# On the reduction of the type-free computational $\lambda$-calculus

Ugo de'Liguoro and Riccardo Treglia

Università di Torino, Turin, Italy
ugo.deliguoro@unito.it
riccardo.treglia@unito.it

**Abstract**

We study the reduction of the computational $\lambda$-calculus in the untyped case. To this aim, we consider a minimal fragment of the $\lambda$-calculus with monads as introduced by Wadler, and define a notion of call-by-value reduction just by orienting the three monad equational laws. We then prove confluence of its compatible closure. Finally, we show factorization of any reduction sequence into essential and inessential steps.

## 1 Introduction

The computational $\lambda$-calculus, called $\lambda_c$, was introduced by Moggi [Mog89, Mog91] as a meta-language to describe non functional effects in programming languages via an incremental approach. Much as for ordinary $\lambda$-calculus, the equational theory of $\lambda_c$ can be modelled by the convertibility relation induced by a reduction relation. Building the reduction theory of $\lambda_c$ is however quite challenging. A first attempt is in §6 of [Mog89], where the defined notion of reduction consists of six rules plus $\eta$. Proving confluence of this reduction relation revealed to be quite hard; it was studied in the context of call-by-need calculi, e.g. in [MOTW99, AFM$^+$95] obtaining partial results, but a full proof has been achieved only recently in [Ham18].

Aiming at a logical analysis of the semantics of the untyped $\lambda_c$ in terms of an intersection type assignment system, we proposed in [dT19] a simplified syntax, which is derived from Wadler's $\lambda$-calculus with monads, and defined reduction just by orientating the three monad laws in [Wad92, Wad95]. We dub $\lambda_c^u$ our calculus, and $\longrightarrow_{\lambda\mathbf{C}}$ the reduction relation. This is the content of section 2 of the present note.

Although one can translate Moggi's syntax into ours, preserving and reflecting the respective reduction relations, the inverse translation just preserves conversion, so that confluence in our calculus cannot rest on the same property of the original $\lambda_c$, and the proof had to be reworked anew. We sketch the proof from [dT19] in section 3.

Confluence is not the only fundamental property of reduction in $\lambda$-calculi; further examples are standardization and the existence of normalizing strategies. Toward the study of these properties in the case of $\lambda_c^u$ and $\longrightarrow_{\lambda\mathbf{C}}$, we explore here in section 4 factorization for our calculus by adapting results in [AFG19].

While the confluence proof is included in a revised version of [dT19] and has been submitted for publication, the factorization results are new.

## 2 Untyped $\lambda_c$-calculus

The syntax of the untyped computational $\lambda$-calculus, shortly $\lambda_c^u$, and its reduction relation as introduced in [dT19], are reported below:

**Definition 2.1** (Terms of $\lambda_c^u$)**.** *The terms of the* untyped computational $\lambda$-calculus*, shortly $\lambda_c^u$, consist of two sorts of expressions:*

$$\begin{array}{llcll} Val: & V, W & ::= & x \mid \lambda x.M & \textit{(values)} \\ Com: & M, N & ::= & unit\, V \mid M \star V & \textit{(computations)} \end{array}$$

*where $x$ ranges over a denumerable set Var of variables. We set $Term = Val \cup Com$; $FV(V)$ and $FV(M)$ are the sets of free variables occurring in $V$ and $M$ respectively, and are defined in the obvious way. Terms are identified up to clash avoiding renaming of bound variables ($\alpha$-congruence).*

With respect to Moggi's $\lambda_c$-syntax, we do not have the *let* construct, which is considered as syntactical sugar for bind and abstraction:

$$let\ x = N\ in\ M \;\equiv\; N \star \lambda x.M$$

Notably we do not have application in the syntax, since it is definable (see below).

**Definition 2.2** (Reduction)**.** *Define the following reduction relation $\mapsto_{\lambda C} = \mapsto_{\beta_c} \cup \mapsto_{id} \cup \mapsto_{ass}$ over Com by:*

$$\begin{array}{rrcll} \beta_c) & unit\, V \star (\lambda x.M) & \mapsto & M[V/x] & \\ id) & M \star \lambda x.unit\, x & \mapsto & M & \\ ass) & (L \star \lambda x.M) \star \lambda y.N & \mapsto & L \star \lambda x.(M \star \lambda y.N) & \textit{for } x \notin FV(N) \end{array}$$

*where $M[V/x]$ denotes the capture avoiding substitution of $V$ for $x$ in $M$. Finally define the relation $\longrightarrow_{\lambda C}$ as the compatible closure of $\mapsto_{\lambda C}$.*

