HAL open science

# Factorization and Normalization, Essentially

Beniamino Accattoli, Claudia Faggian, Giulio Guerrieri

▶ **To cite this version:**

Beniamino Accattoli, Claudia Faggian, Giulio Guerrieri. Factorization and Normalization, Essentially. APLAS 2019 - 17th Asian Symposium on Programming Languages and Systems, Dec 2019, Bali, Indonesia. 10.1007/978-3-030-34175-6_9 . hal-02411556

**HAL Id: hal-02411556**

**https://hal.science/hal-02411556v1**

Submitted on 6 Jan 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

arXiv:1908.11289v2 [cs.LO] 27 Nov 2019

# Factorization and Normalization, Essentially

Beniamino Accattoli[1], Claudia Faggian[2], and Giulio Guerrieri[3]

[1] Inria & LIX, École Polytechnique, UMR 7161, Palaiseau, France
beniamino.accattoli@inria.fr
[2] Université de Paris, IRIF, CNRS, F-75013 Paris, France faggian@irif.fr
[3] University of Bath, Department of Computer Science, Bath, UK
g.guerrieri@bath.ac.uk

**Abstract.** $\lambda$-calculi come with no fixed evaluation strategy. Different strategies may then be considered, and it is important that they satisfy some abstract rewriting property, such as factorization or normalization theorems. In this paper we provide simple proof techniques for these theorems. Our starting point is a revisitation of Takahashi's technique to prove factorization for head reduction. Our technique is both simpler and more powerful, as it works in cases where Takahashi's does not. We then pair factorization with two other abstract properties, defining *essential systems*, and show that normalization follows. Concretely, we apply the technique to four case studies, two classic ones, head and the leftmost-outermost reductions, and two less classic ones, non-deterministic weak call-by-value and least-level reductions.

## 1 Introduction

The $\lambda$-calculus is the model underlying functional programming languages and proof assistants. The gap between the model and its incarnations is huge. In particular, the $\lambda$-calculus does not come with a fixed reduction strategy, while concrete frameworks need one. A desirable property is that the reduction which is implemented terminates on all terms on which $\beta$ reduction has a reduction sequence to normal form. This is guaranteed by a *normalization theorem*. Two classic examples are the *leftmost-outermost* and *head* normalization theorems (theorems 13.2.2 and 11.4.8 in Barendregt [5]). The former states that if the term has a $\beta$-normal form, leftmost-outermost reduction is guaranteed to find it; the latter has a similar but subtler statement, roughly head reduction computes a head normal form, if the term has any.

Another classic theorem for head reduction states that head reduction approximates the $\beta$-normal form by computing an essential part of every evaluation sequence. The precise formulation is a *factorization theorem*: a sequence of $\beta$ steps $t \to_\beta^* s$ can always be re-arranged as a sequence of head steps ($\to_h$) followed by a sequence of non-head steps ($\to_{\neg h}$), that is, $t \to_h^* u \to_{\neg h}^* s$. Both head and leftmost-outermost reductions play a key role in the theory of the $\lambda$-calculus as presented in Barendregt [5] or Krivine [17].

Variants of the $\lambda$-calculus abound and are continuously introduced: weak, call-by-value, call-by-need, classical, with pattern matching, sharing, non-determinism,

probabilistic choice, quantum features, differentiation, *etc.* So, normalization and factorization theorems need to be studied in many variations. Concepts and techniques to prove these theorems do exist, but they do not have the essential, intuitive structure of other fundamental properties, such as confluence.

*This paper.* Here we provide a presentation of factorization and normalization revisiting a simple technique due to Takahashi [29], making it even *simpler* and *more widely applicable*. We separate the abstract reasoning from the concrete details of head reduction, and apply the revisited proof method to several case studies. The presentation is novel and hopefully accessible to anyone familiar with the $\lambda$-calculus, without a background in advanced notions of rewriting theory.

We provide four case studies, all following the same method. Two are revisitations of the classic cases of head and leftmost-outermost (shortened to $\ell o$) reductions. Two are folklore cases. The first is *weak* (*i.e.* out of $\lambda$-abstractions) *call-by-value* (shortened to CbV) *reduction* in its non-deterministic presentation. The second is *least-level* (shortened to $\ell\ell$) *reduction*, a reduction coming from the linear logic literature—sometimes called *by levels*—and which is usually presented using proof nets (see de Carvalho, Pagani and Tortora de Falco [7] or Pagani and Tranquilli [25]) or calculi related to proof nets (see Terui [31] or Accattoli [1]), rather than in the ordinary $\lambda$-calculus. The $\ell o$ and $\ell\ell$ cases are *full* reductions for $\beta$, *i.e.* they have the same normal forms as $\beta$. The head and weak CbV cases are not full, as they may not compute $\beta$ normal forms.

*Takahashi.* In [29], Takahashi uses the natural inductive notion of *parallel*[4] $\beta$ reduction (which reduces simultaneously a number of $\beta$-redexes; it is also the key concept in Tait and Martin-Löf's classic proof of confluence of the $\lambda$-calculus) to introduce a simple proof technique for head factorization, from which head normalization follows. By iterating head factorization, she also obtains leftmost-outermost normalization, via a simple argument on the structure of terms due to Mitschke [20].

Her technique has been employed for various $\lambda$-calculi because of its simplicity. Namely, for the $\lambda$-calculus with $\eta$ by Ishii [14], the call-by-value $\lambda$-calculus by Ronchi Della Rocca and Paolini [26,28], the resource $\lambda$-calculus by Pagani and Tranquilli [24], pattern calculi by Kesner, Lombardi and Ríos [15], the shuffling calculus by Guerrieri, Paolini and Ronchi Della Rocca [10,9,11], and it has been formalized with proof assistants by McKinna and Pollack [18] and Crary [8].

*Takahashi revisited.* Despite its simplicity, Takahashi's proof [29] of factorization relies on substitutivity properties not satisfied by full reductions such as $\ell o$ and $\ell\ell$. Our first contribution is a proof that is independent of the substitutivity properties of the factorizing reductions. It relies on a simpler fact, namely the

[4] The terminology at work in the literature on $\lambda$-calculus and the rewriting terminology often clash: the former calls *parallel $\beta$ reduction* what the latter calls *multi-step $\beta$ reduction*—parallel reduction in rewriting is something else.

substitutivity of an *indexed* variant $\overset{n}{\Rightarrow}_\beta$ of parallel $\beta$ reduction $\Rightarrow_\beta$. The definition of $\overset{n}{\Rightarrow}_\beta$ simply decorates the definition of $\Rightarrow_\beta$ with a natural number $n$ that intuitively corresponds to the number of redexes reduced in parallel by a $\Rightarrow_\beta$ step.

We prove factorization theorems for all our four case studies following this simpler scheme. We also highlight an interesting point: factorization for the two full reductions cannot be obtained directly following Takahashi's method[5].

*From factorization to essential normalization.* The second main contribution of our paper is the isolation of abstract properties that together with factorization imply normalization. First of all we abstract head reduction into a generic reduction $\to_{\mathsf{e}}$, called *essential*, and non-head reduction $\to_{\neg h}$ into a *non-essential* reduction $\to_{\neg\mathsf{e}}$. The first additional property for normalization is *persistence*: steps of the factoring reduction $\to_{\mathsf{e}}$ cannot be erased by the factored out $\to_{\neg\mathsf{e}}$. The second one is a relaxed form of determinism for $\to_{\mathsf{e}}$. We show that in such *essential* rewriting systems $\to_{\mathsf{e}}$ has a normalization theorem. The argument is abstract, that is, independent of the specific nature of terms. This is in contrast to how Takahashi [29] obtains normalization from factorization: her proof is based on an induction over the structure of terms, and cannot then be disentangled by the concrete nature of the rewriting system under study.

*Normalizing reductions for $\beta$.* We apply both our techniques to our case studies of full reduction: $\ell o$ and $\ell\ell$, obtaining simple proofs that they are normalizing reductions for $\beta$. Let us point out that $\ell o$ is also—at present—the only known deterministic reduction to $\beta$ normal form whose number of steps is a reasonable cost model, as shown by Accattoli and Dal Lago [2]. Understanding its normalization is one of the motivations at the inception of this work.

*Normalization with respect to different notions of results.* As a further feature, our approach provides for free normalization theorems for reductions that are not full for the rewrite system in which they live. Typical examples are head and weak CbV reductions, which do not compute $\beta$ and CbV normal forms, respectively. These normalization theorems arise naturally in the theory of the $\lambda$-calculus. For instance, functional programming languages implement only weak notions of reduction, and head reduction (rather than $\ell o$) is the key notion for the $\lambda$-definability of computable functions.

We obtain normalization theorems for head and weak CbV reductions. Catching normalization for non-full reductions sets our work apart from the recent studies on normalization by Hirokawa, Middeldorp, and Moser [13] and Van Oostrom and Toyama [23], discussed below among related works.

*Factorization, Normalization, Standardization.* In the literature of the $\lambda$-calculus, normalization for $\ell o$ reduction is often obtained as a corollary of the standard-

[5] It can be obtained indirectly, as a corollary of standardization, proved by Takahashi [29] using the concrete structure of terms. Thus the proof is not of an abstract nature.

ization theorem, which roughly states that every reduction sequence can be reorganized as to reduce redexes according to the left-to-right order (Terese [30] following Klop [16] and Barendregt [5], for instance). Standardization is a complex and technical result. Takahashi [29], using Mitschke's argument [20] that iterates head factorization, obtains a simpler proof technique for $\ell o$ normalization—and for standardization as well. Our work refines that approach, abstracts from it and shows that factorization is a general technique for normalization.

*Related work.* Factorization is studied in the abstract in [19,1]. Melliès axiomatic approach [19] builds on standardization, and encompasses a wide class of rewriting systems; in particular, like us, he can deal with non-full reductions. Accattoli [1] relies crucially on terminating hypotheses, absent instead here.

Hirokawa, Middeldorp, and Moser [13] and Van Oostrom and Toyama [23] study normalizing strategies via a clean separation between abstract and term rewriting results. Our approach to normalization is similar to the one used in [13] to study $\ell o$ evaluation for first-order term rewriting systems. Our essential systems strictly generalize their conditions: uniform termination replaces determinism (two of the strategies we present here are not deterministic) and—crucially—persistence strictly generalizes the property in their Lemma 7. Conversely, they focus on hyper-normalization and on extending the method to systems in which left-normality is relaxed. We do not deal with these aspects. Van Oostrom and Toyama's study [23] of (hyper-)normalization is based on an elegant and powerful method based on the random descent property and an ordered notion of commutative diagrams. Their method and ours are incomparable: we do not rely on (and do not assume) the random descent property (for its definition and uses see van Oostrom [22])—even if most strategies naturally have that property—and we do focus on factorization (which they explicitly avoid), since we see it as the crucial tool from which normalization can be obtained.

As already pointed out, a fundamental difference with respect to both works is that we consider a more general notion of normalization for reductions that are not full, that is not captured by either of those approaches.

In the literature, normalization is also proved from iterated head factorization (Takahashi [29] for $\ell o$, and Terui [31] or Accattoli [1] for $\ell\ell$ on proof nets-like calculi, or Pagani and Tranquilli [25] for $\ell\ell$ on differential proof nets), or as a corollary of standardization (Terese [30] following Klop [16] and Barendregt [5] for $\ell o$), or using semantic principles such as intersection types (Krivine [17] for $\ell o$ and de Carvalho, Pagani and Tortora de Falco [7] for $\ell\ell$ on proof nets). Last, Bonelli *et al.* develop a sophisticated proof of normalization for a $\lambda$-calculus with powerful pattern matching in [6]. Our technique differs from them all.

*Proofs.* Most proofs are in the appendix. This paper is an extended version of [3].

## 2 Factorization and Normalization, Abstractly

In this section, we study factorization and normalization abstractly, that is, independently of the specific structure of the objects to be rewritten.

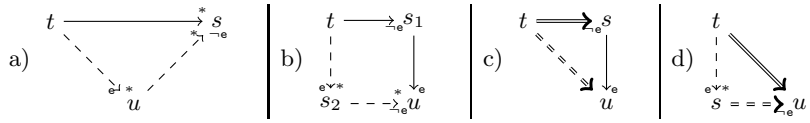


**Fig. 1.** Diagrams: a) factorization, b) weak postponement, c) merge, d) split.

A *rewriting system* (aka abstract reduction system, see Terese [30, Ch. 1]) $\mathcal{S}$ is a pair $(S, \to)$ consisting of a set $S$ and a binary relation $\to \subseteq S \times S$ called *reduction*, whose pairs are written $t \to s$ and called $\to$*-steps*. A $\to$*-sequence* from $t$ is a sequence $t \to s \to \dots$ of $\to$-steps; $t \to^k s$ denotes a sequence of $k$ $\to$-steps from $t$ to $s$. As usual, $\to^*$ (resp. $\to^=$) denotes the transitive-reflexive (resp. reflexive) closure of $\to$. Given two reductions $\to_1$ and $\to_2$ we use $\to_1 \cdot \to_2$ for their composition, defined as $t \to_1 \cdot \to_2 s$ if $t \to_1 u \to_2 s$ for some $u$.

In this section we focus on a given sub-reduction $\to_{\mathsf{e}}$ of $\to$, called *essential*, for which we study factorization and normalization with respect to $\to$. It comes with a second sub-reduction $\to_{\neg\mathsf{e}}$, called *inessential*, such that $\to_{\mathsf{e}} \cup \to_{\neg\mathsf{e}} = \to$. Despite the notation, $\to_{\mathsf{e}}$ and $\to_{\neg\mathsf{e}}$ are not required to be disjoint. In general, we write $(S, \{\to_a, \to_b\})$ for the rewriting system $(S, \to)$ where $\to \; = \; \to_a \cup \to_b$.

### 2.1 Factorization.

A rewriting system $(S, \{\to_{\mathsf{e}}, \to_{\neg\mathsf{e}}\})$ satisfies $\to_{\mathsf{e}}$*-factorization* (also called *postponement of* $\to_{\neg\mathsf{e}}$ *after* $\to_{\mathsf{e}}$) if $t \to^* s$ implies that there exists $u$ such that $t \to_{\mathsf{e}}^* u \to_{\neg\mathsf{e}}^* s$. Compactly, we write $\to^* \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^*$. In diagrams, see Fig. 1.a.

*Proving factorization.* Factorization is a non-trivial rewriting property, because it is *global*, that is, quantified over all reduction sequences from a term. To be able to prove factorization, we would like to reduce it to *local* properties, *i.e.* properties quantified only over one-step reductions from a term. At first sight it may seem that a local diagram such as the one in Fig. 1.b would give factorization by a simple induction. Such a diagram however does not allow to infer factorization without further hypotheses—counterexamples can be found in Barendregt [5].

The following abstract property is a special case for which a local condition implies factorization. It was first observed by Hindley [12].

**Lemma 1 (Hindley, local postponement).** *Let* $(S, \{\to_{\mathsf{e}}, \to_{\neg\mathsf{e}}\})$ *be a rewriting system. If* $\to_{\neg\mathsf{e}} \cdot \to_{\mathsf{e}} \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^=$ *then* $\to^* \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^*$.

*Proof.* The assumption $\to_{\neg\mathsf{e}} \cdot \to_{\mathsf{e}} \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^=$ implies (#) $\to_{\neg\mathsf{e}} \cdot \to_{\mathsf{e}}^* \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^=$ (indeed, it is immediate to prove that $\to_{\neg\mathsf{e}} \cdot \to_{\mathsf{e}}^k \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^=$ by induction on $k$).

We then prove that $\to^k \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^*$, by induction on $k$. The case $k = 0$ is trivial. Assume $\to \cdot \to^{k-1}$. By *i.h.*, $\to \cdot \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^*$. If the first step is $\to_{\mathsf{e}}$, the claim is proved. Otherwise, by (#), from $(\to_{\neg\mathsf{e}} \cdot \to_{\mathsf{e}}^*) \cdot \to_{\neg\mathsf{e}}^*$ we obtain $(\to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^=) \cdot \to_{\neg\mathsf{e}}^*$. $\square$

Hindley's local condition is a strong hypothesis for factorization that in general does not hold in $\lambda$-calculi—not even in the simple case of head reduction. However, the property can be applied in combination with another standard technique: switching to *macro* steps that compress $\to_{\mathsf{e}}^*$ or $\to_{\neg\mathsf{e}}^*$ into just one step, at the price of some light overhead. This idea is the essence of both Tait–Martin-Löf's and Takahashi's techniques, based on *parallel steps*. The role of parallel steps in Takahashi [29] is here captured abstractly by the notion of macro-step system.

**Definition 2 (Macro-step system).** *A rewriting system* $\mathcal{S} = (S, \{\to_{\mathsf{e}}, \to_{\neg\mathsf{e}}\})$ *is a* macro-step system *if there are two reductions* $\Rightarrow$ *and* $\Rightarrow_{\neg\mathsf{e}}$ *(called* macro-steps *and* inessential macro-steps*, respectively) such that*

- Macro*:* $\to_{\neg\mathsf{e}} \subseteq \Rightarrow_{\neg\mathsf{e}} \subseteq \to_{\neg\mathsf{e}}^*$.
- Merge*: if* $t \Rightarrow_{\neg\mathsf{e}} \cdot \to_{\mathsf{e}} u$ *then* $t \Rightarrow u$*. That is, the diagram in Fig. 1.c holds.*
- Split*: if* $t \Rightarrow u$ *then* $t \to_{\mathsf{e}}^* \cdot \Rightarrow_{\neg\mathsf{e}} u$*. That is, the diagram in Fig. 1.d holds.*

Note that $\Rightarrow$ just plays the role of a "bridge" between the hypothesis of the merge condition and the conclusion of the split condition—it shall play a crucial role in the concrete proofs in the next sections. In this paper, concrete instances of $\Rightarrow$ and $\Rightarrow_{\neg\mathsf{e}}$ shall be parallel $\beta$ reduction and some of its variants.

