[M'] - \Theta[M]$.

Here is the relationship between $\rightarrow_{\beta_{eI}}$and $\rightarrow_{\beta^{+}_{eI}}$:

Lemma 26 Let $M, N \in \Lambda$ such that $M \rightarrow_{\beta_{eI}} N$, then there exist $M', N' \in {}^{\boldsymbol{N}}\Lambda$ such that $|M'| \equiv M, |N'| \equiv N$ and $M' \rightarrow_{\beta^{+}_{eI}} N'$. Furthermore, if N is in β_{eI}-nf then N' is in β^{+}_{eI}-nf.

Proof Easy. Put the right labels on M and N obtaining M', N' where $M' \rightarrow_{\beta^{+}_{eI}} N'$.

Now, we generalise Lemma 25 to $\rightarrow_{\beta_{eI}}$.

Theorem 27 If M is β_{eI}-N then M is β_{eI}-SN.

Proof M β_{eI}-N $\Rightarrow^{Lemma\ 26}$ M β^{+}_{eI}-N $\Rightarrow^{Lemma\ 25}$ M β^{+}_{eI}-SN $\Rightarrow$ M β_{eI}-SN (otherwise there exists an infinite β^{+}_{eI}-path).

Finally, from the above theorem and postponement of K-contractions, we can establish conservation:

Theorem 28 (Conservation) If M is β_{eI}-N then M is β_{e}-SN.

Proof If M is not β_{e}-SN then there is an infinite β_{e}-path starting at M. But by postponement of β_{eK} redexes, and by the fact that there can only be a finite β_{eK}-contractions, there must be an infinite β_{eI}-path. But M is β_{eI}-N and so it is β_{eI}-SN by Theorem 27. Contradiction.

7 Preservation of Strong Normalisation

To show PSN, we show that if $M \twoheadrightarrow_F N$ (using the perpetual strategy of Definition 7) and if N is β_{eI}-N then M is β_{eI}-N. Now, we take M which is β-SN, and its perpetual path to its normal form N. As N is β_{eI}-N, then M is β_{eI}-N and hence by conservation, M is β_e-SN. It is possible however to show PSN in many different ways and without using conservation. For example, one may use a result of Regnier in [22] which states that *the length of the longest reduction of a term is invariant by σ-equivalence* noting that the β-reduction modulo σ-equivalence is isomorphic to the β_e-reduction.[3]

In this section, we will use the already available conservation theorem and following Theorem 27, we interchange β_{eI}-SN and β_{eI}-N at liberty. We shall show PSN of β_e. Note that the derivation:

$$M\ \beta\text{-SN} \Rightarrow M\ \beta\text{-N} \Rightarrow M\ \beta_{eI}\text{-N} \Rightarrow M\ \beta_e\text{-SN}$$

is incorrect because $M\ \beta$-N $\not\Rightarrow M\ \beta_{eI}$-N. For example, $(\lambda_x.y)\Omega$ is β-N but not β_{eI}-N for $\Omega \equiv (\lambda_z.zz)(\lambda_z.zz)$. To show PSN, we take M that is β-SN. Then $M \twoheadrightarrow_F N$ where N is the β-nf of M and $\twoheadrightarrow_F$ is the perpetual strategy. As N is in β-nf, then N is β_{eI}-N. But the inverse of $\twoheadrightarrow_F$ preserves β_{eI}-N. Hence, M is β_{eI}-N and by conservation, M is β_e-SN.

In order to establish that the inverse of $\twoheadrightarrow_F$ preserves β_{eI}-normalisation (Theorem 33), we need the following three lemmas which will be combined with the three forms of perpetual reduction for K-redexes as in Lemma 8.

Lemma 29 If $(A_1)\dots(A_n)(A)[x]P$ has β_{eI}-nf, $x \notin FV(P)$, then its β_{eI}-nf is of the form $(B_1)\dots(B_j)(A')[x]Q$ where A' is the β_{eI}-nf of A, $0 \leq j \leq n$, B_j is the β_{eI}-nf of some A_i for $1 \leq i \leq n$. Moreover, $(A_1)\dots(A_n)P$ has $(B_1)\dots(B_j)Q$ as its β_{eI}-nf.