Rule $\beta_c$ is reminiscent of the left unit law in [Wad95]; we call it $\beta_c$ because it performs call-by-value $\beta$-contraction in $\lambda_c^u$. In fact, by reading $\star$ as postfix functional application and merging $V$ into its trivial computation *unit* $V$, $\beta_c$ is the same as $\beta_v$ in [Plo75]. Now, let $V, W \in Val$ and $M, N \in Com$; then define:

$$\begin{array}{lcllcl} VW & \equiv & unit\, W \star V & MV & \equiv & M \star (\lambda z.unit\, V \star z) \\ VN & \equiv & N \star V & MN & \equiv & M \star (\lambda z.N \star z) \end{array}$$

where $z$ is fresh. Then it is easy to see that, if $M \xrightarrow{*}_{\lambda C} unit\ (\lambda x.M')$ and $N \xrightarrow{*}_{\lambda C} unit\ V$ then $MN \xrightarrow{*}_{\lambda C} M'[V/x]$.

# 3 Confluence

Following a strategy used in case of call-by-need calculi with the *let* construct (see [AFM+95, MOTW99]), and more recently with the variant of call-by-value $\lambda$-calculus in [CG14], we split the proof in three steps, proving confluence of $\beta_c \cup id$ and *ass* separatedly, eventually combining these results by means of the commutativity of these relations.

In the first step we adapt the parallel reduction method, originally due to Tait and Martin Löf, and further developed by Takahashi [Tak95]. See e.g. the book [Ter03] ch. 10. Let's define the following relation $\Rightarrow$:

**Definition 3.1.** *The relation $\Rightarrow \subseteq Term \times Term$ is inductively defined by:*

*i)* $x \twoheadrightarrow x$
*ii)* $M \twoheadrightarrow N \Rightarrow \lambda x.M \twoheadrightarrow \lambda x.N$
*iii)* $V \twoheadrightarrow V' \Rightarrow \mathit{unit}\, V \twoheadrightarrow \mathit{unit}\, V'$
*iv)* $M \twoheadrightarrow M'$ *and* $V \twoheadrightarrow V' \Rightarrow M \star V \twoheadrightarrow M' \star V'$
*v)* $M \twoheadrightarrow M'$ *and* $V \twoheadrightarrow V' \Rightarrow \mathit{unit}\, V \star \lambda x.M \twoheadrightarrow M'[V'/x]$
*vi)* $M \twoheadrightarrow M' \Rightarrow M \star \lambda x.\mathit{unit}\, x \twoheadrightarrow M'$

By i) - iv) above, relation $\twoheadrightarrow$ is reflexive and coincides with its compatible closure. Also $\longrightarrow_{\beta_c,id} \subseteq \twoheadrightarrow$; intentionally, this is not the case w.r.t. the whole $\longrightarrow_{\lambda\mathbf{C}}$. Now, by means of Lemma 3.2 one easily proves that $\twoheadrightarrow \subseteq \overset{*}{\longrightarrow}_{\beta_c,id}$.

**Lemma 3.2.** *For $M, M' \in \mathit{Com}$ and $V, V' \in \mathit{Val}$ and every variable $x$, if $M \twoheadrightarrow M'$ and $V \twoheadrightarrow V'$, then $M[V/x] \twoheadrightarrow M'[V'/x]$.*

The next step in the proof is to show that the relation $\twoheadrightarrow$ satisfies the *triangle property*:

$$TP: \qquad \forall P\, \exists P^*\, \forall Q.\ P \twoheadrightarrow Q \ \Rightarrow\ Q \twoheadrightarrow P^*$$

where $P, P^*, Q \in \mathit{Term}$. $TP$ implies the *diamond property*, which for $\twoheadrightarrow$ is:

$$DP: \qquad \forall P, Q, R.\ P \twoheadrightarrow Q \ \&\ P \twoheadrightarrow R \ \Rightarrow\ \exists P'.\, Q \twoheadrightarrow P' \ \&\ R \twoheadrightarrow P'$$

In fact, if $TP$ holds then we can take $P' \equiv P^*$ in $DP$, since the latter only depends on $P$. We then define $P^*$ in terms of $P$ as follows:

i) $x^* \equiv x$
ii) $(\lambda x.M)^* \equiv \lambda x.M^*$
iii) $(\mathit{unit}\, V)^* \equiv \mathit{unit}\, V^*$
iv) $(\mathit{unit}\, V \star \lambda x.M)^* \equiv M^*[V^*/x]$
v) $(M \star \lambda x.\mathit{unit}\, x)^* \equiv M^*$, if $M \not\equiv \mathit{unit}\, V$ for $V \in \mathit{Val}$
vi) $(M \star V)^* \equiv M^* \star V^*$, $M \not\equiv \mathit{unit}\, W$ for $W \in \mathit{Val}$ and $V \not\equiv \lambda x.\mathit{unit}\, x$

**Lemma 3.3.** *For all $P, Q \in \mathit{Term}$, if $P \twoheadrightarrow Q$ then $Q \twoheadrightarrow P^*$, namely $\twoheadrightarrow$ satisfies $TP$.*