**Proposition 3 (Factorization).** *Every macro-step system* $(S, \{\to_{\mathsf{e}}, \to_{\neg\mathsf{e}}\})$ *satisfies* $\to_{\mathsf{e}}$*-factorization.*

*Proof.* By Merge and Split, $\Rightarrow_{\neg\mathsf{e}} \cdot \to_{\mathsf{e}} \subseteq \Rightarrow \subseteq \to_{\mathsf{e}}^* \cdot \Rightarrow_{\neg\mathsf{e}} \subseteq \to_{\mathsf{e}}^* \cdot \Rightarrow_{\neg\mathsf{e}}^=$. By Hindley's lemma (Lemma 1) applied to $\to_{\mathsf{e}}$ and $\Rightarrow_{\neg\mathsf{e}}$ (rather than $\to_{\mathsf{e}}$ and $\to_{\neg\mathsf{e}}$), we obtain $(\to_{\mathsf{e}} \cup \Rightarrow_{\neg\mathsf{e}})^* \subseteq \to_{\mathsf{e}}^* \cdot \Rightarrow_{\neg\mathsf{e}}^*$. Since $\to_{\neg\mathsf{e}} \subseteq \Rightarrow_{\neg\mathsf{e}}$, we have $(\to_{\mathsf{e}} \cup \to_{\neg\mathsf{e}})^* \subseteq (\to_{\mathsf{e}} \cup \Rightarrow_{\neg\mathsf{e}})^* \subseteq \to_{\mathsf{e}}^* \cdot \Rightarrow_{\neg\mathsf{e}}^*$. As $\Rightarrow_{\neg\mathsf{e}} \subseteq \to_{\neg\mathsf{e}}^*$, we have $\to_{\mathsf{e}}^* \cdot \Rightarrow_{\neg\mathsf{e}}^* \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^*$. Therefore, $\to^* = (\to_{\mathsf{e}} \cup \to_{\neg\mathsf{e}})^* \subseteq \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^*$. □

### 2.2 Normalization for full reductions

The interest of factorization comes from the fact that the essential reduction $\to_{\mathsf{e}}$ on which factorization pivots has some good properties. Here we pinpoint the abstract properties which make factorization a privileged method to prove normalization; we collect them into the definition of *essential system* (Def. 5).

*Normal forms and normalization.* Let us recall what normalization is about. In general, a term may or may not reduce to a normal form. And if it does, not all reduction sequences necessarily lead to normal form. A term is *weakly* or *strongly normalizing*, depending on if it may or must reduce to normal form. If a term $t$ is strongly normalizing, any choice of steps will eventually lead to a normal form. However, if $t$ is weakly normalizing, how do we compute a normal form? This is the problem tackled by *normalization*: by repeatedly performing *only specific steps*, a normal form will be computed, provided that $t$ can reduce to any.

Recall the statement of the $\ell o$ normalization theorem: if $t \to_\beta^* u$ with $u$ $\beta$-normal, then $t$ $\ell o$-reduces to $u$. Observe a subtlety: such a formulation relies on

the determinism of $\ell o$ reduction. We give a more general formulation of normalizing reduction, valid also for non-deterministic reductions.

Formally, given a rewriting system $(S, \to)$, a term $t \in S$ is:

- *$\to$-normal* (or in *$\to$-normal form*) if $t \not\to$, *i.e.* there are no $s$ such that $t \to s$;
- *weakly $\to$-normalizing* if *there exists* a sequence $t \to^* s$ with $s$ $\to$-normal;
- *strongly $\to$-normalizing* if there are no infinite $\to$-sequences from $t$, or equivalently, if *all* maximal $\to$-sequences from $t$ are finite.

We call *reduction for* $\to$ any $\to_{\mathsf{e}} \subseteq \to$. It is *full* if $\to_{\mathsf{e}}$ and $\to$ have the same normal forms.[6]

**Definition 4 (Normalizing reduction).** *A full reduction $\to_{\mathsf{e}}$ for $\to$ is* normalizing *(for $\to$) if, for every term $t$, $t$ is strongly $\to_{\mathsf{e}}$-normalizing whenever it is weakly $\to$-normalizing.*

Note that, since the normalizing reduction $\to_{\mathsf{e}}$ is full, if $t$ is strongly $\to_{\mathsf{e}}$-normalizing then *every* maximal $\to_{\mathsf{e}}$-sequence from $t$ ends in a $\to$-normal form.

**Definition 5 (Essential system).** *A rewriting system $(S, \{\to_{\mathsf{e}}, \to_{\neg\mathsf{e}}\})$ is* essential *if the following conditions hold:*

1. Persistence: *if $t \to_{\mathsf{e}} s$ and $t \to_{\neg\mathsf{e}} u$, then $u \to_{\mathsf{e}} r$ for some $r$.*
2. Uniform termination*: if $t$ is weakly $\to_{\mathsf{e}}$-normalizing, then it is strongly $\to_{\mathsf{e}}$-normalizing.*
3. Terminal factorization*: if $t \to^* u$ and $u$ is $\to_{\mathsf{e}}$-normal, then $t \to_{\mathsf{e}}^* \cdot \to_{\neg\mathsf{e}}^* u$.*

*It is moreover* full *if $\to_{\mathsf{e}}$ is a full reduction for $\to$.*

Comments on the definition:

- *Persistence*: it means that essential steps are out of reach for inessential steps, that cannot erase them. The only way of getting rid of essential steps is by reducing them, and so in that sense they are *essential* to normalization.
- *From determinism to uniform termination*: as we already said, in general $\to_{\mathsf{e}}$ is not deterministic. For normalization, then, it is not enough that there is a sequence $t \to_{\mathsf{e}}^* u$ with $u$ $\to$-normal (as in the statement of $\ell o$-normalization). We need to be sure that there are no infinite $\to_{\mathsf{e}}$-sequences from $t$. This is exactly what is ensured by the uniform termination property. Note that if $\to_{\mathsf{e}}$ is deterministic (or has the diamond or random descent properties) then it is uniformly terminating.
- *Terminal factorization*: there are two subtleties. First, we need only a weak form of factorization, namely factorization is only required for $\to$-sequences

[6] In rewriting theory, a *full reduction for* $\to$ is called a *reduction strategy for* $\to$. We prefer not to use the term strategy because it has different meaning in the $\lambda$-calculus, where it is a *deterministic, not necessarily full*, reduction for $\to$.

ending in a $\to_{\mathsf{e}}$-normal form[7]. Second, the reader may expect terminal factorization to be required with respect to $\to$-normal rather than $\to_{\mathsf{e}}$-normal forms. The two notions coincide if $\to_{\mathsf{e}}$ is full, and for the time being we only discuss full essential systems. We discuss the more general case in Sect. 2.3.

*Example 6.* In the $\lambda$-calculus with $\beta$ reduction, head reduction $\to_h$ and its associated $\to_{\neg h}$ reduction (defined in Sect. 4) form an essential system. Similarly, leftmost-outermost $\to_{\ell o}$ reduction and its associated $\to_{\neg \ell o}$ reduction (Sect. 7) form a full essential system. Two more examples are in Sect. 6 and Sect. 8.

**Theorem 7 (Essential full normalization).** *Let $(S, \{\to_{\mathsf{e}}, \to_{\neg\mathsf{e}}\})$ be a full essential system. Then $\to_{\mathsf{e}}$ is a normalizing reduction for $\to$.*

*Proof.* Let $t$ be a weakly $\to$-normalizing term, *i.e.* $t \to^* u$ for some term $u$ in $\to$-normal form (and so in $\to_{\mathsf{e}}$-normal form, since $\to_{\mathsf{e}} \subseteq \to$).

1. *Terminal factorization* implies $t \to_{\mathsf{e}}^* s \to_{\neg\mathsf{e}}^* u$ for some $s$, since $u$ is $\to_{\mathsf{e}}$-normal.
2. Let us show that $s$ is $\to_{\mathsf{e}}$-normal: if not, then $s \to_{\mathsf{e}} r$ for some $r$, and a straightforward induction on the length of $s \to_{\neg\mathsf{e}}^* u$ iterating *persistence* gives that $u \to_{\mathsf{e}} p$ for some $p$, against the hypothesis that $u$ is $\to_{\mathsf{e}}$-normal. Absurd.
3. By the previous point, $t$ is weakly $\to_{\mathsf{e}}$-normalizing. By *uniform termination*, $t$ is strongly $\to_{\mathsf{e}}$-normalizing. □

### 2.3 A more general notion of normalizing reduction.

Essential systems actually encompass also important notions of normalization for reductions that are *not full*, such as *head normalization*. These cases arise naturally in the $\lambda$-calculus literature, where partial notions of result such as head normal forms or values are of common use. Normalization for non-full reductions is instead not so common in the rewriting literature outside the $\lambda$-calculus. This is why, to guide the reader, we presented first the natural case of full reductions.

Let us first discuss head reduction: $\to_h$ is deterministic and not full with respect to $\to_\beta$, as its normal forms may not be $\to_\beta$-normal forms. The well-known property of interest is *head normalization* (Cor. 11.4.8 in Barendregt's book [5]):

If $t \to_\beta^* s$ and $s$ is head normal[8] then $\to_h$ terminates on $t$.

[7] The difference between factorization and its terminal case is relevant for normalization: van Oostrom and Toyama [23, footnote 8] give an example of normalizing full reduction for a rewriting system in which factorization fails but terminal factorization holds.

[8] "$t$ has a head normal form" is the usual formulation for "$t \to_\beta^* s$ for some $s$ that is head normal". We prefer the latter to avoid the ambiguity of the former about the reduction leading from $t$ to one of its head normal forms ($\to_\beta^*$ or $\to_h^*$?).

The statement has two subtleties. First, $t$ may $\to_\beta$-reduce to a term in $\to_h$-normal form in many different ways, possibly without using $\to_h$, so that the hypotheses may not imply that $\to_h$ terminates. Second, the conclusion is "$\to_h$ *terminates on* $t$" and not $t \to_h^* s$, because in general the maximal $\to_h$-sequence from $t$ may end in a term $u \neq s$. For instance, let $I = \lambda y.y$: then $I(x(II)) \to_\beta I(xI) \to_\beta xI$ is a $\to_\beta$-sequence to head normal form, and yet the maximal $\to_h$-sequence $I(x(II)) \to_h x(II)$ ends in a different term.

Now, let us abstract from head normalization, taking into account that in general the essential reduction $\to_{\mathsf{e}}$—unlike head reduction—may not be deterministic, and so we ask for strong $\to_{\mathsf{e}}$-normalization rather than for $\to_{\mathsf{e}}$-termination.

**Theorem 8 (Essential normalization).** *Let $(S, \{\to_{\mathsf{e}}, \to_{\neg\mathsf{e}}\})$ be an essential system. If $t \to^* u$ and $u$ is $\to_{\mathsf{e}}$-normal, then $t$ is strongly $\to_{\mathsf{e}}$-normalizing.*

*Proof.* Exactly as for Theorem 7, fullness is not used in that proof. □

In the next section we shall apply Theorem 8 to head reduction and obtain the head normalization theorem we started with. Another example of a normalization theorem for a non-full reduction is in Sect. 6. Note that the full variant of the theorem (Theorem 7) is in fact an instance of the general one (Theorem 8).

## 3 The λ-Calculus

This short section recalls basic definitions and properties of the λ-calculus and introduces the indexed variant of parallel $\beta$.

The set $\Lambda$ of *terms* of the λ-calculus is given by the following grammar:

$$\textsc{Terms} \quad t, s, u, r ::= x \mid \lambda x.t \mid ts$$

We use the usual notions of free and bound variables, $t\{x \leftarrow s\}$ for the meta-level capture-avoiding substitution of $s$ for the free occurrences of $x$ in $t$, and $|t|_x$ for the number of free occurrences of $x$ in $t$. The definition of $\beta$ reduction $\to_\beta$ is:

$\beta$ REDUCTION

$$\frac{}{(\lambda x.t)s \to_\beta t\{x \leftarrow s\}} \qquad \frac{t \to_\beta t'}{ts \to_\beta t's} \qquad \frac{t \to_\beta t'}{\lambda x.t \to_\beta \lambda x.t'} \qquad \frac{t \to_\beta t'}{st \to_\beta st'}$$

Let us recall two basic substitutivity properties of $\beta$ reduction.

1. *Left substitutivity of* $\to_\beta$: if $t \to_\beta t'$ then $t\{x \leftarrow s\} \to_\beta t'\{x \leftarrow s\}$.
2. *Right substitutivity of* $\to_\beta$: if $s \to_\beta s'$ then $t\{x \leftarrow s\} \to_\beta^* t\{x \leftarrow s'\}$. It is possible to spell out the number of $\to_\beta$-steps, which is exactly the number of free occurrences of $x$ in $t$, that is, $t\{x \leftarrow s\} \to_\beta^{|t|_x} t\{x \leftarrow s'\}$.

*Parallel $\beta$ reduction.* Parallel $\beta$-reduction $\Rightarrow_\beta$ is defined by:

Parallel $\beta$ reduction

$$\frac{}{x \Rightarrow_\beta x} \qquad \frac{t \Rightarrow_\beta t'}{\lambda x.t \Rightarrow_\beta \lambda x.t'} \qquad \frac{t \Rightarrow_\beta t' \quad s \Rightarrow_\beta s'}{ts \Rightarrow_\beta t's'} \qquad \frac{t \Rightarrow_\beta t' \quad s \Rightarrow_\beta s'}{(\lambda x.t)s \Rightarrow_\beta t'\{x{\leftarrow}s'\}}$$

Tait–Martin-Löf's proof of the confluence of $\to_\beta$ relies on the diamond property of $\Rightarrow_\beta$[9], in turn based on the following property (see Takahashi [29, p. 1])

*Substitutivity of* $\Rightarrow_\beta$: if $t \Rightarrow_\beta t'$ and $s \Rightarrow_\beta s'$ then $t\{x{\leftarrow}s\} \Rightarrow_\beta t'\{x{\leftarrow}s'\}$.

While the diamond property of $\Rightarrow_\beta$ does not play a role for factorization, one of the contributions of this work is a new proof technique for factorization relying on the substitutivity property of an indexed refinement of $\Rightarrow_\beta$.

*Indexed parallel $\beta$ reduction.* The new *indexed* version $\overset{n}{\Rightarrow}_\beta$ of parallel $\beta$ reduction $\Rightarrow_\beta$ is equipped with a natural number $n$ which is, roughly, the number of redexes reduced in parallel by a $\Rightarrow_\beta$; more precisely, $n$ is the length of a particular way of sequentializing the redexes reduced by $\Rightarrow_\beta$. The definition of $\overset{n}{\Rightarrow}_\beta$ is as follows (note that erasing the index one obtains exactly $\Rightarrow_\beta$, so that $\Rightarrow_\beta = \bigcup_{n\in\mathbb{N}} \overset{n}{\Rightarrow}_\beta$):

Indexed parallel $\beta$ reduction

$$\frac{}{x \overset{0}{\Rightarrow}_\beta x} \qquad \frac{t \overset{n}{\Rightarrow}_\beta t'}{\lambda x.t \overset{n}{\Rightarrow}_\beta \lambda x.t'} \qquad \frac{t \overset{n}{\Rightarrow}_\beta t' \quad s \overset{m}{\Rightarrow}_\beta s'}{ts \overset{n+m}{\Rightarrow}_\beta t's'} \qquad \frac{t \overset{n}{\Rightarrow}_\beta t' \quad s \overset{m}{\Rightarrow}_\beta s'}{(\lambda x.t)s \overset{n+|t'|_x\cdot m+1}{\Rightarrow}_\beta t'\{x{\leftarrow}s'\}}$$

The intuition behind the last clause is: $(\lambda x.t)s$ reduces to $t'\{x{\leftarrow}s'\}$ by

1. first reducing $(\lambda x.t)s$ to $t\{x{\leftarrow}s\}$ (1 step);
2. then reducing in $t\{x{\leftarrow}s\}$ the $n$ steps corresponding to the sequence $t \overset{n}{\Rightarrow}_\beta t'$, obtaining $t'\{x{\leftarrow}s\}$;
3. then reducing $s$ to $s'$ for every occurrence of $x$ in $t'$ replaced by $s$, that is, $m$ steps $|t'|_x$ times, obtaining $t'\{x{\leftarrow}s'\}$.

Points 2 and 3 hold because of the substitutivity properties of $\beta$ reduction.

It is easily seen that $\overset{0}{\Rightarrow}_\beta$ is the identity relation on terms. Moreover, $\to_\beta = \overset{1}{\Rightarrow}_\beta$, and $\overset{n}{\Rightarrow}_\beta \subseteq \to_\beta^n$, as expected. The substitutivity of $\overset{n}{\Rightarrow}_\beta$ is proved by simply indexing the proof of substitutivity of $\Rightarrow_\beta$.