Proof By induction on $n \geq 0$.

- $n = 0$, the β_{eI}-nf is $(A')[x]Q$ where Q is the β_{eI}-nf of P.
- Assume the property holds for $n \geq 0$.
  As $(A_1)\dots(A_n)(A_{n+1})(A)[x]P$ has β_{eI}-nf, then it is β_{eI}-SN and so $(A_2)\dots(A_n)(A_{n+1})(A)[x]P$ and A_1 have β_{eI}-nf. Call the β_{eI}-nf of A_1, A_1'. Now, by IH, $(B_1)\dots(B_j)(A')[x]Q$ is the β_{eI}-nf of $(A_2)\dots(A_n)(A_{n+1})(A)[x]P$ and $(B_1)\dots(B_j)Q$ is the β_{eI}-nf of $(A_2)\dots(A_n)(A_{n+1})P$.
  - If (A_1') is β_{eI}-bachelor in $(A_1')(B_1)\dots(B_j)(A')[x]Q$ (and hence it is bachelor in $(A_1')(B_1)\dots(B_j)Q$), then:
    $(A_1')(B_1)\dots(B_j)(A')[x]Q$ and $(A_1')(B_1)\dots(B_j)Q$ are the β_{eI}-nfs required.
  - If (A_1') is β_{eI}-partnered in $(A_1')(B_1)\dots(B_j)(A')[x]Q$ then all $(B_1),\dots(B_j)$ start β_{eK}-redexes and $Q \equiv [x_j]\dots[x_1][y]R$. Now, for B the β_{eI}-nf of $[y := A_1']R$ we have:
    $(A_1)\dots(A_n)(A_{n+1})(A)[x]P \twoheadrightarrow_{\beta_{eI}}$
    $(A_1')(B_1)\dots(B_j)(A')[x][x_j]\dots[x_1][y]R \rightarrow_{\beta_{eI}}$
    $(B_1)\dots(B_j)(A')[x][x_j]\dots[x_1][y := A_1']R \twoheadrightarrow_{\beta_{eI}}$

[3] Thanks for an anonymous referee who drew my attention to this point.

$(B_1)\dots(B_j)(A')[x][x_j]\dots[x_1]B$.
Moreover, $(A_1)\dots(A_n)(A_{n+1})P \twoheadrightarrow_{\beta_{eI}} (A'_1)(B_1)\dots(B_j)[x_j]\dots[x_1][y]R \rightarrow_{\beta_{eI}}$
$(B_1)\dots(B_j)[x_j]\dots[x_1][y := A'_1]R \twoheadrightarrow_{\beta_{eI}} (B_1)\dots(B_j)[x_j]\dots[x_1]B$. Now, we are done (note that $B_1,\dots B_j$ start β_{eK}-redexes).

Lemma 30 If $(A_1)\dots(A_n)P$ and A have β_{eI}-nf, $x \notin FV((A_1)\dots(A_n)(A)P)$, then: $(A_1)\dots(A_n)(A)[x]P$ has β_{eI}-nf.

Proof By induction on $n \geq 0$.