According to [Bar84], Def. 3.1.11, a notion of reduction $R$ is said to be *confluent* or *Church-Rosser*, shortly $CR$, if $\overset{*}{\longrightarrow}_R$ satisfies $DP$; more explicitly for all $M, N, L \in \mathit{Com}$:

$$M \overset{*}{\longrightarrow}_R N \ \&\ M \overset{*}{\longrightarrow}_R L \Rightarrow \exists M' \in \mathit{Com}.\, N \overset{*}{\longrightarrow}_R M' \ \&\ L \overset{*}{\longrightarrow}_R M'$$

**Corollary 3.4.** *The notion of reduction $\beta_c \cup id$ is $CR$.*

To prove confluence of *ass* we use Newman Lemma (see [Bar84], Prop. 3.1.24). A notion of reduction $R$ is *weakly Church-Rosser*, shortly $WCR$, if for all $M, N, L \in \mathit{Com}$:

$$M \longrightarrow_R N \ \&\ M \longrightarrow_R L \Rightarrow \exists M' \in \mathit{Com}.\, N \overset{*}{\longrightarrow}_R M' \ \&\ L \overset{*}{\longrightarrow}_R M'$$

**Lemma 3.5.** *The notion of reduction ass is $WCR$.*

Recall that a notion of reduction $R$ is *strongly normalizing*, shortly $SN$, if there exists no infinite reduction $M \longrightarrow_R M_1 \longrightarrow_R M_2 \longrightarrow_R \cdots$ out of any $M \in \mathit{Com}$.

**Lemma 3.6.** *The notion of reduction ass is $SN$.*

**Corollary 3.7.** *The notion of reduction ass is $CR$.*

*Proof.* By Lem. 3.5, 3.6 and by Newman Lemma, stating that a notion of reduction which is *WCR* and *SN* is *CR*. □

Finally we show that $\longrightarrow_{\beta_c,id}$ and $\longrightarrow_{ass}$ commute. The following definitions are from [BN98], Def. 2.7.9. Relations $\longrightarrow_1$ and $\longrightarrow_2$ *strongly commute* if, for all $M, N, L$: $N\ {}_1\!\!\longleftarrow M \longrightarrow_2 L \Rightarrow \exists P.\ N \xrightarrow{=}_2 P\ {}_1\!\!\xleftarrow{*} L$ where $\xrightarrow{=}_2$ is $\longrightarrow_2 \cup =$, namely at most one reduction step.

**Lemma 3.8.** *Reductions $\longrightarrow_{\beta_c,id}$ and $\longrightarrow_{ass}$ strongly commute, then commute.*

*Proof.* By Lemma 2.7.11 in [BN98], two strongly commuting relations commute, and commutativity is clearly symmetric; hence it suffices to show that

$$N\ {}_{\beta_c,id}\!\!\longleftarrow M \longrightarrow_{ass} L \Rightarrow \exists P \in \mathit{Com}.\ N \xrightarrow{=}_{ass} P\ {}_{\beta_c,id}\!\!\xleftarrow{*} L.$$

We can limit the cases to the critical pairs. For a full development see [dT19]. □

By the commutative union lemma (see [BN98], Lem. 2.7.10 and [Bar84], Prop. 3.3.5), if $\longrightarrow_{\beta_c,id}$ and $\longrightarrow_{ass}$ and are both *CR* (Cor. 3.4 and 3.7), and commute (Lem. 3.8) follows:

**Theorem 3.9** (Confluence)**.** *The notion of reduction $\lambda\boldsymbol{C}$ is CR.*

# 4 Factorization

Specializing the definition of factorization in [AFG19], we say that an abstract reduction system $(\mathit{Term}, \longrightarrow)$ *factorizes* via $\longrightarrow_e, \longrightarrow_{\neg e}$ if $\longrightarrow\ =\ \longrightarrow_e \cup \longrightarrow_{\neg e}$ and for all $M, N \in \mathit{Term}$, $M \xrightarrow{*} N$ implies that there exists $L \in \mathit{Term}$ such that $M \xrightarrow{*}_e L \xrightarrow{*}_{\neg e} N$. We abbreviate the last condition by $M \xrightarrow{*}_e \cdot \xrightarrow{*}_{\neg e} N$.