**Lemma 9 (Substitutivity of $\overset{n}{\Rightarrow}_\beta$).** *If $t \overset{n}{\Rightarrow}_\beta t'$ and $s \overset{m}{\Rightarrow}_\beta s'$, then $t\{x{\leftarrow}s\} \overset{k}{\Rightarrow}_\beta t'\{x{\leftarrow}s'\}$ where $k = n + |t'|_x \cdot m$.*

*Proof.* By induction on the derivation of $t \overset{n}{\Rightarrow}_\beta t'$. Consider its last rule. Cases:

– *Variable*: two sub-cases

[9] Namely, if $s_1 \ {}_\beta\!\Leftarrow t \Rightarrow_\beta s_2$ then there exists $u$ such that $s_1 \Rightarrow_\beta u \ {}_\beta\!\Leftarrow s_2$.

- $t = x$: then $t = x \overset{0}{\Rightarrow}_\beta x = t'$ then $t\{x{\leftarrow}s\} = x\{x{\leftarrow}s\} = s \overset{m}{\Rightarrow}_\beta s' = x\{x{\leftarrow}s'\} = t'\{x{\leftarrow}s'\}$ that satisfies the statement because $n + |t'|_x \cdot m = 0 + 1 \cdot m = m$.
- $t = y$: then $t = y \overset{0}{\Rightarrow}_\beta y = t'$ and $t\{x{\leftarrow}s\} = y\{x{\leftarrow}s\} = y \overset{0}{\Rightarrow}_\beta y = y\{x{\leftarrow}s'\} = t'\{x{\leftarrow}s'\}$ that satisfies the statement because $n + |t'|_x \cdot m = 0 + 0 \cdot m = 0$.

– *Abstraction*, *i.e.* $t = \lambda y.u \overset{n}{\Rightarrow}_\beta \lambda y.u' = t'$ because $u \overset{n}{\Rightarrow}_\beta u'$; we can suppose without loss of generality that $y \neq x$ and $y$ is not free in $s$ (and hence in $s'$), so $|u'|_x = |t'|_x$ and $t\{x{\leftarrow}s\} = \lambda y.(u\{x{\leftarrow}s\})$ and $t'\{x{\leftarrow}s'\} = \lambda y.(u'\{x{\leftarrow}s'\})$. By *i.h.*, $u\{x{\leftarrow}s\} \overset{n+|u'|_x\cdot m}{\Rightarrow}_\beta u'\{x{\leftarrow}s'\}$, thus

$$\frac{u\{x{\leftarrow}s\} \overset{n+|u'|_x\cdot m}{\Rightarrow}_\beta u'\{x{\leftarrow}s'\}}{t\{x{\leftarrow}s\} = \lambda y.u\{x{\leftarrow}s\} \overset{n+|t'|_x\cdot m}{\Rightarrow}_\beta \lambda y.u'\{x{\leftarrow}s'\} = t'\{x{\leftarrow}s'\}}.$$

– *Application*, *i.e.* $t = ur \overset{n}{\Rightarrow}_\beta u'r' = t'$ with $u \overset{n_u}{\Rightarrow}_\beta u'$, $r \overset{n_r}{\Rightarrow}_\beta r'$ and $n = n_u + n_r$. By *i.h.*, $u\{x{\leftarrow}s\} \overset{n_u+|u'|_x\cdot m}{\Rightarrow}_\beta u'\{x{\leftarrow}s'\}$ and $r\{x{\leftarrow}s\} \overset{n_r+|r'|_x\cdot m}{\Rightarrow}_\beta r'\{x{\leftarrow}s'\}$. Then

$$\frac{u\{x{\leftarrow}s\} \overset{n_u+|u'|_x\cdot m}{\Rightarrow}_\beta u'\{x{\leftarrow}s'\} \qquad r\{x{\leftarrow}s\} \overset{n_r+|r'|_x\cdot m}{\Rightarrow}_\beta r'\{x{\leftarrow}s'\}}{t\{x{\leftarrow}s\} = u\{x{\leftarrow}s\}r\{x{\leftarrow}s\} \overset{k}{\Rightarrow}_\beta u'\{x{\leftarrow}s'\}r'\{x{\leftarrow}s'\} = t'\{x{\leftarrow}s'\}}$$

where $k = n_u + |u'|_x \cdot m + n_r + |r'|_x \cdot m = n + (|u'|_x + |r'|_x) \cdot m = n + |t'|_x \cdot m$.

– *$\beta$-step*, *i.e.* $t = (\lambda y.u)r \overset{n}{\Rightarrow}_\beta u'\{y{\leftarrow}r'\} = t'$ with $u \overset{n_u}{\Rightarrow}_\beta u'$, $r \overset{n_r}{\Rightarrow}_\beta r'$ and $n = n_u + |u'|_y \cdot n_r + 1$. We can assume without loss of generality that $y \neq x$ and $y$ is not free in $s$ (and so in $s'$), hence $|t'|_x = |u'\{y{\leftarrow}r'\}|_x = |u'|_x + |u'|_y \cdot |r'|_x$ and $|u'\{x{\leftarrow}s'\}|_y = |u'|_y$ and $t\{x{\leftarrow}s\} = (\lambda y.u\{x{\leftarrow}s\})(r\{x{\leftarrow}s\})$ and $t'\{x{\leftarrow}s'\} = u'\{x{\leftarrow}s'\}\{y{\leftarrow}r'\{x{\leftarrow}s'\}\}$.

  By *i.h.*, $u\{x{\leftarrow}s\} \overset{n_u+|u'|_x\cdot m}{\Rightarrow}_\beta u'\{x{\leftarrow}s'\}$ and $r\{x{\leftarrow}s\} \overset{n_r+|r'|_x\cdot m}{\Rightarrow}_\beta r'\{x{\leftarrow}s'\}$. Then

$$\frac{u\{x{\leftarrow}s\} \overset{n_u+|u'|_x\cdot m}{\Rightarrow}_\beta u'\{x{\leftarrow}s'\} \qquad r\{x{\leftarrow}s\} \overset{n_r+|r'|_x\cdot m}{\Rightarrow}_\beta r'\{x{\leftarrow}s'\}}{t\{x{\leftarrow}s\} = (\lambda y.u\{x{\leftarrow}s\})(r\{x{\leftarrow}s\}) \overset{k}{\Rightarrow}_\beta u'\{x{\leftarrow}s'\}\{y{\leftarrow}r'\{x{\leftarrow}s'\}\} = t'\{x{\leftarrow}s'\}}$$

  where $k = n_u + |u'|_x \cdot m + |u'|_y \cdot (n_r + |r'|_x \cdot m) + 1 = n_u + |u'|_y \cdot n_r + 1 + |u'|_x \cdot m + |u'|_y \cdot |r'|_x \cdot m = n + |t'|_x \cdot m$. □

## 4 Head Reduction, Essentially

We here revisit Takahashi's study [29] of head reduction. We apply the abstract schema for essential reductions developed in Sect. 2, which is the same schema used by Takahashi, but we provide a simpler proof technique for one of the required properties (split). First of all, head reduction $\rightarrow_h$ (our essential reduction

here) and its associated inessential reduction $\to_{\neg h}$ are defined by:

Head reduction

$$\frac{}{(\lambda x.t)s \to_h t\{x \leftarrow s\}} \qquad \frac{t \to_h s \qquad t \neq \lambda x.t'}{tu \to_h su} \qquad \frac{t \to_h s}{\lambda x.t \to_h \lambda x.s}$$

¬Head reduction

$$\frac{t \to_\beta t'}{(\lambda x.t)s \to_{\neg h} (\lambda x.t')s} \qquad \frac{t \to_\beta t'}{st \to_{\neg h} st'} \qquad \frac{t \to_{\neg h} t'}{\lambda x.t \to_{\neg h} \lambda x.t'} \qquad \frac{t \to_{\neg h} t'}{ts \to_{\neg h} t's}.$$

Note that $\to_\beta = \to_h \cup \to_{\neg h}$ but $\to_h$ and $\to_{\neg h}$ are not disjoint: $I(II) \to_h II$ and $I(II) \to_{\neg h} II$ with $I = \lambda z.z$. Indeed, $I(II)$ contains two distinct redexes, one is $I(II)$ and is fired by $\to_h$, the other one is $II$ and is fired by $\to_{\neg h}$; coincidentally, the two reductions lead to the same term.

As for Takahashi, a parallel ¬head step $t \Rightarrow_{\neg h} s$ is a parallel step $t \Rightarrow_\beta s$ such that $t \to^*_{\neg h} s$. We give explicitly the inference rules for $\Rightarrow_{\neg h}$:

Parallel ¬head reduction

$$\frac{}{x \Rightarrow_{\neg h} x} \qquad \frac{t \Rightarrow_\beta t' \quad s \Rightarrow_\beta s'}{(\lambda x.t)s \Rightarrow_{\neg h} (\lambda x.t')s'} \qquad \frac{t \Rightarrow_{\neg h} t'}{\lambda x.t \Rightarrow_{\neg h} \lambda x.t'} \qquad \frac{t \Rightarrow_{\neg h} t' \quad s \Rightarrow_\beta s'}{ts \Rightarrow_{\neg h} t's'}$$

Easy inductions show that $\to_{\neg h} \subseteq \Rightarrow_{\neg h} \subseteq \to^*_{\neg h}$. It is immediate that $\to_h$-normal terms are head normal forms in the sense of Barendregt [5, Def. 2.2.11]. We do not describe the shape of head normal forms. Our proofs never use it, unlike Takahashi's ones. This fact stresses the abstract nature of our proof method.

*Head factorization.* We show that $\to_h$ induces a macro-step system, with respect to $\to_{\neg h}$, $\Rightarrow_\beta$, and $\Rightarrow_{\neg h}$, to obtain $\to_h$-factorization by Proposition 3.

Therefore, we need to prove merge and split. Merge is easily verified by induction on $t \Rightarrow_{\neg h} s$. The interesting part is the proof of the split property, that in the concrete case of head reduction becomes: if $t \Rightarrow_\beta s$ then $t \to^*_h \cdot \Rightarrow_{\neg h} s$. This is obtained as a consequence of the following easy *indexed split* property based on the indexed variant of parallel $\beta$. The original argument of Takahashi [29] is more involved, we discuss it after the new proof.

**Proposition 10 (Head macro-step system).**

1. Merge*: if $t \Rightarrow_{\neg h} \cdot \to_h u$ then $t \Rightarrow_\beta u$.*
2. Indexed split*: if $t \overset{n}{\Rightarrow}_\beta s$ then $t \Rightarrow_{\neg h} s$, or $n > 0$ and $t \to_h \cdot \overset{n-1}{\Rightarrow}_\beta s$.*
3. Split*: if $t \Rightarrow_\beta s$ then $t \to^*_h \cdot \Rightarrow_{\neg h} s$.*

*That is, $(\Lambda, \{\to_h, \to_{\neg h}\})$ is a macro-step system with respect to $\Rightarrow_\beta$ and $\Rightarrow_{\neg h}$.*

*Proof.* 1. Easy induction on $t \Rightarrow_{\neg h} s$. Details are in the Appendix, p. 23.

2. By induction on $t \overset{n}{\Rightarrow}_\beta s$. We freely use the fact that if $t \overset{n}{\Rightarrow}_\beta s$ then $t \Rightarrow_\beta s$. Cases:

- *Variable*: $t = x \overset{0}{\Rightarrow}_\beta x = s$. Then $t = x \Rightarrow_{\neg h} x = s$.
- *Abstraction*: $t = \lambda x.t' \overset{n}{\Rightarrow}_\beta \lambda x.s' = s$ with $t' \overset{n}{\Rightarrow}_\beta s'$. It follows from the *i.h.*
- *Application*: $t = rp \overset{n}{\Rightarrow}_\beta r'p' = s$ with $r \overset{n_1}{\Rightarrow}_\beta r'$, $p \overset{n_2}{\Rightarrow}_\beta p'$ and $n = n_1 + n_2$. There are only two subcases:
  - either $rp \Rightarrow_{\neg h} r'p'$, and then the claim holds;
  - or $rp \not\Rightarrow_{\neg h} r'p'$, and then neither $r \Rightarrow_{\neg h} r'$ nor $r$ is an abstraction (otherwise $rp \Rightarrow_{\neg h} r'p'$). By *i.h.* applied to $r \overset{n_1}{\Rightarrow}_\beta r'$, $n_1 > 0$ and there exists $r''$ such that $r \to_h r'' \overset{n_1-1}{\Rightarrow}_\beta r'$. Thus, $t = rp \to_h r''p$ and

$$\frac{r'' \overset{n_1-1}{\Rightarrow}_\beta r' \qquad p \overset{n_2}{\Rightarrow}_\beta p'}{r''p \overset{n_1-1+n_2}{\Rightarrow}_\beta r'p' = s}\ .$$

- *$\beta$-step*: $t = (\lambda x.u)r \overset{n}{\Rightarrow}_\beta u'\{x \leftarrow r'\} = s$ with $u \overset{n_1}{\Rightarrow}_\beta u'$, $r \overset{n_2}{\Rightarrow}_\beta r''$ and $n = n_1 + |u'|_x \cdot n_2 + 1 > 0$. We have $t = (\lambda x.u)r \to_h u\{x \leftarrow r\}$ and by substitutivity of $\overset{n}{\Rightarrow}_\beta$ (Lemma 9) $u\{x \leftarrow r\} \overset{n_1+|u'|_x \cdot n_2}{\Rightarrow}_\beta u'\{x \leftarrow r'\} = s$.

3. If $t \Rightarrow_\beta s$ then $t \overset{n}{\Rightarrow}_\beta s$ for some $n$. We prove the statement by induction $n$. By *indexed split* (Point 2), there are only two cases:
   - $t \Rightarrow_{\neg h} s$. This is an instance of the statement (since $\to_h^*$ is reflexive).
   - $n > 0$ and there exists $r$ such that $t \to_h r \overset{n-1}{\Rightarrow}_\beta s$. By *i.h.* applied to $r \overset{n-1}{\Rightarrow}_\beta s$, there is $u$ such that $r \to_h^* u \Rightarrow_{\neg h} s$, and so $t \to_h^* u \Rightarrow_{\neg h} s$. □

**Theorem 11 (Head factorization).** *If $t \to_\beta^* u$ then $t \to_h^* \cdot \to_{\neg h}^* u$.*

*Head normalization.* We show that $(\Lambda, \{\to_h, \to_{\neg h}\})$ is an essential system (Def. 5); thus the essential normalization theorem (Theorem 8) provides normalization. We already proved *factorization* (Theorem 11, hence terminal factorization). We verify persistence and determinism (which implies uniform termination) of $\to_h$.

**Proposition 12 (Head essential system).** *Proof p. 24*

1. Persistence*: if $t \to_h s$ and $t \to_{\neg h} u$ then $u \to_h r$ for some $r$.*
2. Determinism*: if $t \to_h s_1$ and $t \to_h s_2$ then $s_1 = s_2$.*

*Then, $(\Lambda, \{\to_h, \to_{\neg h}\})$ is an essential system.*

**Theorem 13 (Head normalization).** *If $t \to_\beta^* s$ and $s$ is a $\to_h$-normal form, then $\to_h$ terminates on $t$.*

### 4.1 Comparison with Takahashi's proof of the split property.

Our technique differs from Takahashi's in that it is built on simpler properties: it exploits directly the substitutivity of $\Rightarrow_\beta$, which is instead not used by Takahashi. Takahashi's original argument [29] for the split property (*if $t \Rightarrow_\beta s$ then $t \to_h^* \cdot \Rightarrow_{\neg h}$*, what she calls the *main lemma*) is by induction on the (concrete) definition of $\Rightarrow_\beta$ and relies on two substitutivity properties of $\to_h$ and $\Rightarrow_{\neg h}$. Looking at them as the reductions $\to_{\mathsf{e}}$ and $\to_{\neg\mathsf{e}}$ of an essential system, these properties are:

- *Left substitutivity of* $\to_{\mathsf{e}}$: if $u \to_{\mathsf{e}} q$ then $u\{x{\leftarrow}r\} \to_{\mathsf{e}} q\{x{\leftarrow}r\}$;
- *Left substitutivity of* $\Rightarrow_{\neg\mathsf{e}}$: if $u \Rightarrow_{\neg\mathsf{e}} q$ then $u\{x{\leftarrow}r\} \Rightarrow_{\neg\mathsf{e}} q\{x{\leftarrow}r\}$.

From them, left substitutivity of the composed reduction $\to_{\mathsf{e}}^* \cdot \Rightarrow_{\neg\mathsf{e}}$ easily follows. That is, Takahashi's proof of the split property is by induction on $t \Rightarrow s$ using left substitutivity of $\to_{\mathsf{e}}^* \cdot \Rightarrow_{\neg\mathsf{e}}$ for the inductive case.

We exploit the substitutivity of $\overset{n}{\Rightarrow}$ instead of left substitutivity of $\to_{\mathsf{e}}$ and $\Rightarrow_{\neg\mathsf{e}}$. It holds for a larger number of essential systems because $\overset{n}{\Rightarrow}$ is simply a decoration of $\Rightarrow$, which is substitutive *by design*. There are important systems where Takahashi's hypotheses do not hold. One such case is $\ell o$ reduction (Sect. 7)—*the* normalizing reduction of the $\lambda$-calculus—we discuss the failure of left substitutivity for $\ell o$ on p. 18; another notable case is $\ell\ell$ reduction (Sect. 8); both are full reductions for $\beta$.