- Case $n = 0$, P and A have P' and A' as β_{eI}-nfs, then $(A)[x]P$ has $(A')[x]P'$ as β_{eI}-nf.
- Assume the property holds for $n \geq 0$. Let $(A_1)\dots(A_n)(A_{n+1})P$ have β_{eI}-nf, hence it is β_{eI}-SN and so $(A_2)\dots(A_{n+1})P$ has β_{eI}-nf and A_1 has A'_1 as β_{eI}-nf. By IH, $(A_2)\dots(A_{n+1})(A)[x]P$ has β_{eI}-nf which is by Lemma 29, $M \equiv (B_1)\dots(B_j)(A')[x]Q$ and $(A_2)\dots(A_{n+1})P$ has $(B_1)\dots(B_j)Q$ as its β_{eI}-nf.
  - If (A'_1) is β_{eI}-bachelor in $(A'_1)M$ then $(A'_1)M$ is the β_{eI}-nf of $(A_1)\dots(A_{n+1})(A)[x]P$.
  - If (A'_1) is β_{eI}-partnered in $(A'_1)M$ then $Q \equiv [x_j]\dots[x_1][y]R$. Now,
    $(A_1)\dots(A_n)(A_{n+1})P \twoheadrightarrow_{\beta_{eI}} (A'_1)(B_1)\dots(B_j)[x_j]\dots[x_1][y]R \rightarrow_{\beta_{eI}}$
    $(B_1)\dots(B_j)[x_j]\dots[x_1][y := A'_1]R$.
    Hence $[y := A'_1]R$ is β_{eI}-SN as $(A_1)\dots(A_{n+1})P$ is.
    Let B be the β_{eI}-nf of $[y := A'_1]R$. Now,
    $(A_1)\dots(A_{n+1})(A)[x]P \twoheadrightarrow_{\beta_{eI}} (A'_1)(B_1)\dots(B_j)(A')[x][x_j]\dots[x_1][y]R \rightarrow_{\beta_{eI}}$
    $(B_1)\dots(B_j)(A')[x][x_j]\dots[x_1][y := A'_1]R \rightarrow_{\beta_{eI}}$
    $(B_1)\dots(B_j)(A')[x][x_j]\dots[x_1]B$ which is in β_{eI}-nf.

Lemma 31 If $A_i \rightarrow_{\beta_K} B_i$, $(A_1)\dots(A_{i-1})(B_i)(A_{i+1})\dots(A_n)z$ and A_i have β_{eI}-nf then $(A_1)\dots(A_{i-1})(A_i)(A_{i+1})\dots(A_n)z$ has β_{eI}-nf.

Proof $A_1,\dots A_{i-1}, A_i, A_{i+1},\dots A_n$ all have β_{eI}-nf, $A'_1,\dots A'_{i-1}, A'_i, A'_{i+1},\dots A'_n$. Hence, $(A_1)\dots(A_{i-1})(A_i)(A_{i+1})\dots(A_n)z \twoheadrightarrow_{\beta_{eI}} (A'_1)\dots(A'_{i-1})(A'_i)(A'_{i+1})\dots(A'_n)z$ which is in β_{eI}-nf.

Lemma 32 If $M \rightarrow_F N$ using a β_K-redex, and N has a β_{eI}-nf, then M has β_{eI}-nf.

Proof By induction on the depth of the F-redex (following Lemma 8).

- If $M \equiv [x_1][x_2]\dots[x_m](A_1)(A_2)\dots(A_n)(A)[x]P$, where $x \notin FV(P)$, A is in β-nf and $m \geq 0$, and if $N \equiv [x_1][x_2]\dots[x_m](A_1)(A_2)\dots(A_n)P$ where $n \geq 0$, use Lemma 30 (A in β-nf $\Rightarrow$ A in β_{eI}-nf).
- Let $S \equiv [x_1][x_2]\dots[x_m](A_1)(A_2)\dots(A_n)$. If $M \equiv S(A)[x]P$, $N \equiv S(A')[x]P$ where $x \notin FV(P)$, A is not in β-nf, $A \rightarrow_F A'$, $n \geq 0$ and $m \geq 0$. Use IH to deduce that A has β_{eI}-nf. As N has β_{eI}-nf, then $[x_1][x_2]\dots[x_m](A_1)(A_2)\dots(A_n)P$ has β_{eI}-nf by Lemma 29. Hence, $[x_1][x_2]\dots[x_m](A_1)(A_2)\dots(A_n)(A)[x]P$ has β_{eI}-nf by Lemma 30.

- If $M \equiv [x_1][x_2]\dots[x_m](A_1)(A_2)\dots(A_n)(A)(B_1)(B_2)\dots(B_r)z$,
  $N \equiv [x_1][x_2]\dots[x_m](A_1)(A_2)\dots(A_n)(A')(B_1)(B_2)\dots(B_r)z$ where
  A is not in β-nf, $A \to_F A'$, $n \geq 0$, $m \geq 0$ and $r \geq 0$ and $\forall i, 1 \leq i \leq r, B_i$ is in β-nf. As N has β_{eI}-nf, so does A' and by IH, so does A. By Lemma 31, M has β_{eI}-nf.