Now, we take $\longrightarrow\ =\ \longrightarrow_{\lambda\mathbf{C}}$ and construct the relations $\longrightarrow_e, \longrightarrow_{\neg e}$, called the *essential* and *inessential* in [AFG19], by closing $\mapsto_{\lambda\mathbf{C}}$ under two sorts of contexts:

$$\begin{array}{rrcl} \textit{Inessential contexts:} & \neg\mathcal{E} & ::= & \langle\cdot_{\mathsf{C}}\rangle \mid \mathit{unit}\,\lambda x.\neg\mathcal{E} \mid M \star \lambda x.\neg\mathcal{E} \mid \neg\mathcal{E} \star V \\ \textit{Essential contexts:} & \mathcal{E} & ::= & \langle\cdot_{\mathsf{C}}\rangle \mid \mathcal{E} \star V \end{array}$$

where the hole $\langle\cdot_{\mathsf{C}}\rangle$ can be filled by terms in *Com* only. Then $\longrightarrow_e$ and $\longrightarrow_{\neg e}$ are the least relations including $\mapsto_{\lambda\mathbf{C}}$ such that for all $M, N \in \mathit{Com}$, essential context $\mathcal{E}$ and inessential context $\neg\mathcal{E}$ it holds:

$$M \mapsto_{\lambda\mathbf{C}} N \Longrightarrow \mathcal{E}\langle M\rangle \longrightarrow_e \mathcal{E}\langle N\rangle \qquad \text{and} \qquad M \mapsto_{\lambda\mathbf{C}} N \Longrightarrow \neg\mathcal{E}\langle M\rangle \longrightarrow_{\neg e} \neg\mathcal{E}\langle N\rangle$$

We highlight that relations $\longrightarrow_e$ and $\longrightarrow_{\neg e}$ are actually not disjoint, as essential steps are also inessential.

The factorization property ensures that any finite reduction can be re-arranged into an essential reduction followed by some inessential steps. In our case, this corresponds to a weak head reduction, with the twist that in a bind expression the argument appears to the left of the function.

The key of the proof of the Factorization Theorem 4.4 is the construction of two further auxiliary relations $\Rightarrow_{\neg e}$ and $\Rightarrow_{\lambda\mathbf{C}}$, such that the conditions in Proposition 4.3 hold.

**Definition 4.1** (Inessential parallel reduction)**.** *The relation $\Rightarrow_{\neg e} \subseteq \mathit{Term} \times \mathit{Term}$ is inductively defined by:*

*i)* $x \Rightarrow_{\neg e} x$
*ii)* $M \Rightarrow_{\lambda C} N \Rightarrow \lambda x.M \Rightarrow_{\neg e} \lambda x.N$
*iii)* $V \Rightarrow_{\lambda C} V' \Rightarrow \mathit{unit}\, V \Rightarrow_{\neg e} \mathit{unit}\, V'$
*iv)* $M \Rightarrow_{\neg e} M'$ *and* $V \Rightarrow_{\neg e} V' \Rightarrow M \star V \Rightarrow_{\neg e} M' \star V'$
*v)* $L \Rightarrow_{\neg e} L'$ *and* $M \Rightarrow_{\lambda C} M'$ *and* $N \Rightarrow_{\lambda C} N' \Rightarrow (L \star \lambda x.M) \star \lambda y.N \Rightarrow_{\neg e} L' \star \lambda x.(M' \star \lambda y.N')$

**Definition 4.2** (Indexed parallel reduction)**.** *The relation* $\stackrel{n}{\Longrightarrow} \subseteq \mathit{Term} \times \mathit{Term}$ *is inductively defined by:*

*i)* $x \stackrel{0}{\Longrightarrow} x$

*ii)* $M \stackrel{n}{\Longrightarrow} N \Rightarrow \lambda x.M \stackrel{n}{\Longrightarrow} \lambda x.N$

*iii)* $V \stackrel{n}{\Longrightarrow} V' \Rightarrow \mathit{unit}\, V \stackrel{n}{\Longrightarrow} \mathit{unit}\, V'$

*iv)* $M \stackrel{n}{\Longrightarrow} M'$ *and* $V \stackrel{m}{\Longrightarrow} V' \Rightarrow M \star V \stackrel{n+m}{\Longrightarrow} M' \star V'$

*v)* $M \stackrel{n}{\Longrightarrow} M'$ *and* $V \stackrel{m}{\Longrightarrow} V' \Rightarrow \mathit{unit}\, V \star \lambda x.M \xRightarrow{n+|M'|_x \cdot m+1} M'[V'/x]$

*vi)* $M \stackrel{n}{\Longrightarrow} M' \Rightarrow M \star \lambda x.\mathit{unit}\, x \stackrel{n}{\Longrightarrow} M'$

*vii)* $L \stackrel{n}{\Longrightarrow} L'$ *and* $M \stackrel{m}{\Longrightarrow} M'$ *and* $N \stackrel{p}{\Longrightarrow} N' \Rightarrow (L \star \lambda x.M) \star \lambda y.N \xRightarrow{n+m+p} L' \star \lambda x.(M' \star \lambda y.N')$

*where* $|M|_x$ *is the number of free occurrences of* $x$ *in* $M$*.*

Note that $\stackrel{0}{\Longrightarrow}$ is the identity relation on *Term*, $\stackrel{1}{\Longrightarrow}$ is $\longrightarrow_{\lambda\mathbf{C}}$ defined in 2.2, and $\stackrel{n}{\Longrightarrow} \subseteq \longrightarrow^n$. Define $\Rightarrow_{\lambda\mathbf{C}} := \cup_{n\in\mathbb{N}} \stackrel{n}{\Longrightarrow}$. Observe that the above definition is essentially the same as that one of $\multimap\!\!\!\to$ in Def. 3.1, but for clause vii): adding the latter to $\multimap\!\!\!\to$ would break property *DP*, that indeed is not satisfied by $\Rightarrow_{\lambda\mathbf{C}}$.