Let us point out where the idea behind our approach stems from. In a sense, Takahashi's proof works by chance: the split hypothesis is about a *parallel* step $\Rightarrow_\beta$ but then the key fact used in the proof, left substitutivity of $\to_h^* \cdot \Rightarrow_{\neg h}$, does no longer stay in the borders of the parallel step, since the prefix $\to_h^*$ is an arbitrary long sequence that may reduce *created* steps. Our proof scheme instead only focuses on the (expected) substitutivity of $\overset{n}{\Rightarrow}$, independently of creations.

## 5 The Call-by-Value λ-Calculus

In this short section, we introduce Plotkin's call-by-value $\lambda$-calculus [27], where $\beta$ reduction fires only when the argument is a value. In the next section we define *weak* reduction and prove factorization and normalization theorems using the essential technique, exactly as done in the previous section for head reduction.

The set $\Lambda$ of terms is the same as in Sect. 3. Values, call-by-value (CbV) $\beta$-reduction $\to_{\beta_v}$, and CbV indexed parallel reduction $\overset{n}{\Rightarrow}_{\beta_v}$ are defined as follows:

$$\textsc{Values} \quad v ::= x \mid \lambda x.t$$

$\beta_v$ REDUCTION

$$\frac{v \text{ value}}{(\lambda x.t)v \to_{\beta_v} t\{x{\leftarrow}v\}} \qquad \frac{t \to_{\beta_v} t'}{\lambda x.t \to_{\beta_v} \lambda x.t'} \qquad \frac{t \to_{\beta_v} t'}{ts \to_{\beta_v} t's} \qquad \frac{t \to_{\beta_v} t'}{st \to_{\beta_v} st'}$$

INDEXED PARALLEL $\beta_v$ REDUCTION

$$\frac{}{x \overset{0}{\Rightarrow}_{\beta_v} x} \qquad \frac{t \overset{n}{\Rightarrow}_{\beta_v} t'}{\lambda x.t \overset{n}{\Rightarrow}_{\beta_v} \lambda x.t'} \qquad \frac{t \overset{n}{\Rightarrow}_{\beta_v} t' \quad s \overset{m}{\Rightarrow}_{\beta_v} s'}{ts \overset{n+m}{\Rightarrow}_{\beta_v} t's'} \qquad \frac{t \overset{n}{\Rightarrow}_{\beta_v} t' \quad v \overset{m}{\Rightarrow}_{\beta_v} v'}{(\lambda x.t)v \overset{n+|t'|_x\cdot m+1}{\Rightarrow}_{\beta_v} t'\{x{\leftarrow}v'\}}$$

The only difference with the usual parallel $\beta$ (defined in Sect. 3) is the requirement that the argument is a value in the last rule. As before, the non-indexed parallel reduction $\Rightarrow_{\beta_v}$ is simply obtained by erasing the index, so that $\Rightarrow_{\beta_v} = \bigcup_{n\in\mathbb{N}} \overset{n}{\Rightarrow}_{\beta_v}$. Similarly, it is easily seen that $\overset{0}{\Rightarrow}_{\beta_v}$ is the identity relation on terms, $\to_{\beta_v} = \overset{1}{\Rightarrow}_{\beta_v}$ and $\overset{n}{\Rightarrow}_{\beta_v} \subseteq \to_{\beta_v}^n$. Substitutivity of $\overset{n}{\Rightarrow}_{\beta_v}$ is proved exactly as for $\overset{n}{\Rightarrow}_\beta$ (Lemma 9).

**Lemma 14 (Substitutivity of $\overset{n}{\Rightarrow}_{\beta_v}$).** *If $t \overset{n}{\Rightarrow}_{\beta_v} t'$ and $v \overset{m}{\Rightarrow}_{\beta_v} v'$, then $t\{x \leftarrow v\} \overset{k}{\Rightarrow}_{\beta_v} t'\{x \leftarrow v'\}$ where $k = n + |t'|_x \cdot m$.* *Proof p. 25*

## 6 Weak Call-by-Value Reduction, Essentially

The essential step we study for the CbV $\lambda$-calculus is weak CbV reduction $\to_w$, which does not evaluate function bodies (the scope of $\lambda$-abstractions). Weak CbV reduction has practical importance, because it is the base of the ML/CAML family of functional programming languages. We choose it also because it admits the natural and more general *non-deterministic* presentation that follows, even if most of the literature rather presents it in a deterministic way.

Weak CbV reduction

$$\frac{}{(\lambda x.t)v \to_w t\{x \leftarrow v\}} \qquad \frac{t \to_w t'}{ts \to_w t's} \qquad \frac{t \to_w t'}{st \to_w st'}$$

Note that in the case of an application there is no fixed order in the $\to_w$-reduction of the left and right subterms. Such a non-determinism is harmless as $\to_w$ satisfies a diamond-like property implying confluence, see Prop. 17.2 below. It is well-known that the diamond property implies uniform termination, because it implies that all maximal sequences from a term have the same length. Such a further property is known as *random descent*, a special form of uniform termination already considered by Newman [21] in 1942, see also van Oostrom [22].

The inessential reduction $\to_{\neg w}$ and its parallel version $\Rightarrow_{\neg w}$ are defined by:

$\neg$Weak reduction

$$\frac{t \to_{\beta_v} s}{\lambda x.t \to_{\neg w} \lambda x.s} \qquad \frac{t \to_{\neg w} t'}{ts \to_{\neg w} t's} \qquad \frac{t \to_{\neg w} t'}{st \to_{\neg w} st'}$$

Parallel $\neg$weak reduction

$$\frac{}{x \Rightarrow_{\neg w} x} \qquad \frac{t \Rightarrow_{\beta_v} t'}{\lambda x.t \Rightarrow_{\neg w} \lambda x.t'} \qquad \frac{t \Rightarrow_{\neg w} t' \qquad s \Rightarrow_{\neg w} s'}{ts \Rightarrow_{\neg w} t's'}$$

It is immediate to check that $\to_{\beta_v} = \to_w \cup \to_{\neg w}$ and $\to_{\neg w} \subseteq \Rightarrow_{\neg w} \subseteq \to^*_{\neg w}$.

*Weak CbV factorization.* We show that $(\Lambda, \{\to_w, \to_{\neg w}\})$ is a macro-step system, with $\Rightarrow_{\beta_v}, \Rightarrow_{\neg w}$ as macro-steps. Merge and split are proved exactly as in Sect. 4.

**Proposition 15 (Weak CbV macro-step system).** *Proof p. 26*

1. Merge*: if $t \Rightarrow_{\neg w} \cdot \to_w u$ then $t \Rightarrow_{\beta_v} u$.*
2. Indexed split*: if $t \overset{n}{\Rightarrow}_{\beta_v} s$ then $t \Rightarrow_{\neg w} s$, or $n > 0$ and $t \to_w \cdot \overset{n-1}{\Rightarrow}_{\beta_v} s$.*
3. Split*: if $t \Rightarrow_{\beta_v} s$ then $t \to^*_w \cdot \Rightarrow_{\neg w} s$.*

*That is, $(\Lambda, \{\to_w, \to_{\neg w}\})$ is a macro-step system with respect to $\Rightarrow_{\beta_v}$ and $\Rightarrow_{\neg w}$.*

**Theorem 16 (Weak CbV factorization).** *If $t \to^*_{\beta_v} s$ then $t \to^*_w \cdot \to^*_{\neg w} s$.*

*Plotkin's left reduction.* The same argument at work in this section adapts easily to factorization with respect to leftmost weak reduction (used by Plotkin [27]), or to rightmost weak reduction, the two natural deterministic variants of $\to_w$.

*Weak CbV normalization.* To obtain a normalization theorem for $\to_w$ via the essential normalization theorem (Theorem 8), we need persistence and uniform termination. The latter follows from the well-known diamond property of $\to_w$.

*Proof p. 27* **Proposition 17 (Weak CbV essential system).**

1. Persistence*: if $t \to_w s$ and $t \to_{\neg w} u$ then $u \to_w r$ for some $r$.*
2. Diamond*: if $s \ {}_w\!\leftarrow \cdot \to_w u$ with $s \neq u$ then $s \to_w \cdot \ {}_w\!\leftarrow u$.*

*Then, $(\Lambda, \{\to_w, \to_{\neg w}\})$ is an essential system.*

**Theorem 18 (Weak CbV normalization).** *If $t \to^*_{\beta_v} s$ and $s$ is a $\to_w$-normal form, then $t$ is strongly $\to_w$-normalizing.*

CbV is often considered with respect to *closed* terms only. In such a case the $\to_w$-normal forms are exactly the (closed) values. Then weak CbV normalization (Theorem 18) implies the following, analogous to Corollary 1 in Plotkin [27] (the result is there obtained from standardization).

**Corollary 19.** *Let $t$ be a closed term. If $t \to^*_{\beta_v} v$ for some value $v$, then every maximal $\to_w$-sequence from $t$ is finite and ends in a value.*

## 7 Leftmost-Outermost Reduction, Essentially

Here we apply our technique to leftmost-outermost (shortened to $\ell o$) reduction $\to_{\ell o}$, the first example of *full* reduction for $\to_\beta$. The technical development is slightly different from the ones in the previous sections, as factorization relies on persistence. The same shall happen for the full $\ell\ell$ reduction of the next section. It seems to be a feature of full reductions for $\to_\beta$.

*$\ell o$ and $\neg\ell o$ reductions.* The definition of $\ell o$ reduction relies on two mutually recursive predicates defining normal and neutral terms (neutral = normal and not an abstraction):

Normal and neutral terms

$$\frac{}{x \text{ is neutral}} \qquad \frac{t \text{ is neutral} \quad t \text{ is normal}}{ts \text{ is neutral}} \qquad \frac{t \text{ is neutral}}{t \text{ is normal}} \qquad \frac{t \text{ is normal}}{\lambda x.t \text{ is normal}}$$

Dually, a term is not neutral if it is an abstraction or it is not normal. It is standard that these predicates correctly capture $\beta$ normal forms and neutrality.

The reductions of the $\ell o$ macro-step system are:

$\ell o$ REDUCTION

$$\frac{}{(\lambda x.t)s \to_{\ell o} t\{x \leftarrow s\}} \qquad \frac{t \to_{\ell o} s \qquad t \neq \lambda x.t'}{tu \to_{\ell o} su}$$

$$\frac{t \to_{\ell o} s}{\lambda x.t \to_{\ell o} \lambda x.s} \qquad \frac{u \text{ is neutral} \qquad t \to_{\ell o} s}{ut \to_{\ell o} us}$$

$\neg\ell o$ REDUCTION

$$\frac{t \to_{\beta} t'}{(\lambda x.t)s \to_{\neg\ell o} (\lambda x.t')s} \qquad \frac{t \text{ is not neutral} \qquad s \to_{\beta} s'}{ts \to_{\neg\ell o} ts'}$$

$$\frac{t \to_{\neg\ell o} t'}{ts \to_{\neg\ell o} t's} \qquad \frac{t \to_{\neg\ell o} t'}{st \to_{\neg\ell o} st'} \qquad \frac{t \to_{\neg\ell o} t'}{\lambda x.t \to_{\neg\ell o} \lambda x.t'}$$

PARALLEL $\neg\ell o$ REDUCTION

$$\frac{}{x \Rightarrow_{\neg\ell o} x} \qquad \frac{t \text{ is not neutral} \qquad t \Rightarrow_{\neg\ell o} t' \qquad s \Rightarrow_{\beta} s'}{ts \Rightarrow_{\neg\ell o} t's'}$$

$$\frac{t \Rightarrow_{\beta} t' \qquad s \Rightarrow_{\beta} s''}{(\lambda x.t)s \Rightarrow_{\neg\ell o} (\lambda x.t')s'} \qquad \frac{t \Rightarrow_{\neg\ell o} t'}{\lambda x.t \Rightarrow_{\neg\ell o} \lambda x.t'} \qquad \frac{t \text{ neutral} \qquad s \Rightarrow_{\neg\ell o} s'}{ts \Rightarrow_{\neg\ell o} ts'}$$

As usual, easy inductions show that $\to_{\beta} = \to_{\ell o} \cup \to_{\neg\ell o}$ and $\to_{\neg\ell o} \subseteq \Rightarrow_{\neg\ell o} \subseteq \to^*_{\neg\ell o}$.

Factorization depends on persistence, which is why for $\ell o$ reduction most essential properties are proved before factorization. The proofs are easy inductions.

**Proposition 20 ($\ell o$ essential properties).** *Proof p. 28*

1. Fullness*: if $t \to_{\beta} s$ then there exists $u$ such that $t \to_{\ell o} u$.*
2. Determinism*: if $t \to_{\ell o} s_1$ and $t \to_{\ell o} s_2$ then $s_1 = s_2$.*
3. Persistence*: if $t \to_{\ell o} s_1$ and $t \to_{\neg\ell o} s_2$ then $s_2 \to_{\ell o} u$ for some $u$.*

**Proposition 21 ($\ell o$ macro-step system).** *Proof p. 30*

1. Merge*: if $t \Rightarrow_{\neg\ell o} \cdot \to_{\ell o} u$ then $t \Rightarrow_{\beta} u$.*
2. Indexed split*: if $t \overset{n}{\Rightarrow}_{\beta} s$ then $t \Rightarrow_{\neg\ell o} s$, or $n > 0$ and $t \to_{\ell o} \cdot \overset{n-1}{\Rightarrow}_{\beta} s$.*
3. Split*: if $t \Rightarrow_{\beta} s$ then $t \to^*_{\ell o} \cdot \Rightarrow_{\neg\ell o} s$.*

*That is, $(\Lambda, \{\to_{\ell o}, \to_{\neg\ell o}\})$ is a macro-step system with respect to $\Rightarrow_{\beta}$ and $\Rightarrow_{\neg\ell o}$.*

*Proof.* We only show the merge property, and only the case that requires persistence—the rest of the proof is in the Appendix. The proof of the merge property is by induction on $t \Rightarrow_{\neg\ell o} s$. Consider the rule

$$\frac{r \text{ not neutral} \qquad r \Rightarrow_{\neg\ell o} r' \qquad p \Rightarrow_{\beta} p'}{t = rp \Rightarrow_{\neg\ell o} r'p' = s}\ .$$

Since $r$ is not neutral, it is an abstraction or it is not normal. If $r$ is an abstraction this case continues as the easy case of $\Rightarrow_{\neg\ell o}$ for $\beta$-redexes (see the Appendix).

Otherwise, $r$ is not normal, *i.e.* $r \to_\beta q$. By fullness $r \to_{\ell o} q'$ for some $q'$, and by persistence (Prop. 20.3) $r' \to_{\ell o} r''$ for some $r''$. The hypothesis becomes $t = rp \Rightarrow_{\neg\ell o} r'p' \to_{\ell o} r''p' = u$ with $r \Rightarrow_{\neg\ell o} r' \to_{\ell o} r''$. By *i.h.*, $r \Rightarrow_\beta r''$. Then,

$$\frac{r \Rightarrow_\beta r'' \qquad p \Rightarrow_\beta p'}{t = rp \Rightarrow_\beta r''p' = u}\ . \qquad \square$$

*$\ell o$ split.* As pointed out in Sect. 4.1, Takahashi's proof [29] of the split property relies on left substitutivity of head reduction, that is, if $t \to_h s$ then $t\{x \leftarrow u\} \to_h s\{x \leftarrow u\}$ for all terms $u$. Such a property does not hold for $\ell o$ reduction. For instance, $t = x(Iy) \to_{\ell o} xy = t'$ but $t\{x \leftarrow \lambda z.zz\} = (\lambda z.zz)(Iy) \not\to_{\ell o} (\lambda z.zz)y = t'\{x \leftarrow \lambda z.zz\}$. Therefore her proof technique for factorization cannot prove the factorization theorem for $\ell o$ reduction (see also footnote 5, page 3).

From Proposition 21 it follows the factorization theorem for $\ell o$ reduction, that together with Proposition 20 proves that $(\Lambda, \{\to_{\ell o}, \to_{\neg\ell o}\})$ is an essential system, giving normalization of $\to_{\ell o}$ for $\to_\beta$.

**Theorem 22.**

1. $\ell o$ factorization*: if $t \to_\beta^* u$ then $t \to_{\ell o}^* \cdot \to_{\neg\ell o}^* u$.*
2. $\ell o$ normalization*: $\to_{\ell o}$ is a normalizing reduction for $\to_\beta$.*

## 8 Least-Level Reduction, Essentially

In this section we study another normalizing full reduction for $\to_\beta$, namely *least-level* (shortened to $\ell\ell$) *reduction* $\to_{\ell\ell}$, which is non-deterministic. The intuition is that $\ell\ell$ reduction fires a $\beta$-redex of minimal level, where the *level* of a $\beta$-redex is the number of arguments containing it.

The definition of $\to_{\ell\ell}$ relies on an indexed variant $\to_{\beta:k}$ of $\to_\beta$, where $k \in \mathbb{N}$ is the level of the fired $\beta$-redex (do not mix it up with the index of $\overset{n}{\Rightarrow}_\beta$). We also define a parallel version $\Rightarrow_{\beta:n}$ (with $n \in \mathbb{N} \cup \{\infty\}$) of $\to_{\beta:k}$, obtained as a decoration of $\Rightarrow_\beta$, where $n$ is the minimal level of the $\beta$-redexes fired by a $\Rightarrow_\beta$ step ($\Rightarrow_{\beta:\infty}$ does not reduce any $\beta$-redex). From now on, $\mathbb{N} \cup \{\infty\}$ is considered with its usual order and arithmetic, that is, $\infty + 1 = \infty$.