Now, here is the key theorem to PSN:

Theorem 33 (The inverse of $\to_F$ preserves β_{eI}-N)
If $M \twoheadrightarrow_F N$ and N is β_{eI}-N, then M is β_{eI}-N.

Proof We show it for $M \to_F N$. Note that if $M \to_F N$ and $F(M)$ is a β_I-redex, then the theorem is obvious as a β_I-redex is a β_{eI}-redex. Hence, we only need to prove the theorem for the case when $F(M)$ is a β_K-redex. But this is already done in Lemma 32.

Finally, here is the PSN result.

Corollary 34 (Preservation of Strong Normalisation)
If M is β-SN then M is β_e-SN.

Proof As M is β-SN, the perpetual strategy of M terminates. Let $M \twoheadrightarrow_F N$ where N is in β-nf. As N has no β-redexes, N is β_{eI}-N. Hence, by Theorem 33, M is β_{eI}-N. So, by Theorem 28 M is β_e-SN.

8 Conclusion

In this paper, we established that there is indeed a reduction relation which satisfies both postponement of K-contractions and conservation. This reduction relation is a generalisation of the ordinary β-reduction and has been extensively used since '73 for theoretical and practical reasons (see Section 2). We showed moreover that this generalised reduction (called β_e) is indeed a desirable generalisation of β-reduction by showing that β_e preserves strong normalisation in the sense that if M is β-SN then M is β_e-SN. Preservation of Strong Normalisation (PSN) is a property that has to be established for any extension of a reduction relation in the sense that: if a term is strongly normalising for a reduction relation, then it must remain strongly normalising for its extension. For example, a lot of research has been carried out lately to establish PSN for β-reduction extended with explicit substitution (see [4], [9] and [18]). The results of this paper establish that β_e is indeed a safe extension of β.

It is worth noting that we used item notation in this paper in order to reach the results desired. There is a reason for this. In the usual notation, generalised redexes are not easily visible whereas they are in item notation (see [8]). For example, in $()()[]()[][]$, we can clearly see that the leftmost $()$ matches the rightmost$[]$. Using item notation enables us to write the proofs clearly. Compare with [6] who used a more restricted generalised reduction but it was still hard to discuss where generalised redexes occurs in a term. For more information on the simplicity and usefulness of item notation, the reader is referred to [8]. It should be noted moreover, that using

item notation is not restrictive and that the results of this paper would still hold if we used the classical notation. Only the proofs will be cumbersome to write as the classical notation cannot easily enable us to express generalised redexes.

The following is an itemised summary of this paper:

1. Postponement and conservation are shown for the same reduction relation β_e.
2. This reduction relation is a generalisation of that presented in de Groote's article and of many of the existing generalisations of β-reduction. Because of this, it enables greater flexibility in the ordering of evaluation.
3. The fact that this reduction relation preserves strong normalisation and enables the postponement of some work allowing unnecessary K-redexes to be bypassed, means that one can investigate a programming language evaluation strategy based on this reduction together with explicit substitutions (which allow bypassing any work inside a subterm that will be discarded later). This means that one can combine the advantages of the complementary ways of postponing work due to both explicit substitutions and generalised reduction. Hence, one can achieve an even greater flexibility in the order of reduction and evaluation, something very welcome in the implementation and compilation of programming languages.
4. The syntax of this paper may well be the answer to the existence of a syntax that realises Regnier's σ-reduction. We leave this for future explorations.
5. It is our belief that automating proofs written in this fashion may be more efficient than automating the proofs written using σ-equivalence (which was said to need a good syntax to describe it in [22]).

Acknowledgements

I am grateful to Joe Wells and the anonymous referees for for their comments on the paper. This work is supported by EPSRC grants number GR/L15685 and GR/L36963. Anonymous referees provided useful comments for which I am grateful.

References

[1] Z.M. Ariola, M. Felleisen, J. Maraist, M. Odersky and P. Wadler. A call by need lambda calculus. *ACM Symposium on Principles of Programming Languages*, 1995.

[2] H. Barendregt. *Lambda Calculus: its Syntax and Semantics.* North-Holland, 1984.

[3] H. Barendregt. Lambda calculi with types. *Handbook of Logic in Computer Science*, volume II, ed. Abramsky S., Gabbay D.M., Maibaum T.S.E., Oxford University Press, 1992.