An abstract reduction system that satisfies the following conditions is called a *macro-step system* in [AFG19].

**Proposition 4.3** ($\lambda\mathbf{C}$ Macro-step system)**.**
*i)* Merge*: if* $M \Rightarrow_{\neg e} \cdot \longrightarrow_e M'$ *then* $M \Rightarrow_{\lambda C} M'$
*ii)* Indexed split*: if* $M \stackrel{n}{\Longrightarrow} M'$*, then* $M \Rightarrow_{\neg e} M'$*, or* $n > 0$ *and* $M \longrightarrow_e \cdot \xRightarrow{n-1} M'$
*iii)* Split*: If* $M \Rightarrow_{\lambda C} M'$*, then* $M \stackrel{*}{\longrightarrow}_e \cdot \Rightarrow_{\neg e} M'$*.*

Once we have established that $(\mathit{Term}, \longrightarrow_e \cup \longrightarrow_{\neg e})$ is a macro-step system with respect to $\Rightarrow_{\lambda\mathbf{C}}$ and $\Rightarrow_{\neg e}$. Since in [AFG19] is proved that every Macro-step system satisfies factorization, we have the following theorem.

**Theorem 4.4** (Factorization)**.** *The reduction system* $(\mathit{Term}, \longrightarrow_{\lambda C})$ *factorizes via* $\longrightarrow_e, \longrightarrow_{\neg e}$ *namely*

$$M \longrightarrow_{\lambda C} M' \Rightarrow M \stackrel{*}{\longrightarrow}_e \cdot \stackrel{*}{\longrightarrow}_{\neg e} M'$$

# References

[AFG19] Beniamino Accattoli, Claudia Faggian, and Giulio Guerrieri. Factorization and normalization, essentially. In *APLAS 2019: Programming Languages and Systems*, volume 11893 of *Lecture Notes in Computer Science*, page 159–180. Springer Verlag, 12 2019.

[AFM+95] Zena M. Ariola, Matthias Felleisen, John Maraist, Martin Odersky, and Philip Wadler. The call-by-need lambda calculus. In Ron K. Cytron and Peter Lee, editors, *Conference Record of POPL'95: 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Francisco, California, USA, January 23-25, 1995*, pages 233–246. ACM Press, 1995.

[Bar84] H. Barendregt. *The Lambda Calculus: its Syntax and Semantics*. North-Holland, Amsterdam, revised edition, 1984.

[BN98] Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998.

[CG14] Alberto Carraro and Giulio Guerrieri. A semantical and operational account of call-by-value solvability. In Anca Muscholl, editor, *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS 2014*, volume 8412 of *Lecture Notes in Computer Science*, pages 103–118. Springer, 2014.

[dT19] Ugo de'Liguoro and Riccardo Treglia. Intersection types for the computational lambda-calculus. *CoRR*, abs/1907.05706, 2019.

[Ham18] M. Hamana. Polymorphic rewrite rules: Confluence, type inference, and instance validation. In *Functional and Logic Programming - 14th International Symposium, FLOPS 2018, Nagoya, Japan, May 9-11, 2018, Proceedings*, volume 10818 of *Lecture Notes in Computer Science*, pages 99–115, 2018.

[Mog89] E. Moggi. Computational Lambda-calculus and Monads. In *Proceedings of Logic in Computer Science (LICS)*, pages 14–23, 1989.

[Mog91] E. Moggi. Notions of Computation and Monads. *Information and Computation*, 93:55–92, 1991.

[MOTW99] John Maraist, Martin Odersky, David N. Turner, and Philip Wadler. Call-by-name, call-by-value, call-by-need and the linear lambda calculus. *Theor. Comput. Sci.*, 228(1-2):175–210, 1999.

[Plo75] G. D. Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theoretical Computer Science*, 1:125–159, 1975.

[Tak95] M. Takahashi. Parallel reduction in lambda-calculus. *Information and Computation*, 118:120–127, 1995.

[Ter03] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[Wad92] P. Wadler. The essence of functional programming. In *Conference Record of the Nineteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Albuquerque, New Mexico, USA, January 19-22, 1992*, pages 1–14, 1992.