$\beta$ reduction of level $k$

$$\frac{}{(\lambda x.t)s \to_{\beta:0} t\{x \leftarrow s\}} \qquad \frac{t \to_{\beta:k} t'}{\lambda x.t \to_{\beta:k} \lambda x.t'} \qquad \frac{t \to_{\beta:k} t'}{ts \to_{\beta:k} t's} \qquad \frac{t \to_{\beta:k} t'}{st \to_{\beta:k+1} st'}$$

Parallel $\beta$ reduction of least level $n$

$$\frac{t \Rightarrow_{\beta:k} t' \qquad s \Rightarrow_{\beta:h} s'}{(\lambda x.t)s \Rightarrow_{\beta:0} t'\{x \leftarrow s'\}} \qquad \frac{t \Rightarrow_{\beta:k} t'}{\lambda x.t \Rightarrow_{\beta:k} \lambda x.t'} \qquad \frac{t \Rightarrow_{\beta:k} t' \qquad s \Rightarrow_{\beta:h} s'}{ts \Rightarrow_{\beta:\min\{k,h+1\}} t's'} \qquad \frac{}{x \Rightarrow_{\beta:\infty} x}$$

Note that $t \to_\beta s$ if and only if $t \to_{\beta:k} s$ for some $k \in \mathbb{N}$.

The *least (reduction) level* $\ell\ell(t) \in \mathbb{N} \cup \{\infty\}$ of a term $t$ is defined as follows:

$$\ell\ell(x) = \infty \quad \ell\ell(\lambda x.t) = \ell\ell(t) \quad \ell\ell(ts) = \begin{cases} 0 & \text{if } t = \lambda x.u \\ \min\{\ell\ell(t), \ell\ell(s)+1\} & \text{otherwise.} \end{cases}$$

The definitions of $\ell\ell$, $\neg\ell\ell$, and *parallel* $\neg\ell\ell$ *reductions* are:

| | | |
|---|---|---|
| $\ell\ell$ REDUCTION | $t \to_{\ell\ell} s$ | if $t \to_{\beta:k} s$ with $\ell\ell(t) = k \in \mathbb{N}$; |
| $\neg\ell\ell$ REDUCTION | $t \to_{\neg\ell\ell} s$ | if $t \to_{\beta:k} s$ with $\ell\ell(t) < k \in \mathbb{N}$; |
| PARALLEL $\neg\ell\ell$ REDUCTION | $t \Rightarrow_{\neg\ell\ell} s$ | if $t \Rightarrow_{\beta:k} s$ with $k = \infty$ or $k > \ell\ell(t)$. |

As usual, easy inductions show that $\to_\beta = \to_{\ell\ell} \cup \to_{\neg\ell\ell}$ and $\to_{\neg\ell\ell} \subseteq \Rightarrow_{\neg\ell\ell} \subseteq \to^*_{\neg\ell\ell}$.

**Proposition 23 (Least level properties).** *Let $t$ be a term.* *Proof p. 32*

1. Computational meaning of $\ell\ell$: $\ell\ell(t) = \inf\{k \in \mathbb{N} \mid t \to_{\beta:k} u$ *for some term* $u\}$.
2. Monotonicity: *if* $t \to_\beta s$ *then* $\ell\ell(s) \geq \ell\ell(t)$.
3. Invariance by $\to_{\neg\ell\ell}$: *if* $t \to_{\neg\ell\ell} s$ *then* $\ell\ell(s) = \ell\ell(t)$.

Point 1 captures the meaning of the least level, and gives fullness of $\to_{\ell\ell}$. In particular, $\ell\ell(t) = \infty$ if and only if $t$ is $\to_\beta$-normal, since $\inf \emptyset = \infty$. Monotonicity states that $\beta$ steps cannot decrease the least level. Invariance by $\to_{\neg\ell\ell}$ says that $\to_{\neg\ell\ell}$ cannot change the least level. Essentially, this is persistence.

**Proposition 24 ($\ell\ell$ essential properties).** *Proof p. 33*

1. Fullness: *if* $t \to_\beta s$ *then* $t \to_{\ell\ell} u$ *for some* $u$.
2. Persistence: *if* $t \to_{\ell\ell} s_1$ *and* $t \to_{\neg\ell\ell} s_2$ *then* $s_2 \to_{\ell\ell} u$ *for some* $u$.
3. Diamond: *if* $s \,{}_{\ell\ell}\!\leftarrow \cdot \to_{\ell\ell} u$ *with* $s \neq u$ *then* $s \to_{\ell\ell} \cdot \,{}_{\ell\ell}\!\leftarrow u$.

As for $\ell o$, merge needs persistence, or, more precisely, invariance by $\to_{\neg\ell\ell}$.

**Proposition 25 ($\ell\ell$ macro-step system).** *Proof p. 35*

1. Merge: *if* $t \Rightarrow_{\neg\ell\ell} s \to_{\ell\ell} u$, *then* $t \Rightarrow_\beta u$.
2. Indexed split: *if* $t \overset{n}{\Rightarrow}_\beta s$ *then* $t \Rightarrow_{\neg\ell\ell} s$, *or* $n > 0$ *and* $t \to_{\ell\ell} \cdot \overset{n-1}{\Rightarrow}_\beta s$.
3. Split: *if* $t \Rightarrow_\beta s$ *then* $t \to^*_{\ell\ell} \cdot \Rightarrow_{\neg\ell\ell} s$.

*That is,* $(\Lambda, \{\to_{\ell\ell}, \to_{\neg\ell\ell}\})$ *is a macro-step system with respect to* $\Rightarrow_\beta$ *and* $\Rightarrow_{\neg\ell\ell}$.

**Theorem 26.**

1. $\ell\ell$ factorization: *if* $t \to^*_\beta u$ *then* $t \to^*_{\ell\ell} \cdot \to^*_{\neg\ell\ell} u$.
2. $\ell\ell$ normalization: $\to_{\ell\ell}$ *is a normalizing reduction for* $\to_\beta$.

*$\ell\ell$ split and $\ell o$ vs. $\ell\ell$.* As for $\ell o$ reduction, left substitutivity does not hold for $\to_{\ell\ell}$. Consider $t = x(Iy) \to_{\ell\ell} xy = t'$ where the step has level 1, and $t\{x{\leftarrow}\lambda z.zz\} = (\lambda z.zz)(Iy) \not\to_{\ell\ell} (\lambda z.zz)y = t'\{x{\leftarrow}\lambda z.zz\}$ since now there also is a step $(\lambda z.zz)(Iy) \to_{\ell\ell} (Iy)(Iy)$ at level 0.

Moreover, $\ell\ell$ and $\ell o$ reductions are incomparable. First, note that $\to_{\ell\ell} \not\subseteq \to_{\ell o}$: $t = (\lambda x.II)y \to_{\ell\ell} (\lambda x.I)y = s$, because $t \to_{\beta:0} (\lambda x.I)y$ and $\ell\ell(t) = 0$, but $t \not\to_{\ell o} s$, indeed $t \to_{\ell o} II$. This fact also shows that $\to_{\ell\ell}$ is not left–outer in the sense of van Oostrom and Toyama [23]. Second, $\to_{\ell o} \not\subseteq \to_{\ell\ell}$: $t = x(x(II))(II) \to_{\ell o} x(xI)(II) = s$ but $t \not\to_{\ell\ell} s$, indeed $t \to_{\neg\ell\ell} s$ because $t \to_{\beta:2} s$ and $\ell\ell(t) = 1$, and $t \to_{\ell\ell} x(x(II))I \neq s$.

## 9 Conclusions

We provide simple proof techniques for factorization and normalization theorems in the $\lambda$-calculus, simplifying Takahashi's parallel method [29], extending its scope and making it more abstract at the same time.

About the use of parallel reduction, Takahashi claims: "*once the idea is stated properly, the essential part of the proof is almost over, because the inductive verification of the statement is easy, even mechanical*" [29, p. 122]. Our work reinforces this point of view, as our case studies smoothly follow the abstract schema.

*Range of application.* We apply our method for factorization and normalization to two notions of reductions that compute full normal forms:

- the classic example of $\ell o$ reduction, covered also by the recent techniques by Hirokawa, Middeldorp and Moser [13] and van Oostrom and Toyama [23];
- $\ell\ell$ reduction, which is out of the scope of [13,23] because it is neither deterministic (as required by [13]), nor left–outer in the sense of [23] (as pointed out here in Sect. 8).

Our approach naturally covers also reductions that do not compute full normal forms, such as head and weak CbV reductions. These results are out of reach for van Oostrom and Toyama's technique [23], as they clarify in their conclusions.

Because of the minimality of our assumptions, we believe that our method applies to a large variety of other cases and variants of the $\lambda$-calculus. A key feature of our approach is that it derives normalization from factorization. However, it is worth noting that factorization is not a necessary condition for normalization.[10]

[10] For instance, in the weak $\lambda$-calculus—where weak $\beta$-reduction $\to_{w\beta}$ does not reduce under abstractions—our technique does not apply because weak head reduction $\to_{wh}$ (*i.e.* head reduction that does not reduce under abstractions) satisfies a weak head normalization theorem (if $t \to^*_{w\beta} s$ with $s \to_{wh}$-normal then $\to_{wh}$ terminates on $t$) but does not factorize: indeed, given the $\to_{w\beta}$-sequence $(\lambda x.\lambda y.x)(II) \to_{\neg wh} (\lambda x.\lambda y.x)I \to_{wh} \lambda y.I$, there is no term $t$ such that $(\lambda x.\lambda y.x)(II) \to^*_{wh} t \to^*_{\neg wh} \lambda y.I$.

*Acknowledgments.* This work has been partially funded by the ANR JCJC grant COCA HOLA (ANR-16-CE40-004-01) and by the EPSRC grant EP/R029121/1 "Typed Lambda-Calculi with Sharing and Unsharing".

## References

1. Accattoli, B.: An abstract factorization theorem for explicit substitutions. In: 23rd International Conference on Rewriting Techniques and Applications, RTA 2012. LIPIcs, vol. 15, pp. 6–21 (2012), https://doi.org/10.4230/LIPIcs.RTA.2012.6
2. Accattoli, B., Dal Lago, U.: (Leftmost-Outermost) Beta-Reduction is Invariant, Indeed. Logical Methods in Computer Science **12**(1) (2016), https://doi.org/10.2168/LMCS-12(1:4)2016
3. Accattoli, B., Faggian, C., Guerrieri, G.: Factorization and normalization, essentially. In: APLAS 2019 (2019)
4. Accattoli, B., Faggian, C., Guerrieri, G.: Factorization and Normalization, Essentially (extended version). CoRR **abs/1902.05945** (2019), http://arxiv.org/abs/1908.11289
5. Barendregt, H.P.: The Lambda Calculus – Its Syntax and Semantics, Studies in logic and the foundations of mathematics, vol. 103. North-Holland (1984)
6. Bonelli, E., Kesner, D., Lombardi, C., Ríos, A.: On abstract normalisation beyond neededness. Theor. Comput. Sci. **672**, 36–63 (2017), https://doi.org/10.1016/j.tcs.2017.01.025
7. de Carvalho, D., Pagani, M., Tortora de Falco, L.: A semantic measure of the execution time in linear logic. Theor. Comput. Sci. **412**(20), 1884–1902 (2011), https://doi.org/10.1016/j.tcs.2010.12.017
8. Crary, K.: A simple proof of call-by-value standardization. Tech. Rep. CMU-CS-09-137, Carnegie Mellon University (2009)
9. Guerrieri, G.: Head reduction and normalization in a call-by-value lambda-calculus. In: 2nd International Workshop on Rewriting Techniques for Program Transformations and Evaluation, WPTE 2015. OASICS, vol. 46, pp. 3–17 (2015), https://doi.org/10.4230/OASIcs.WPTE.2015.3
10. Guerrieri, G., Paolini, L., Ronchi Della Rocca, S.: Standardization of a Call-By-Value Lambda-Calculus. In: 13th International Conference on Typed Lambda Calculi and Applications, TLCA 2015. LIPIcs, vol. 38, pp. 211–225 (2015), https://doi.org/10.4230/LIPIcs.TLCA.2015.211
11. Guerrieri, G., Paolini, L., Ronchi Della Rocca, S.: Standardization and conservativity of a refined call-by-value lambda-calculus. Logical Methods in Computer Science **13**(4) (2017), https://doi.org/10.23638/LMCS-13(4:29)2017
12. Hindley, J.: The Church-Rosser Property and a Result in Combinatory Logic. Ph.D. thesis, University of Newcastle-upon-Tyne (1964)
13. Hirokawa, N., Middeldorp, A., Moser, G.: Leftmost outermost revisited. In: 26th International Conference on Rewriting Techniques and Applications, RTA 2015. LIPIcs, vol. 36, pp. 209–222 (2015), https://doi.org/10.4230/LIPIcs.RTA.2015.209
14. Ishii, K.: A proof of the leftmost reduction theorem for $\lambda\beta\eta$-calculus. Theor. Comput. Sci. **747**, 26–32 (2018), https://doi.org/10.1016/j.tcs.2018.06.003
15. Kesner, D., Lombardi, C., Ríos, A.: A standardisation proof for algebraic pattern calculi. In: 5th International Workshop on Higher-Order Rewriting, HOR 2010. EPTCS, vol. 49, pp. 58–72 (2010), https://doi.org/10.4204/EPTCS.49.5

16. Klop, J.W.: Combinatory Reduction Systems. Phd thesis, Utrecht University (1980)
17. Krivine, J.: Lambda-calculus, types and models. Ellis Horwood series in computers and their applications, Masson (1993)
18. McKinna, J., Pollack, R.: Some lambda calculus and type theory formalized. J. Autom. Reasoning **23**(3-4), 373–409 (1999), https://doi.org/10.1007/BFb0026981
19. Melliès, P.A.: A factorisation theorem in rewriting theory. In: Category Theory and Computer Science, 7th International Conference, CTCS '97. Lecture Notes in Computer Science, vol. 1290, pp. 49–68 (1997), https://doi.org/10.1007/BFb0026981
20. Mitschke, G.: The standardization theorem for $\lambda$-calculus. Mathematical Logic Quarterly **25**(1-2), 29–31 (1979), https://doi.org/10.1002/malq.19790250104
21. Newman, M.: On theories with a combinatorial definition of "Equivalence". Annals of Mathematics **43**(2), 223–243 (1942)
22. van Oostrom, V.: Random descent. In: Term Rewriting and Applications, 18th International Conference, RTA 2007. Lecture Notes in Computer Science, vol. 4533, pp. 314–328 (2007), https://doi.org/10.1007/978-3-540-73449-9_24
23. van Oostrom, V., Toyama, Y.: Normalisation by random descent. In: 1st International Conference on Formal Structures for Computation and Deduction, FSCD 2016. LIPIcs, vol. 52, pp. 32:1–32:18 (2016), https://doi.org/10.4230/LIPIcs.FSCD.2016.32
24. Pagani, M., Tranquilli, P.: Parallel reduction in resource lambda-calculus. In: Programming Languages and Systems, 7th Asian Symposium, APLAS 2009. Lecture Notes in Computer Science, vol. 5904, pp. 226–242 (2009), https://doi.org/10.1007/978-3-642-10672-9_17
25. Pagani, M., Tranquilli, P.: The conservation theorem for differential nets. Mathematical Structures in Computer Science **27**(6), 939–992 (2017), https://doi.org/10.1017/S0960129515000456
26. Paolini, L., Ronchi Della Rocca, S.: Parametric parameter passing lambda-calculus. Information and Computation **189**(1), 87–106 (2004), https://doi.org/10.1016/j.ic.2003.08.003
27. Plotkin, G.D.: Call-by-name, call-by-value and the lambda-calculus. Theor. Comput. Sci. **1**(2), 125–159 (1975), https://doi.org/10.1016/0304-3975(75)90017-1
28. Ronchi Della Rocca, S., Paolini, L.: The Parametric Lambda Calculus - A Metamodel for Computation. Texts in Theoretical Computer Science. An EATCS Series, Springer (2004), https://doi.org/10.1007/978-3-662-10394-4
29. Takahashi, M.: Parallel reductions in $\lambda$-calculus. Inf. Comput. **118**(1), 120–127 (1995), https://doi.org/10.1006/inco.1995.1057
30. Terese: Term Rewriting Systems, Cambridge Tracts in Theoretical Computer Science, vol. 55. Cambridge University Press (2003)
31. Terui, K.: Light affine lambda calculus and polynomial time strong normalization. Arch. Math. Log. **46**(3-4), 253–280 (2007), https://doi.org/10.1007/s00153-007-0042-6

# A Technical appendix: omitted proofs and lemmas

The enumeration of propositions, theorems, lemmas already stated in the body of the article is unchanged.