[4] Z. Benaissa, D. Briaud, P. Lescanne and J. Rouyer-Degli. T $\lambda\upsilon$, a calculus of explicit substitutions which preserves strong normalisation. *Functional Programming*, 6(5):699-722, September 1996.

[5] R. Bloo, F. Kamareddine and R. Nederpelt. The Barendregt Cube with Definitions and Generalised Reduction. *Information and Computation 126(2)*, 123-143, 1996.

[6] P. de Groote. The conservation theorem revisited. *International Conference on Typed Lambda Calculi and Applications. Lecture Notes in Computer Science LNCS 664*, 163-178, Springer-Verlag, 1993.

[7] F. Kamareddine and R.P. Nederpelt. Generalising reduction in the λ-calculus. *Journal of Functional Programming 5 (4)*, 637-651, 1995.

[8] F. Kamareddine and R.P. Nederpelt. A useful λ-notation. *Theoretical Computer Science 155*, 85-109, 1996.

[9] F. Kamareddine and A. Ríos, λ-calculus à la de Bruijn & explicit substitution. *Lecture Notes in Computer Science 982*, 7th international symposium on Programming Languages: Implementations, Logics and Programs, PLILP '95, 45-62, Springer-Verlag, 1995.

[10] F. Kamareddine and A. Ríos, Generalised β_e-reduction and explicit substitution. *Lecture Notes in Computer Science 1140*, 8th international symposium on Programming Languages: Implementations, Logics and Programs, PLILP '96, 378-392, Springer-Verlag, 1996.

[11] F. Kamareddine, A. Ríos, and J.B. Wells. Calculi of Generalised β-Reduction and Explicit Substitutions: The Type free and Simply Typed Versions. *Journal of Functional and Logic Programming*, Volume 1998, ISSN 1080-5230, MIT Press

[12] A.J. Kfoury, J. Tiuryn and P. Urzyczyn. An analysis of ML typability. Journal of the ACM 41(2), 368-398, 1994.

[13] A.J. Kfoury and J.B. Wells. A direct algorithm for type inference in the rank-2 fragment of the second order λ-calculus. *Proceedings of the 1994 ACM Conference on LISP in Functional Programming*, 1994.

[14] A.J. Kfoury and J.B. Wells. New notions of reductions and non-semantic proofs of β-strong normalisation in typed λ-calculi. *IEEE Logic In Computer Science*, 1995.

[15] A.J. Kfoury and J.B. Wells. Addendum to new notions of reduction and non-semantic proofs of β-strong normalisation in typed λ-calculi. Technical report, Boston University.

[16] J.W. Klop. *Combinatory Reduction Systems.* Number 127 in Mathematical Centre Tracts. Mathematisch Centrum, Amsterdam, 1980.

[17] J.L. Krivine. *Lambda-calcul, types et modèles.* Masson, 1990.

[18] C. Muñoz. Confluence and preservation of strong normalisation in an explicit substitution calculus. Rapport de Recherche No 2762, INRIA.

[19] R.P. Nederpelt. *Strong normalisation in a typed lambda calculus with lambda structured types*. Ph.D. thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, 1973. Also appears in [20].

[20] R.P. Nederpelt, J.H. Geuvers and R.C. de Vrijer, eds., *Selected Papers on Automath*. North Holland, 1994.

[21] L. Regnier. *Lambda calcul et réseaux*. Thèse de doctorat de l'université Paris 7, 1992.

[22] L. Regnier. Une équivalence sur les lambda termes. *Theoretical Computer Science 126*, 281-292, 1994.

[23] A. Sabry, and M. Felleisen. Reasoning about programs in continuation-passing style. *Proc. 1992 ACM Conf. LISP Funct. Program.*, 288-298, 1992.

[24] M.H. Sørensen. Strong normalization from weak normalization in typed λ-calculi. *Journal of Information and Comuptation 133(1)*, 35-71, 1997.

[25] D. Vidal. *Nouvelles notions de réduction en lambda calcul*. Thèse de doctorat, Université de Nancy 1, 1989.

[26] H. Xi. On weak and strong normalizations. Technical Report 96-187, Carnegie Mellon University, 1996.