[Wad95] P. Wadler. Monads for Functional Programming. In *Advanced Functional Programming, First International Spring School on Advanced Functional Programming Techniques-Tutorial Text*, volume 925 of *Lecture Notes in Computer Science*, pages 24–52. Springer-Verlag, 1995.

# A Proof of Factorization

**Lemma A.1.** *(Substitutivity of $\overset{n}{\Rightarrow}$) If $M \overset{n}{\Rightarrow} M'$ and $V \overset{m}{\Rightarrow} V'$ then $M[V/x] \overset{k}{\Rightarrow} M'[V'/x]$ where $k = n + |M'|_x \cdot m$.*

*Proof.* Proof by induction on the structure of $M$. We will show just notable cases:
*Application*: This case occurs when the last rule applied is

$$\frac{N \overset{n_N}{\Longrightarrow} N' \qquad W \overset{n_W}{\Longrightarrow} W'}{M = N \star W \overset{n_N+n_W}{\Longrightarrow} N' \star W' = M'}$$

by i.h. $N[V/x] \overset{k_1}{\Longrightarrow} N'[V'/x]$ where $k_1 = n_N + |N'|_x \cdot m$

and $W[V/x] \overset{k_2}{\Longrightarrow} W'[V'/x]$ where $k_2 = n_W + |W'|_x \cdot m$

then

$$\frac{N[V/x] \overset{k_1}{\Longrightarrow} N'[V'/x] \qquad W[V/x] \overset{k_2}{\Longrightarrow} W'[V'/x]}{M[V/x] = N[V/x] \star W[V/x] \overset{k}{\Rightarrow} N'[V'/x] \star W'[V'/x] = M'[V'/x]}$$

where $k = k_1+k_2 = n_N+|N'|_x \cdot m+n_W+|W'|_x \cdot m = n+|M'|_x \cdot m$, in fact $|M'|_x = |N'|_x+|W'|_x$.

*$\beta_c$−step*: This case occurs when the last step has the following shape:

$$\frac{W \overset{n_W}{\Longrightarrow} W' \qquad N \overset{n_N}{\Longrightarrow} N'}{M = \mathit{unit}\, W \star \lambda y.N \overset{n}{\Rightarrow} N'[W'/y] = M'}$$

where $n = n_N + |N'|_y \cdot n_W + 1$.
Assuming wlog $x \neq y$, $|M'|_x = |N'[W'/y]|_x = |N'|_x + |N'|_y \cdot |W'|_x$

$$M[V/x] = \mathit{unit}\, W[V/x] \star \lambda y.N[V/x]$$
$$M'[V'/x] = N'[V'/x][W'[V'/x]/y]$$

By i.h. $N[V/x] \overset{k_1}{\Longrightarrow} N'[V'/x]$ where $k_1 = n_N + |N'|_x \cdot m$

$W[V/x] \overset{k_2}{\Longrightarrow} W'[V'/x]$ where $K_2 = n_W + |W'|_x \cdot m$, then $M[V/x] \overset{k}{\Rightarrow} M'[V'/x]$ where

$$\begin{aligned} k &= k_1 + |N'|_y \cdot k_2 + 1 = \\ &= n_N + |N'|_x \cdot m + |N'|_y \cdot (n_W + |N'|_x \cdot m) + 1 = \\ &= n_N + |N'|_y \cdot n_W + 1 + |N'| \cdot m + |N'|_y \cdot |W'|_x \cdot m = \\ &= n + |M'|_x \cdot m \end{aligned}$$

*id−step*:

$$\frac{N \overset{n}{\Rightarrow} N'}{M = N \star \lambda y.\mathit{unit}\, y \overset{n}{\Rightarrow} N' = M'}$$

And $M[V/x] = N[V/x] \star \lambda y.\mathit{unit}\, y[V/x]$
By i. h. $N[V/x] \overset{k_1}{\Longrightarrow} N'[V'/x]$ where $k_1 = n + |N'|_x \cdot m$

$\lambda y.unit\, y[V/x] \stackrel{0}{\Rightarrow} \lambda y.unit\, y$.

$$\frac{N[V/x] \stackrel{k_1}{\Longrightarrow} N'[V'/x]}{M[V/x] = N[V/x] \star \lambda y.unit\, y[V/x] \stackrel{k}{\Rightarrow} N'[V'/x] = M'[V'/x]}$$

Then $k = k_1$.