## A.1 Omitted proofs of Sect. 4 (head)

**Proposition 10** (Head macro-step system)**.** *See p. 12*

1. Merge*: if* $t \Rightarrow_{\neg h} \cdot \to_h u$ *then* $t \Rightarrow_\beta u$*.*
2. Indexed split*: if* $t \overset{n}{\Rightarrow}_\beta s$ *then* $t \Rightarrow_{\neg h} s$*, or* $n > 0$ *and* $t \to_h \cdot \overset{n-1}{\Rightarrow}_\beta s$*.*
3. Split*: if* $t \Rightarrow_\beta s$ *then* $t \to_h^* \cdot \Rightarrow_{\neg h} s$*.*

*that is,* $(\Lambda, \{\to_h, \to_{\neg h}\})$ *is a macro-step system with respect to* $\Rightarrow_\beta, \Rightarrow_{\neg h}$*.*

*Proof.* Points 2-3 are already proved on p. 12. We prove Point 1 by induction on the definition of $t \Rightarrow_{\neg h} s$. Cases:

- *Variable*: $t = x \Rightarrow_{\neg h} x = s$. Then, there is no $u$ such that $s \to_h u$.
- *Abstraction*: $t = \lambda x.t' \Rightarrow_{\neg h} \lambda x.s' = s$ because $t' \Rightarrow_{\neg h} s'$. According to the definition of $s \to_h u$, necessarily $u = \lambda x.u'$ with $s' \to_h u'$. By *i.h.* applied to $t'$, we have $t' \Rightarrow_\beta u'$ and hence:

$$\frac{t' \Rightarrow_\beta u'}{t = \lambda x.t' \Rightarrow_\beta \lambda x.u' = u}.$$

- *Application*:

$$\frac{r \Rightarrow_{\neg h} r' \qquad p \Rightarrow_\beta p'}{t = rp \Rightarrow_{\neg h} r'p' = s} \tag{1}$$

  Sub-cases:
  1. $s = r'p' \to_h r''p' = u$ with $r' \to_h r''$; by *i.h.* applied to $r \Rightarrow_{\neg h} r' \to_h r''$, we have $r \Rightarrow_\beta r''$, and so (as $\Rightarrow_{\neg h} \subseteq \Rightarrow_\beta$)

$$\frac{r \Rightarrow_\beta r'' \qquad p \Rightarrow_\beta p'}{t = rp \Rightarrow_\beta r''p' = u};$$

  2. $s = (\lambda x.q')p' \to_h q'\{x \leftarrow p'\} = u$, which means that $r' = \lambda x.q'$ in (1). According to the definition of $r \Rightarrow_{\neg h} r'$, necessarily $r = \lambda x.q$ with $q \Rightarrow_{\neg h} q'$. Thus, (as $\Rightarrow_{\neg h} \subseteq \Rightarrow_\beta$)

$$\frac{q \Rightarrow_\beta q' \qquad p \Rightarrow_\beta p'}{t = (\lambda x.q)p \Rightarrow_\beta q'\{x \leftarrow p'\} = u}.$$

- *β-redex*:

$$\frac{r \Rightarrow_\beta r' \qquad p \Rightarrow_\beta p'}{t = (\lambda x.r)p \Rightarrow_{\neg h} (\lambda x.r')p' = s}$$

  According to the definition of $s \to_h u$, we have $u = r'\{x \leftarrow p'\}$. Hence,

$$\frac{r \Rightarrow_\beta r' \qquad p \Rightarrow_\beta p'}{t = (\lambda x.r)p \Rightarrow_\beta r'\{x \leftarrow p'\} = u}.$$

□

*See p. 13* **Proposition 12** (Head essential system)**.**

1. Persistence*: if $t \to_h s$ and $t \to_{\neg h} u$ then $u \to_h r$ for some $r$.*
2. Determinism*: if $t \to_h s_1$ and $t \to_h s_2$ then $s_1 = s_2$.*

*Then, $(\Lambda, \{\to_w, \to_{\neg w}\})$ is an essential system.*

*Proof.*

1. *Persistence*: By induction on $t \to_h s$. Cases:
   - *Root step, i.e.* $t = (\lambda x.p)q \to_h p\{x \leftarrow q\} = s$. Sub-cases:
     - $\to_{\neg h}$ *in the left sub-term*: $t = (\lambda x.p)q \to_{\neg h} (\lambda x.p')q = u$ because $p \to_\beta p'$ or $p \to_{\neg h} p'$. In both cases $u = (\lambda x.p')q \to_h p'\{x \leftarrow q\} =: r$.
     - $\to_{\neg h}$ *in the right sub-term*: $t = (\lambda x.p)q \to_{\neg h} (\lambda x.p)q' = u$ because $q \to_\beta q'$. Then $u = (\lambda x.p)q' \to_h p\{x \leftarrow q'\} =: r$.
   - *Application*: *i.e.* $t = pq \to_h p'q = s$ with $p \to_h p'$ and $p$ not an abstraction. Sub-cases:
     - $\to_{\neg h}$ *in the left sub-term*: $t = pq \to_{\neg h} p''q = u$ because $p \to_{\neg h} p''$. By *i.h.*, $p'' \to_h r'$ for some $r'$. Then $u = p''q \to_h r'q =: r$.
     - $\to_{\neg h}$ *in the right sub-term*: $t = pq \to_{\neg h} pq' = u$ because $q \to_\beta q'$. Then $u = pq' \to_h p'q' =: r$.
   - *Abstraction*: *i.e.* $t = \lambda x.p \to_h p' = s$. Then $t = \lambda x.p \to_{\neg h} \lambda x.p'' = u$ for some $p''$ with $p \to_{\neg h} p''$. By *i.h.*, $p'' \to_h r'$ for some $r'$. Then $u = \lambda x.p'' \to_h \lambda x.r'$.
2. *Determinism*: By induction on a derivation with conclusion $t \to_h s_1$. Consider its last rule. Cases:
   - *Abstraction*, *i.e.* $t = \lambda x.t' \to_h \lambda x.s_1' = s_1$ because $t' \to_h s_1'$. According to the definition of $\to_h$, the only possibility for the last rule of a derivation for $t \to_h s_2$ is
   $$\frac{t' \to_h s_2'}{t = \lambda x.t' \to_h \lambda x.s_2' = s_2}\,.$$
   By *i.h.* applied to $t'$, we have $s_1' = s_2'$ and hence $s_1 = \lambda x.s_1' = \lambda x.s_2' = s_2$.
   - *Application*, *i.e.* $t = t't'' \to_h s_1't'' = s_1$ because $t' \to_h s_1'$ and $t' \neq \lambda x.r$. According to the definition of $\to_h$, the the last rule of a derivation for $t \to_h s_2$ can be only
   $$\frac{t' \to_h s_2'}{t = t't'' \to_h s_2't'' = s_2}\,.$$
   By *i.h.* applied to $t'$, we have $s_1' = s_2'$ and hence $s_1 = s_1't'' = s_2't'' = s_2$.
   - *$\beta$-rule*, *i.e.* $t = (\lambda x.t')t'' \to_h t'\{x \leftarrow t''\} = s_1$. According to the definition of $\to_h$, the only possibility for the last rule of a derivation for $t \to_h s_2$ is
   $$\frac{}{t = (\lambda x.t')t'' \to_h t'\{x \leftarrow t''\} = s_2} \quad \text{and hence } s_1 = t'\{x \leftarrow t''\} = s_2. \quad \square$$

### A.2 Omitted proofs of Sect. 5 (CbV λ-calculus)

**Lemma 14** (Substitutivity of $\overset{n}{\Rightarrow}_{\beta_v}$). *If* $t \overset{n}{\Rightarrow}_{\beta_v} t'$ *and* $v \overset{m}{\Rightarrow}_{\beta_v} v'$ *then* $t\{x{\leftarrow}v\} \overset{k}{\Rightarrow}_{\beta_v} t'\{x{\leftarrow}v'\}$ *where* $k = n + |t'|_x \cdot m$. *See p. 15*

*Proof.* By induction on $t \overset{n}{\Rightarrow}_{\beta} t'$. It follows the exact same pattern of the proof of substitutivity of $\overset{n}{\Rightarrow}_{\beta}$. Cases:

- *Variable*: two sub-cases
  - $t = x$: then $t = x \overset{0}{\Rightarrow}_{\beta_v} x = t'$ then $t\{x{\leftarrow}v\} = x\{x{\leftarrow}v\} = v \overset{m}{\Rightarrow}_{\beta_v} v' = x\{x{\leftarrow}v'\} = t'\{x{\leftarrow}v'\}$ that satisfies the statement because $n + |t'|_x \cdot m = 0 + 1 \cdot m = m$.
  - $t = y$: then $t = y \overset{0}{\Rightarrow}_{\beta_v} y = t'$ then $t\{x{\leftarrow}v\} = y\{x{\leftarrow}v\} = y \overset{0}{\Rightarrow}_{\beta_v} y = y\{x{\leftarrow}v'\} = t'\{x{\leftarrow}v'\}$ that satisfies the statement because $n + |t'|_x \cdot m = n + 0 \cdot m = n$.
- *Abstraction, i.e.* $t = \lambda y.u \overset{n}{\Rightarrow}_{\beta_v} \lambda y.u' = t'$ because $u \overset{n}{\Rightarrow}_{\beta_v} u'$; we can suppose without loss of generality that $y \neq x$, hence $|u'|_x = |t'|_x$ and $t\{x{\leftarrow}v\} = \lambda y.(u\{x{\leftarrow}v\})$ and $t'\{x{\leftarrow}v'\} = \lambda y.(u'\{x{\leftarrow}v'\})$. By *i.h.*, $u\{x{\leftarrow}v\} \overset{n+|u'|_x \cdot m}{\Rightarrow}_{\beta_v} u'\{x{\leftarrow}v'\}$, thus

$$\frac{u\{x{\leftarrow}v\} \overset{n+|u'|_x \cdot m}{\Rightarrow}_{\beta_v} u'\{x{\leftarrow}v'\}}{t\{x{\leftarrow}v\} = \lambda y.u\{x{\leftarrow}v\} \overset{n+|t'|_x \cdot m}{\Rightarrow}_{\beta_v} \lambda y.u'\{x{\leftarrow}v'\} = t'\{x{\leftarrow}v'\}}.$$

- *Application, i.e.*

$$\frac{u \overset{n_u}{\Rightarrow}_{\beta_v} u' \qquad r \overset{n_r}{\Rightarrow}_{\beta_v} r'}{t = ur \overset{n_u+n_r}{\Rightarrow}_{\beta_v} u'r' = t'}$$

  with $n = n_u + n_r$. By *i.h.*, $u\{x{\leftarrow}v\} \overset{n_u+|u'|_x \cdot m}{\Rightarrow}_{\beta_v} u'\{x{\leftarrow}v'\}$ and $r\{x{\leftarrow}v\} \overset{n_r+|r'|_x \cdot m}{\Rightarrow}_{\beta_v} r'\{x{\leftarrow}v'\}$. Then

$$\frac{u\{x{\leftarrow}v\} \overset{n_u+|u'|_x \cdot m}{\Rightarrow}_{\beta_v} u'\{x{\leftarrow}v'\} \qquad r\{x{\leftarrow}v\} \overset{n_r+|r'|_x \cdot m}{\Rightarrow}_{\beta_v} r'\{x{\leftarrow}v'\}}{t\{x{\leftarrow}v\} = u\{x{\leftarrow}v\}r\{x{\leftarrow}v\} \overset{k}{\Rightarrow}_{\beta_v} u'\{x{\leftarrow}v'\}r'\{x{\leftarrow}v'\} = t'\{x{\leftarrow}v'\}}$$

  with $k = n_u + |u'|_x \cdot m + n_r + |r'|_x \cdot m$, which proves the statement because

$$k = n_u + |u'|_x \cdot m + n_r + |r'|_x \cdot m = n + (|u'|_x + |r'|_x) \cdot m = n + |t'|_x \cdot m.$$

- $\beta_v$*-step, i.e.* ($w$ and $w'$ are values)

$$\frac{u \overset{n_u}{\Rightarrow}_{\beta_v} u' \qquad w \overset{n_w}{\Rightarrow}_{\beta_v} w'}{t = (\lambda y.u)w \overset{n_u+|u'|_y \cdot n_w+1}{\Rightarrow}_{\beta_v} u'\{y{\leftarrow}w'\} = t'}$$

  with $n = n_u + |u'|_y \cdot n_w + 1$. We can assume without loss of generality that $y \neq x$, hence, $|t'|_x = |u'\{y{\leftarrow}w'\}|_x = |u'|_x + |u'|_y \cdot |w'|_x$ and $t\{x{\leftarrow}v\} = (\lambda y.u\{x{\leftarrow}v\})(w\{x{\leftarrow}v\})$ and $t'\{x{\leftarrow}v'\} = u'\{x{\leftarrow}v'\}\{y{\leftarrow}w'\{x{\leftarrow}v'\}\}$.

By *i.h.*, $u\{x{\leftarrow}v\} \overset{n_u+|u'|_x\cdot m}{\Rightarrow_{\beta_v}} u'\{x{\leftarrow}v'\}$ and $w\{x{\leftarrow}v\} \overset{n_w+|w'|_x\cdot m}{\Rightarrow_{\beta_v}} w'\{x{\leftarrow}v'\}$. Then

$$\frac{u\{x{\leftarrow}v\} \overset{n_u+|u'|_x\cdot m}{\Rightarrow_{\beta_v}} u'\{x{\leftarrow}v'\} \qquad w\{x{\leftarrow}v\} \overset{n_w+|w'|_x\cdot m}{\Rightarrow_{\beta_v}} w'\{x{\leftarrow}v'\}}{t\{x{\leftarrow}v\} = (\lambda y.u\{x{\leftarrow}v\})(w\{x{\leftarrow}v\}) \overset{k}{\Rightarrow_{\beta_v}} u'\{x{\leftarrow}v'\}\{y{\leftarrow}w'\{x{\leftarrow}v'\}\} = t'\{x{\leftarrow}v'\}}$$

where $k = n_u+|u'|_x\cdot m+|u'|_y\cdot(n_w+|w'|_x\cdot m)+1 = n_u+|u'|_x\cdot m+|u'|_y\cdot(n_w+|w'|_x\cdot m)+1 = n_u+|u'|_y\cdot n_w+1+|u'|_x\cdot m+|u'|_y\cdot|w'|_x\cdot m = n+|t'|_x\cdot m$. □

### A.3 Omitted proofs of Sect. 6 (weak CbV)

*See p. 15* **Proposition 15** (Weak CbV macro-step system)**.**

1. Merge*: if $t \Rightarrow_{\neg w} \cdot \to_w u$ then $t \Rightarrow_{\beta_v} u$.*
2. Indexed split*: if $t \overset{n}{\Rightarrow_{\beta_v}} s$ then $t \Rightarrow_{\neg w} s$, or $n > 0$ and $t \to_w \cdot \overset{n-1}{\Rightarrow_{\beta_v}} s$.*
3. Split*: if $t \Rightarrow_{\beta_v} s$ then $t \to_w^* \cdot \Rightarrow_{\neg w} s$.*

*That is, $(\Lambda, \{\to_w, \to_{\neg w}\})$ is a macro-step system with respect to $\Rightarrow_{\beta_v}$ and $\Rightarrow_{\neg w}$.*

*Proof.*

1. *Merge*: by induction on $t \Rightarrow_{\neg w} s$. Note that the cases in which $s = x$ or $s = \lambda x.s'$ are not possible. Hence $s = r'p'$ and $t \Rightarrow_{\neg w} s$ is derived as follows

$$\frac{r \Rightarrow_{\neg w} r' \qquad p \Rightarrow_{\neg w} p'}{t = rp \Rightarrow_{\neg w} r'p' = s}$$

   (a) If $r' \to_w r''$ then $s = r''p'$. The *i.h.* gives $r \Rightarrow_{\beta_v} r''$, and $t \Rightarrow_{\beta_v} s$ is derived as follows (remember that $\Rightarrow_{\neg w} \subseteq \Rightarrow_{\beta_v}$):

$$\frac{r \Rightarrow_{\beta_v} r'' \qquad p \Rightarrow_{\beta_v} p'}{t = rp \Rightarrow_{\beta_v} r''p' = s}$$

   (b) If $p' \to_w p''$ it is analogous to the previous case.
   (c) If $s \to_w u$ by a top $\beta_v$ step then $r' = \lambda x.q'$. Now, by definition of $\Rightarrow_{\neg w}$ the step $r \Rightarrow_{\neg w} r'$ necessarily has the form $r = \lambda x.q \Rightarrow_{\neg w} \lambda x.q' = r'$ for some $q$ such that $q \Rightarrow_{\beta_v} q'$. Then the hypothesis is $t = (\lambda x.q)p \Rightarrow_{\neg w} (\lambda x.q')p' \to_w q'\{x{\leftarrow}p'\} = u$ and $t \Rightarrow_{\beta_v} s$ is derived as follows (remember that $\Rightarrow_{\neg w} \subseteq \Rightarrow_{\beta_v}$):

$$\frac{q \Rightarrow_{\beta_v} q' \qquad p \Rightarrow_{\beta_v} p'}{t = (\lambda x.q)p \Rightarrow_{\beta_v} q'\{x{\leftarrow}p'\} = s}$$

2. *Indexed split*: by induction on the definition of $t \overset{n}{\Rightarrow_{\beta_v}} s$. We freely use the fact that if $t \overset{n}{\Rightarrow_{\beta_v}} s$ then $t \Rightarrow_{\beta_v} s$. Cases:
   – *Variable*: $t = x \overset{0}{\Rightarrow_{\beta_v}} x = s$. Then, $t = x \Rightarrow_{\neg w} x = s$.

– *Abstraction*: $t = \lambda x.t' \overset{n}{\Rightarrow}_{\beta_v} \lambda x.s' = s$ because $t' \overset{n}{\Rightarrow}_{\beta_v} s'$. Then,

$$\frac{t' \Rightarrow_{\beta_v} s'}{t = \lambda x.t' \Rightarrow_{\neg w} \lambda x.s' = s}.$$

– *Application*:

$$\frac{r \overset{n_1}{\Rightarrow}_{\beta_v} r' \qquad p \overset{n_2}{\Rightarrow}_{\beta_v} p'}{t = rp \overset{n_1+n_2}{\Rightarrow}_{\beta_v} r'p' = s}$$

with $n = n_1 + n_2$. There are only two cases:
- either $rp \Rightarrow_{\neg w} r'p'$, and then the claim holds.
- or $rp \not\Rightarrow_{\neg w} r'p'$, and then $r \not\Rightarrow_{\neg w} r'$ or $p \not\Rightarrow_{\neg w} p'$ (otherwise $rp \Rightarrow_{\neg w} r'p'$). Suppose $r \not\Rightarrow_{\neg w} r'$ (the other case is analogous). By *i.h.* applied to $r \overset{n_1}{\Rightarrow}_{\beta_v} r'$, $n_1 > 0$ and there is $r''$ such that $r \to_w r'' \overset{n_1-1}{\Rightarrow}_{\beta_v} r'$. So, $t = rp \to_w r''p$ and

$$\frac{r'' \overset{n_1-1}{\Rightarrow}_{\beta_v} r' \qquad p \overset{n_2}{\Rightarrow}_{\beta_v} p'}{r''p \overset{n_1-1+n_2}{\Rightarrow}_{\beta_v} r'p' = s}.$$

– $\beta_v$ *step*:

$$\frac{p \overset{n_1}{\Rightarrow}_{\beta_v} p' \qquad r \text{ is a value} \qquad r \overset{n_2}{\Rightarrow}_{\beta_v} r'}{t = (\lambda x.p)r \overset{n_1+|p'|_x \cdot n_2+1}{\Rightarrow}_{\beta_v} p'\{x \leftarrow r'\} = s}$$

with $n = n_1 + |p'|_x \cdot n_2 + 1 > 0$. We have $t = (\lambda x.p)r \to_w p\{x \leftarrow r\} := u$ and substitutivity of $\overset{n}{\Rightarrow}_{\beta_v}$ (Lemma 14) gives $u = p\{x \leftarrow r\} \overset{n_1+|p'|_x \cdot n_2}{\Rightarrow}_{\beta_v} p'\{x \leftarrow r'\} = s$.

3. *Split*: exactly as in the head case (Prop. 10.3), using the Indexed Split property for weak CbV (Point 2 above). □

**Proposition 17** (Weak CbV essential system)**.** *See p. 16*

1. Persistence*: if $t \to_w s$ and $t \to_{\neg w} u$ then $u \to_w r$ for some $r$.*
2. Diamond*: if $s \,{}_w\!\leftarrow t \to_w u$ with $s \neq u$ then $s \to_w r \,{}_w\!\leftarrow u$ for some $r$.*

*Then, $(\Lambda, \{\to_w, \to_{\neg w}\})$ is an essential system.*

*Proof.*

1. *Persistence*: By induciton on the definition of $t \to_{\neg w} s$. Cases:
   – *Abstraction*: $t = \lambda x.p \to_{\neg w} \lambda x.p'$ because $p \to_{\beta_v} p'$. This case is impossible because $t$ is $\to_w$ normal, against the hypothesis that $t \to_w u$.
   – *Application left*: $t = pq \to_{\neg w} p_{\neg w}q = s$ because $p \to_{\neg w} p_{\neg w}$. According to the definition of $\to_w$, there are the following sub-cases:
   (a) $t = pq \to_w p_w q = u$ with $p \to_w p_w$; by *i.h.* applied to $p$, we have $p_{\neg w} \to_w r'$ for some term $r'$, and so

$$\frac{p_{\neg w} \to_w r'}{s = p_{\neg w}q \to_w r'q =: r};$$

(b) $t = pq \to_w pq' = u$ with $q \to_w q'$; hence

$$\frac{q \to_w q'}{s = p_{\neg w}q \to_w p_{\neg w}q' = \text{`} : r}\ ;$$

(c) $t = (\lambda x.p')q \to_w p'\{x \leftarrow q\} = u$ where $p = \lambda x.p'$ and $q$ is a value. According to the definition of $t \to_{\neg w} s$,

$$\frac{\dfrac{p \to_{\beta_v} p_{\neg w}}{\lambda x.p' \to_{\neg w} \lambda x.p'_{\neg w}}}{t = (\lambda x.p')q \to_{\neg w} (\lambda x.p'_{\neg w})q = s}$$

where $p_{\neg w} = \lambda x.p'_{\neg w}$; therefore

$$\frac{}{s = (\lambda x.p'_{\neg w})q \to_w p'_{\neg w}\{x \leftarrow q\} =: r}\ .$$

– *Application right*: analogous to the previous point.

2. *Diamond*: The idea of the proof is that, when $(\lambda x.t)v \to_w t\{x \leftarrow v\}$, the $\beta_v$-redexes in $v$ (which is a value) can be duplicated in $t\{x \leftarrow v\}$ but they are under an abstraction and $\to_w$ does not reduce under abstractions.
Formally, the proof is by induction on $t$. Note that $t$ is not a value, because values are $\to_w$-normal, since $\to_w$ does not reduce under abstractions. Therefore, $t = t_0 t_1$. The case where $t_0 = \lambda x.t'$ and $t_1$ is a value is impossible, because $t_0$ and $t_1$ would be $\to_w$-normal and so from $s \ {}_w\!\leftarrow t \to_w u$ it would follow $s = t'\{x \leftarrow t_1\} = u$, which contradicts the hypothesis. The remaining cases for $t = t_0 t_1$ are:
   – $s = t_0 s_1 \ {}_w\!\leftarrow t = t_0 t_1 \to_w u_0 t_1 = u$ with $t_0 \to_w u_0$, and $t_1 \to_w s_1$. Then, $s \to_w u_0 s_1 \ {}_w\!\leftarrow u$.
   – $s = s_0 t_1 \ {}_w\!\leftarrow t = t_0 t_1 \to_w u_0 t_1 = u$ with $s_0 \ {}_w\!\leftarrow t_0 \to_w u_0$. By *i.h.*, there is $r_0$ such that $s_0 \to_w r_0 \ {}_w\!\leftarrow u_0$. Thus, $s \to_w r_0 t_1 \ {}_w\!\leftarrow u$.
   – $s = t_0 s_1 \ {}_w\!\leftarrow t = t_0 t_1 \to_w t_0 u_0 = u$ with $s_1 \ {}_w\!\leftarrow t_1 \to_w u_1$. Analogous to the previous case. □

### A.4 Omitted proofs of Sect. 7 (leftmost-outermost)

*See p. 17* **Proposition 20** ($\ell o$ essential properties)**.**

1. Fullness*: if $t \to_\beta s$ then there exists $u$ such that $t \to_{\ell o} u$.*
2. Determinism*: if $t \to_{\ell o} s_1$ and $t \to_{\ell o} s_2$, then $s_1 = s_2$.*
3. Persistence*: if $t \to_{\ell o} s_1$ and $t \to_{\neg \ell o} s_2$, then $s_2 \to_{\ell o} u$ for some $u$.*

*Proof.*

1. *Fullness*: by induction on $t$. Cases:
   – *Variable*, *i.e.* $t = x$: then $t \not\to_\beta s$, and so the statement trivially holds.
   – *Abstraction*, *i.e.* $t = \lambda x.t' \to_\beta \lambda x.s' = s$. It follows by the *i.h.*
   – *Application*, *i.e.* $t = rp$. Three sub-cases:
      • *$r$ is an abstraction*, *i.e.* $r = \lambda x.q$: then $t = (\lambda x.q)p \to_{\ell o} q\{x \leftarrow q\}$.

- *$r$ is not an abstraction but it is not normal*, *i.e.* $r \to_\beta r'$ for some $r'$: then by *i.h.* $r \to_{\ell o} q$ for some $q$ and so $t = rp \to_{\ell o} qp$.
- *$r$ is neutral*, *i.e.* $t$ is not normal implies $p$ not normal. Then by *i.h.* $p \to_{\ell o} p'$ for some $p'$, and so $t = rp \to_{\ell o} rp'$.

2. *Determinism*: By induction on a derivation with conclusion $t \to_{\ell o} s_1$. Consider its last rule. Cases:
   - *Abstraction*, *i.e.* $t = \lambda x.t' \to_{\ell o} \lambda x.s_1' = s_1$ because $t' \to_{\ell o} s_1'$. According to the definition of $\to_{\ell o}$, the last rule of a derivation for $t \to_{\ell o} s_2$ can be only
   $$\frac{t' \to_{\ell o} s_2'}{t = \lambda x.t' \to_{\ell o} \lambda x.s_2' = s_2}\,.$$
   By *i.h.* applied to $t'$, we have $s_1' = s_2'$ and hence $s_1 = \lambda x.s_1' = \lambda x.s_2' = s_2$.
   - *Application left*, *i.e.* $t = t't'' \to_{\ell o} s_1't'' = s_1$ because $t' \to_{\ell o} s_1'$ and $t' \neq \lambda x.r$. According to the definition of $\to_{\ell o}$, the last rule of a derivation for $t \to_{\ell o} s_2$ can only be (since $t$ is neither an abstraction nor neutral)
   $$\frac{t' \to_{\ell o} s_2'}{t = t't'' \to_{\ell o} s_2't'' = s_2}\,.$$
   By *i.h.* applied to $t'$, we have $s_1' = s_2'$ and hence $s_1 = s_1't'' = s_2't'' = s_2$.
   - *Application right*, *i.e.* $t = t't'' \to_{\ell o} t's_1'' = s_1$ because $t'' \to_{\ell o} s_1''$ and $t'$ is neutral. According to the definition of $\to_{\ell o}$, the last rule of a derivation for $t \to_{\ell o} s_2$ can only be (since $t$ is normal and not an abstraction)
   $$\frac{t'' \to_{\ell o} s_2''}{t = t't'' \to_{\ell o} t's_2'' = s_2}\,.$$
   By *i.h.* applied to $t'$, we have $s_1' = s_2'$ and hence $s_1 = t's_1'' = t's_2'' = s_2$.
   - *$\beta$-rule*, *i.e.* $t = (\lambda x.t')t'' \to_{\ell o} t'\{x{\leftarrow}t''\} = s_1$. According to the definition of $\to_{\ell o}$, the only possibility for the last rule of a derivation for $t \to_{\ell o} s_2$ is
   $$\frac{}{t = (\lambda x.t')t'' \to_{\ell o} t'\{x{\leftarrow}t''\} = s_2} \quad \text{and hence } s_1 = t'\{x{\leftarrow}t''\} = s_2. \quad \square$$

3. *Persistence*: by induction on $t \to_{\ell o} s_1$. Cases:
   - *Root*: $t = (\lambda x.r)p \to_{\ell o} r\{x{\leftarrow}p\} = s_1$. Three sub-cases:
     - $t = (\lambda x.r)p \to_{\neg\ell o} (\lambda x.r')p = s_2$ because $r \to_\beta r'$. Then $s_2 = (\lambda x.r')p \to_{\ell o} r'\{x{\leftarrow}p\} =: u$.
     - $t = (\lambda x.r)p \to_{\neg\ell o} (\lambda x.r')p = s_2$ because $r \to_{\neg\ell o} r'$. Exactly as the previous one.
     - $t = (\lambda x.r)p \to_{\neg\ell o} (\lambda x.r)p' = s_2$ because $p \to_\beta p'$. Then $s_2 = (\lambda x.r)p' \to_{\ell o} r\{x{\leftarrow}p\}' =: u$.
   - *Abstraction*: $t = \lambda x.r \to_{\ell o} \lambda x.r' = s_1$. Then $t = \lambda x.r \to_{\neg\ell o} \lambda x.r'' = s_2$ and the statement follows from the *i.h.* and closure of $\to_{\ell o}$.
   - *Left of an application*: $t = rp \to_{\ell o} r_1p = s_1$ with $r \to_{\ell o} r_1$ and $r$ not an abstraction. Two sub-cases:

- $t = rp \to_{\neg\ell o} r_2 p = s_2$ because $r \to_{\neg\ell o} r_2$. Then by *i.h.* there exists $q$ such that $r_2 \to_{\ell o} q$. Note that $\to_{\neg\ell o}$ cannot create a root abstraction (because it never reduces the root redex) so that if $r$ is not an abstraction then $r_2$ is not an abstraction and $s_2 = r_2 p \to_{\ell o} qp =: u$.
- $t = rp \to_{\neg\ell o} rp' = s_2$ by one of the two rules for able to derive it. In both cases $p \to_\beta p'$. Then $s_2 = rp' \to_{\ell o} r_1 p' =: u$ and $s_1 = r_1 p \to_\beta r_1 p' = u$.

– *Right of an application*: $t = rp \to_{\ell o} rp_1 = s_1$ with $p \to_{\ell o} p_1$ and $r$ neutral. Then necessarily $t = rp \to_{\ell o} rp_2 = s_2$ with $p \to_{\neg\ell o} p_2$. The statement then follows by the *i.h.*

*See p. 17* **Proposition 21** ($\ell o$ macro-step system)**.**

1. Merge*: if $t \Rightarrow_{\neg\ell o} \cdot \to_{\ell o} u$ then $t \Rightarrow_\beta u$.*
2. Indexed split*: if $t \overset{n}{\Rightarrow}_\beta s$ then $t \Rightarrow_{\neg\ell o} s$, or $n > 0$ and $t \to_{\ell o} \cdot \overset{n-1}{\Rightarrow}_\beta s$.*
3. Split*: if $t \Rightarrow_\beta s$ then $t \to_{\ell o}^* \cdot \Rightarrow_{\neg\ell o} s$.*

*That is, $(\Lambda, \{\to_{\ell o}, \to_{\neg\ell o}\})$ is a macro-step system with respect to $\Rightarrow_\beta$ and $\Rightarrow_{\neg\ell o}$.*

*Proof.*

1. *Merge*: by induction on $t \Rightarrow_{\neg\ell o} s$. Cases:
   – Rule
   $$\frac{r \Rightarrow_\beta r' \qquad p \Rightarrow_\beta p''}{t = (\lambda x.r)p \Rightarrow_{\neg\ell o} (\lambda x.r')p' = s}$$
   Then $(\lambda x.r)p \Rightarrow_{\neg\ell o} (\lambda x.r')p' \to_{\ell o} r'\{x \leftarrow p'\} = u$. We simply have:
   $$\frac{r \Rightarrow_\beta r' \qquad p \Rightarrow_\beta p''}{t = (\lambda x.r)p \Rightarrow_\beta r'\{x \leftarrow p'\} = u}$$
   – Rule
   $$\frac{r \text{ not neutral} \qquad r \Rightarrow_{\neg\ell o} r' \qquad p \Rightarrow_\beta p'}{t = rp \Rightarrow_{\neg\ell o} r'p' = s}$$
   Since $r$ is not neutral, it is an abstraction or it is not normal. If $r$ is an abstraction this case continues goes as the first case. Otherwise, $r$ is not normal, and by persistence (Prop. 20.3) $r'$ is not normal. Fullness (Prop. 20.1) of $\to_{\ell o}$ gives $r' \to_{\ell o} r''$ for a certain $r''$. The hypothesis becomes $t = rp \Rightarrow_{\neg\ell o} r'p' \to_{\ell o} r''p' = u$. By *i.h.*, $r \Rightarrow_\beta r''$. Then,
   $$\frac{r \Rightarrow_\beta r'' \qquad p \Rightarrow_\beta p'}{t = rp \Rightarrow_\beta r''p' = u}$$
   – Rule
   $$\frac{r \Rightarrow_{\neg\ell o} r'}{t = \lambda x.r \Rightarrow_{\neg\ell o} \lambda x.r' = s}$$
   Then $\lambda x.r \Rightarrow_{\neg\ell o} \lambda x.r' \to_{\ell o} \lambda x.r'' = u$ with $r' \to_{\ell o} r''$. By *i.h.*, $r \Rightarrow_\beta r''$ and
   $$\frac{r \Rightarrow_\beta r'}{t = \lambda x.r \Rightarrow_\beta \lambda x.r'' = u}$$

– Rule

$$\frac{r \text{ neutral} \qquad p \Rightarrow_{\neg\ell o} p'}{t = rp \Rightarrow_{\neg\ell o} rp' = s}$$

Then the hypothesis is $rp \Rightarrow_{\neg\ell o} rp' \to_{\ell o} rp'' = u$ with $p' \to_{\ell o} p''$. By *i.h.*, $p \Rightarrow_\beta p''$, and since $\Rightarrow_\beta$ is reflexive,

$$\frac{r \Rightarrow_\beta r \qquad p \Rightarrow_\beta p''}{t = rp \Rightarrow_\beta rp'' = u}$$