*ass−step*: $M = (L \star \lambda y.N) \star \lambda z.P \stackrel{n}{\Rightarrow} L' \star \lambda y.(N' \star \lambda z.P)$ where $L \stackrel{n_L}{\Longrightarrow} L'$, $N \stackrel{n_N}{\Longrightarrow} N'$, $P \stackrel{n_P}{\Longrightarrow} P'$ and $n = N_l + n_N + n_P$.
by i.h.

$$L[V/x] \stackrel{k_1}{\Longrightarrow} L'[V'/x] \text{ where } k_1 = n_L + |L'|_x \cdot m$$
$$N[V/x] \stackrel{k_2}{\Longrightarrow} N'[V'/x] \text{ where } k_2 = n_N + |N'|_x \cdot m$$
$$P[V/x] \stackrel{k_3}{\Longrightarrow} P'[V'/x] \text{ where } k_3 = n_P + |P'|_x \cdot m$$

$$\frac{L[V/x] \stackrel{k_1}{\Longrightarrow} L'[V'/x] \qquad N[V/x] \stackrel{k_2}{\Longrightarrow} N'[V'/x] \qquad P[V/x] \stackrel{k_3}{\Longrightarrow} P'[V'/x]}{M[V/x] = (L[V/x] \star \lambda y.N[V/x]) \star \lambda z.P[V/x] \stackrel{k}{\Rightarrow} L'[V'/x] \star \lambda y.(N'[V'/x] \star \lambda z.P[V'/x])}$$

where $k = k_1 + k_2 + k_3 = n_L + |L'|_x \cdot m + n_N + |N'|_x \cdot m + n_P + |P'|_x \cdot m = n + |M'|_x \cdot m$. $\square$

**Proposition A.2** ($\lambda$**C** Macro-step system)**.**
1. Merge*: if $M \Rightarrow_{\neg e} \cdot \longrightarrow_e M'$ then $M \Rightarrow_{\lambda \mathbf{C}} M'$*
2. Indexed split*: if $M \stackrel{n}{\Rightarrow} M'$, then $M \Rightarrow_{\neg e} M'$, or $n > 0$ and $M \longrightarrow_e \cdot \stackrel{n-1}{\Longrightarrow} M'$*
3. Split*: If $M \Rightarrow_{\lambda \mathbf{C}} M'$, then $M \longrightarrow_e^* \cdot \Rightarrow_{\neg e} M'$.*

*Proof. 1.Merge:* by structural induction on $M \Rightarrow_{\neg e} N$.
Following hypothesis, since $N \longrightarrow_e M'$, $M'$ cannot be *unit* $V$ for any $V \in Val$, then there exists an essential context $\mathcal{E}$, computations $\bar{N}$, $\bar{M}'$, such that $N = \mathcal{E}\langle \bar{N} \rangle \to_e \mathcal{E}\langle \bar{M}' \rangle = M'$.
Hence $N = \bar{N} \star \bar{V} = \to_e \bar{M}' \star \bar{V}' = M'$ and $M \Rightarrow_{\neg e} N$ is derived as follows

$$\frac{N_0 \Rightarrow_{\neg e} \bar{N} \qquad V_0 \Rightarrow_{\neg e} \bar{V}}{M = N_0 \star V_0 \Rightarrow_{\neg e} \bar{N} \star \bar{V} = N}$$

- if $\bar{N} \longrightarrow_e \bar{M}'$ then $M' = \bar{M}' \star \bar{V}$.
  The i.h. gives $N_0 \Rightarrow_{\lambda \mathbf{C}} \bar{M}'$, and $M \Rightarrow_{\lambda \mathbf{C}} M'$ is derived as follows

$$\frac{N_0 \Rightarrow_{\lambda \mathbf{C}} \bar{M}' \qquad V_0 \Rightarrow_{\lambda \mathbf{C}} \bar{V}}{M = N_0 \star V_0 \Rightarrow_{\lambda \mathbf{C}} \bar{M}' \star \bar{V} = M'}$$

- if $N \mapsto_{id} M'$ this means that $\bar{N} = M'$ and $\bar{V} = \lambda x.unit\, x$, and $M \Rightarrow_{\lambda \mathbf{C}} M'$ is derived as follows (since $\Rightarrow_{\neg e} \subseteq \Rightarrow_{\lambda \mathbf{C}}$)

$$\frac{N_0 \Rightarrow_{\lambda \mathbf{C}} \bar{N}}{M = N_0 \star \lambda x.unit\, x \Rightarrow_{\lambda \mathbf{C}} \bar{N} = M'}$$

- if $N \mapsto_{\beta_c} M'$ then $\bar{V} = \lambda x.L$ and $\bar{N} = \mathit{unit}\, W'$.
  By definition of $\Rightarrow_{\neg e}$ the step $V_0 \Rightarrow_{\neg e} \bar{V}$ ha the form

$$\lambda x.L \Rightarrow_{\neg e} \lambda x.L' \text{ for some } L \text{ such that } L \Rightarrow_{\lambda\mathbf{C}} L'$$

  this means that $M \Rightarrow_{\lambda\mathbf{C}} M' = L[W'/x]$ following the next derivation

$$\frac{L \Rightarrow_{\lambda\mathbf{C}} L' \qquad W \Rightarrow_{\lambda\mathbf{C}} W'}{M = \mathit{unit}\, W \star \lambda x.L \Rightarrow_{\lambda\mathbf{C}} L'[W'/x] = M'}$$

- if $N \mapsto_{ass} M'$ then $N = \bar{N} \star \bar{V} = (P \star \lambda x.Q) \star \bar{V}$ and $M' = \bar{M}' \star \bar{V}' = P \star \lambda x.(Q \star \bar{V})$.
  Since $N_0 \Rightarrow_{\neg e} \bar{N} = P \star \lambda x.Q$, it follows that $N_0$ has the shape $N_0 = P_0 \star \lambda x.Q_0$ where $P_0 \Rightarrow_{\neg e} P$ and $Q_0 \Rightarrow_{\lambda\mathbf{C}} Q$, then

$$\frac{P_0 \Rightarrow_{\lambda\mathbf{C}} P \qquad Q_0 \Rightarrow_{\lambda\mathbf{C}} Q \qquad V_0 \Rightarrow_{\lambda\mathbf{C}} \bar{V}}{M = (P_0 \star \lambda x.Q_0) \star V_0 \Rightarrow_{\lambda\mathbf{C}} P \star \lambda x.(Q \star \bar{V}) = M'}$$

The associativity case follows similarly.

*2. Indexed split:* by induction on $M \overset{n}{\Longrightarrow} M'$. We will show just notable cases concerning to the reduction steps:
*id−step*: $M = N \star \lambda x.\mathit{unit}\, x \overset{n}{\Longrightarrow} N' = M'$. Then

$$\frac{N \overset{n}{\Longrightarrow} N'}{M = N \star \lambda x.\mathit{unit}\, x \overset{n}{\Longrightarrow} N' = M'}$$

by i.h. either $M \Rightarrow_{\neg e} M'$ (but there is no $\Rightarrow_{\neg e}$ rule that can occur) or $M \not\Rightarrow_{\neg e} M'$. This means that $M \not\Rightarrow_{\neg e} N'$ and by i.h. there exists $N''$ s.t. $M \to_e N'' \overset{n-1}{\Longrightarrow} N'$ so $M = N \star \lambda x.\mathit{unit}\, x \to_e N'' \star \lambda x.\mathit{unit}\, x \overset{n-1}{\Longrightarrow} N'$.

*$\beta_c$−step*: $M = \mathit{unit}\ V \star \lambda x.N \overset{k}{\Longrightarrow} N'[V'/x]$ where $k = n + |N'|_x \cdot m + 1$, where $N \overset{n}{\Longrightarrow} N'$ and $V \overset{m}{\Longrightarrow} V'$.
We have $M = \mathit{unit}\ V \star \lambda x.N \to_e N[V/x]$ and the substitutivity of $\overset{n}{\Longrightarrow}$ gives $M'' = N[V/x] \overset{n+|N'|_x \cdot m}{\Longrightarrow} N'[V'/x]$.

*ass−step*: If $M \overset{n}{\Longrightarrow} M'$ where $M = (L \star \lambda x.N) \star \lambda y.P$, $M' = L' \star \lambda x.(N' \star \lambda y.P')$ and $L \overset{n_L}{\Longrightarrow} L'$, $N \overset{n_N}{\Longrightarrow} N'$, $P \overset{n_P}{\Longrightarrow} P'$. There are two sub cases: either it is the case $M \Rightarrow_{\neg e} M'$, and $L \Rightarrow_{\neg e} L'$, $N \Rightarrow_{\neg e} N'$, $P \Rightarrow_{\neg e} P'$, then the claim holds.
Otherwise, if $M \not\Rightarrow_{\neg e} M'$, $L \not\Rightarrow_{\neg e} L'$ and $n_L > 0$ have to hold (otherwise $M \Rightarrow_{\neg e} M'$). By i.h. there exists $\bar{L}$ such that $L \longrightarrow_e \bar{L} \overset{n_L-1}{\Longrightarrow} L'$. So $M = (L \star \lambda x.N) \star \lambda y.P \longrightarrow_e (\bar{L} \star \lambda x.N) \star \lambda y.P \overset{n-1}{\Longrightarrow} M'$.

*3. Split:* if $M \Rightarrow_{\lambda\mathbf{C}} M'$ then there exists $n$ such that $M \overset{n}{\Longrightarrow} M'$. By induction on $n$: by indexed split property just proved there are two cases:

1. $M \Rightarrow_{\neg e} M'$ and the statement is proved since $\xrightarrow{*}_e$ is reflexive.
2. $n > 0$ and there exists $\bar{M}$ such that $M \longrightarrow_e \bar{M} \xRightarrow{n-1} M'$. By i.h. applied to $\bar{M} \xRightarrow{n-1} M'$ there exists $M''$ such that $\bar{M} \longrightarrow_e M'' \Rightarrow_{\neg e} M'$ and so $M \xrightarrow{*}_e M'' \Rightarrow_{\neg e} M'$.

$\square$