2. *Indexed split*: By induction on the definition of $t \overset{n}{\Rightarrow}_\beta s$. We use freely the fact that if $t \overset{n}{\Rightarrow}_\beta s$ then $t \Rightarrow_\beta s$. Cases:
   – *Variable*: $t = x \overset{0}{\Rightarrow}_\beta x = s$. Then $t = x \Rightarrow_{\neg\ell o} x = s$.
   – *Abstraction*: $t = \lambda x.t' \overset{n}{\Rightarrow}_\beta \lambda x.s' = s$ because $t' \overset{n}{\Rightarrow}_\beta s'$. It follows from the *i.h.*.
   – *Application*:

$$\frac{r \overset{n_1}{\Rightarrow}_\beta r' \qquad p \overset{n_2}{\Rightarrow}_\beta p'}{t = rp \overset{n_1+n_2}{\Rightarrow}_\beta r'p' = s}$$

   with $n = n_1 + n_2$. There are only two cases:
   - either $rp \Rightarrow_{\neg\ell o} r'p'$, and then the claim holds;
   - or $rp \not\Rightarrow_{\neg\ell o} r'p'$, then the following conditions hold (otherwise $rp \Rightarrow_{\neg\ell o} r'p'$):
     (a) $r$ is not an abstraction;
     (b) if $r$ is neutral then $p \not\Rightarrow_{\neg\ell o} p'$;
     (c) if $r$ is not neutral then $r \not\Rightarrow_{\neg\ell o} r'$;
     So, if $r$ is neutral, then by *i.h.* applied to $p \overset{n_2}{\Rightarrow}_\beta p'$, $n_2 > 0$ and there is $p''$ such that $p \to_{\ell o} p'' \overset{n_2-1}{\Rightarrow}_\beta r'$; thus, $t = rp \to_{\ell o} rp''$ and

$$\frac{r \overset{n_1}{\Rightarrow}_\beta r' \qquad p'' \overset{n_2-1}{\Rightarrow}_\beta p'}{rp'' \overset{n_1+n_2-1}{\Rightarrow}_\beta r'p' = s}\ .$$

     Otherwise $r$ is not neutral and hence, by *i.h.* applied to $r \overset{n_1}{\Rightarrow}_\beta r'$, $n_1 > 0$ and there exists $r''$ such that $r \to_{\ell o} r'' \overset{n_1-1}{\Rightarrow}_\beta r'$; thus, $t = rp \to_{\ell o} r''p$ and

$$\frac{r'' \overset{n_1-1}{\Rightarrow}_\beta r' \qquad p \overset{n_2}{\Rightarrow}_\beta p'}{r''p \overset{n_1-1+n_2}{\Rightarrow}_\beta r'p' = s}\ .$$

   – *$\beta$ step*:

$$\frac{u \overset{n_1}{\Rightarrow}_\beta u' \qquad r \overset{n_2}{\Rightarrow}_\beta r''}{t = (\lambda x.u)r \overset{n_1+|s|_x\cdot n_2+1}{\Rightarrow}_\beta u'\{x{\leftarrow}r'\} = s}$$

   With $n = n_1 + |s|_x \cdot n_2 + 1 > 0$. We have $t = (\lambda x.u)r \to_{\ell o} u\{x{\leftarrow}r\}$ and by substitutivity of $\overset{n}{\Rightarrow}_\beta$ (Lemma 9) $u\{x{\leftarrow}r\} \overset{n_1+|s|_x\cdot n_2}{\Rightarrow}_\beta u'\{x{\leftarrow}r'\} = s$.
3. *Split*: exactly as in the head case (Prop. 10.3), using the Indexed Split property for $\ell o$ (Point 2 above). □

### A.5 Omitted proofs and lemmas of Sect. 8 (least-level)

*See p. 19* **Proposition 23** (Least level properties)**.**

1. Computational meaning of $\ell\ell$*:* $\ell\ell(t) = \inf\{k \in \mathbb{N} \mid t \to_{\beta:k} u$ *for some term* $u\}$.
2. Monotonicity*: if* $t \to_\beta s$ *then* $\ell\ell(s) \geq \ell\ell(t)$.
3. Invariance by $\to_{\neg\ell\ell}$*: if* $t \to_{\neg\ell\ell} s$ *then* $\ell\ell(s) = \ell\ell(t)$.

*Proof.*

1. By induction on $t$. For any term $r$, we set $\inf_r = \inf\{k \in \mathbb{N} \mid r \to_{\beta:k} u$ for some term $u\}$. Cases:
   - *Variable*, *i.e.* $t$ is a variable. Then, $\ell\ell(t) = \infty$ and $t$ is $\to_\beta$-normal.
   - *Abstraction*, *i.e.* $t = \lambda x.s$. Then, $\ell\ell(t) = \ell\ell(s)$ and, by *i.h.*, $\ell\ell(s) = \inf_s$; now,
   $$\frac{s \to_{\beta:k} u}{t = \lambda x.s \to_{\beta:k} \lambda x.u}$$
   and there is no other rule for $\to_{\beta:n}$ whose conclusion is of the form $\lambda x.s \to_{\beta:n} p$; therefore, $\ell\ell(t) = \ell\ell(s) = \inf_s = \inf_t$.
   - *Application*, *i.e.* $t = t't''$. There are two sub-cases:
     - $t' = \lambda x.s'$, then $\ell\ell(t) = 0$ and
     $$\frac{}{t = (\lambda x.s')t'' \to_{\beta:0} s'\{x \leftarrow t''\}}$$
     thus $\inf_t = 0 = \ell\ell(t)$.
     - $t'$ is not an abstraction, then $\ell\ell(t) = \min\{\ell\ell(t'), \ell\ell(t'') + 1\}$. By *i.h.*, $\ell\ell(t') = \inf_{t'}$ and $\ell\ell(t'') = \inf_{t''}$; now,
     $$\frac{t' \to_{\beta:k} s'}{t = t't'' \to_{\beta:k} s't''} \qquad \text{and} \qquad \frac{t'' \to_{\beta:k} s''}{t = t't'' \to_{\beta:k+1} t's''}$$
     and there is no other rule for $\to_{\beta:n}$ whose conclusion is of the form $t't'' \to_{\beta:n} p$ (as $t'$ is not an abstraction); hence, $\ell\ell(t) = \min\{\inf_{t'}, \inf_{t''} + 1\} = \inf_t$.
2. *Monotonicity*: by induction on the definition of $t \to_\beta s$. Cases:
   - *Abstraction*, *i.e.* $t = \lambda x.t' \to_\beta \lambda x.s' = s$ because $t' \to_\beta s'$. Then, $\ell\ell(t) = \ell\ell(t') \leq \ell\ell(s') = \ell\ell(s)$ by *i.h.*
   - *Application left*, *i.e.* $t = t't'' \to_\beta s't'' = s$ because $t' \to_\beta s'$. By *i.h.*, $\ell\ell(t') \leq \ell\ell(s')$. Hence, $\ell\ell(t) = \min\{\ell\ell(t'), \ell\ell(t'')+1\} \leq \min\{\ell\ell(s'), \ell\ell(t'')+1\} = \ell\ell(s)$.
   - *Application right*, *i.e.* $t = t't'' \to_\beta t's'' = s$ because $t'' \to_\beta s''$. Analogous to the previous case.
   - *$\beta$-redex*, *i.e.* $t = (\lambda x.t')t'' \to_\beta t'\{x \leftarrow t''\} = s$. Then, $\ell\ell(t) = 0 \leq \ell\ell(s)$.
3. *Invariance by* $\to_{\neg\ell\ell}$: By hypothesis, $t \to_{\beta:n} s$ for some $n > \ell\ell(t)$. We proceed by induction on the definition of $t \to_{\beta:n} s$. Cases:
   - *Abstraction*: $t = \lambda x.t' \to_{\beta:n} \lambda x.s' = s$ because $t' \to_{\beta:n} s'$. As $n > \ell\ell(t) = \ell\ell(t')$, then $\ell\ell(s) = \ell\ell(s') = \ell\ell(t') = \ell\ell(t)$ by *i.h.*

- *Application left*, *i.e.* $t = t't'' \to_{\beta:n} s't'' = s$ because $t' \to_{\beta:n} s'$. According to the definition of $\ell\ell(t)$, there are the following sub-cases:
  (a) $\ell\ell(t) = \ell\ell(t') \leq \ell\ell(t'')+1$ and $t'$ is not an abstraction; by *i.h.* applied to $t'$ (since $\ell\ell(t') = \ell\ell(t) < n$), we have $\ell\ell(s') = \ell\ell(t')$, and so $\ell\ell(s) = \min\{\ell\ell(s'), \ell\ell(t'')+1\} = \min\{\ell\ell(t'), \ell\ell(t'')+1\} = \ell\ell(t)$.
  (b) $\ell\ell(t) = \ell\ell(t'')+1 \leq \ell\ell(t')$ and $t'$ is not an abstraction; by Prop. 23.2 (since $\to_{\beta:n} \subseteq \to_\beta$), $\ell\ell(t') \leq \ell\ell(s')$ and therefore $\ell\ell(s) = \min\{\ell\ell(s'), \ell\ell(t'')+1\} = \ell\ell(t'')+1 = \min\{\ell\ell(t'), \ell\ell(t'')+1\} = \ell\ell(t)$.
  (c) $\ell\ell(t) = 0$ and $t' = \lambda x.u'$. According to the definition of $t \to_{\beta:n} s$, since $n > 0$, we have

$$\dfrac{\dfrac{u' \to_{\beta:n} s'}{\lambda x.u' \to_{\beta:n} \lambda x.s'}}{t = (\lambda x.u')t'' \to_{\beta:n} (\lambda x.s')t'' = s} \quad \text{or} \quad \dfrac{t'' \to_{\beta:n-1} s''}{t = (\lambda x.u')t'' \to_{\beta:n} (\lambda x.u')s'' = s}$$

  therefore $\ell\ell(s) = 0 = \ell\ell(t)$.
- *Application right*, *i.e.* $t = t't'' \to_{\beta:n} t's'' = s$ because $t'' \to_{\beta:n-1} s''$. Analogous to the previous case.
- *$\beta$-step*, *i.e.* $t = (\lambda x.t')t'' \to_{\beta:0} t'\{x \leftarrow t''\} = s$ where $0 = n > \ell\ell(t) = 0$, which is impossible. □

**Lemma 27.** *Let $t \to_{\ell\ell} s$ with $\ell\ell(t) > 0$. If $t$ is not an abstraction, then $s$ is not an abstraction.*

*Proof.* By hypothesis, $t = t't''$ and $t'$ is not an abstraction (otherwise $\ell\ell(t) = 0$). Therefore, according to the definition of $\to_{\ell\ell}$, there are only two possibilities:

1. either $t = t't'' \to_{\beta:k} s't'' = s$ because $t' \to_{\beta:k} s'$ and $k = \ell\ell(t)$;
2. or $t = t't'' \to_{\beta:k} t's'' = s$ because $t'' \to_{\beta:k-1} s''$ and $k = \ell\ell(t)$.

In both cases, $s$ is not an abstraction. □

**Lemma 28 (Substitutivity by level).** *If $t \to_{\beta:k} s$ then $t\{x \leftarrow u\} \to_{\beta:k} s\{x \leftarrow u\}$.*

*Proof.* By induction on the definition of $t \to_{\beta:k} s$. Cases:

- *Abstraction*: $t = \lambda y.t' \to_{\beta:k} \lambda y.s' = s$ with $t' \to_{\beta:k} s'$. We can suppose without loss of generality that $y \notin \mathtt{fv}(u) \cup \{x\}$. By *i.h.*, $t'\{x \leftarrow u\} \to_{\beta:k} s'\{x \leftarrow u\}$ and hence $t\{x \leftarrow u\} = \lambda y.t'\{x \leftarrow u\} \to_{\beta:k} \lambda y.s'\{x \leftarrow u\} = s\{x \leftarrow u\}$.
- *Application left*: $t = t't'' \to_{\beta:k} s't'' = s$ with $t' \to_{\beta:k} s'$. By *i.h.*, $t'\{x \leftarrow u\} \to_{\beta:k} s'\{x \leftarrow u\}$ and hence $t\{x \leftarrow u\} = t'\{x \leftarrow u\}t''\{x \leftarrow u\} \to_{\beta:k} s'\{x \leftarrow u\}t''\{x \leftarrow u\} = s\{x \leftarrow u\}$.
- *Application right*: $t = t't'' \to_{\beta:k} t's'' = s$ with $k > 0$ and $t'' \to_{\beta:k} s''$. Analogous to the previous case.
- *$\beta$-redex*: $t = (\lambda y.t')t'' \to_{\beta:0} t'\{y \leftarrow t''\} = s$. We can suppose without loss of generality that $y \notin \mathtt{fv}(u) \cup \{x\}$. Then, $t\{x \leftarrow u\} = (\lambda y.t'\{x \leftarrow u\})t''\{x \leftarrow u\} \to_{\beta:0} t'\{x \leftarrow u\}\{y \leftarrow t''\{x \leftarrow u\}\} = t'\{y \leftarrow t''\}\{x \leftarrow u\} = s\{x \leftarrow u\}$. □

**Proposition 24** ($\ell\ell$ essential properties)**.** *See p. 19*

1. Fullness*: if* $t \to_\beta s$ *then* $t \to_{\ell\ell} u$ *for some* $u$*.*
2. Persistence*: if* $t \to_{\ell\ell} s_1$ *and* $t \to_{\neg\ell\ell} s_2$ *then* $s_2 \to_{\ell\ell} u$ *for some* $u$*.*
3. Diamond*: if* $s \;{}_{\ell\ell}\!\leftarrow \cdot \to_{\ell\ell} u$ *with* $s \neq u$ *then* $s \to_{\ell\ell} \cdot \;{}_{\ell\ell}\!\leftarrow u$*.*

*Proof.*

1. *Fullness*: Since $t$ is not $\to_\beta$-normal, then $\infty \neq \ell\ell(t) = \min\{k \in \mathbb{N} \mid t \to_{\beta:k} r \text{ for some } r\}$ by Prop. 23.1. Therefore, $t \to_{\ell\ell} u$ for some $u$.
2. *Persistence*: Since $t$ is not $\to_\beta$-normal, $\infty \neq \ell\ell(t) = \min\{k \in \mathbb{N} \mid t \to_{\beta:k} r \text{ for some } r\}$ by Prop. 23.1. By least level invariance (Prop. 23.3), $\ell\ell(t) = \ell\ell(s_2)$ and hence, according to Prop. 23.1 again, we have $\infty \neq \ell\ell(s_2) = \min\{k \in \mathbb{N} \mid s_2 \to_{\beta:k} u \text{ for some } u\}$. Therefore, there exists $u$ such that $s_2 \to_{\ell\ell} u$.
3. *Diamond*: The idea of the proof is that, when $(\lambda x.t)s \to_{\ell\ell} t\{x{\leftarrow}s\}$, the $\beta$-redexes in $s$ can be duplicated in $t\{x{\leftarrow}s\}$ but they are not at least level and $\to_{\ell\ell}$ does not reduce outside the least level.
   Formally, according to the definition of $\to_{\ell\ell}$, we have to prove that $s \;{}_{\beta:k}\!\leftarrow t \to_{\beta:k} u$ with $k = \ell\ell(t)$, then $s \to_{\beta:m} r \;{}_{\beta:n}\!\leftarrow u$ for some $r$, where $m = \ell\ell(s)$ and $n = \ell\ell(u)$. To get the right *i.h.*, we prove also that $\ell\ell(s) = k = \ell\ell(u)$ (and so $s \to_{\beta:k} r \;{}_{\beta:k}\!\leftarrow u$).
   Clearly, $t$ is not a variable, otherwise it would be $\to_\beta$-normal.
   If $t = \lambda x.t'$, then $s = \lambda x.s'$ and $u = \lambda x.u'$ with $s' \;{}_{\beta:k}\!\leftarrow t' \to_{\beta:k} u'$ and $s' \neq u'$ and $\ell\ell(t) = \ell\ell(t')$. By *i.h.*, $s' \to_{\beta:m} r \;{}_{\beta:n}\!\leftarrow u'$ for some term $r$, with $\ell\ell(s) = \ell\ell(s') = k = \ell\ell(u') = \ell\ell(u)$, hence $s = \lambda x.s' \to_{\beta:k} \lambda x.r \;{}_{\beta:k}\!\leftarrow \lambda x.u' = u$.
   Finally, consider $t = t_0t_1$. The case where $t_0 = \lambda x.t_2$ and $s = t_2\{x{\leftarrow}t_1\} \;{}_{\beta:k}\!\leftarrow t \to_{\beta:k} (\lambda x.t_2)u_1 = u$ with $t_1 \to_{\beta:k-1} u_1$ is impossible, because $k = \ell\ell(t) = 0$. The remaining cases for $t = t_0t_1$ are:
   - $s = t_0s_1 \;{}_{\beta:k}\!\leftarrow t = t_0t_1 \to_{\beta:k} u_0t_1 = u$ with $t_0 \to_{\beta:k} u_0$ and $t_1 \to_{\beta:k} s_1$ and $\ell\ell(t_0) = \ell\ell(t) = \ell\ell(t_1) + 1 = k > 0$. Then, $t_0$ is not an abstraction (otherwise $\ell\ell(t) = 0$) and, by Lemma 27, $u_0$ is not an abstraction. By Prop. 23.2, $\ell\ell(s_1) \geq \ell\ell(t_1)$ and $\ell\ell(u_0) \geq \ell\ell(t_0)$. Hence, $\ell\ell(s) = \min\{\ell\ell(t_0), \ell\ell(s_1) + 1\} = \ell\ell(t_0) = \ell\ell(t_1) + 1 = \min\{\ell\ell(u_0), \ell\ell(t_1) + 1\} = \ell\ell(u)$ and so $s \to_{\beta:k} u_0s_1 \;{}_{\beta:k}\!\leftarrow u$.
   - $s = s_0t_1 \;{}_{\beta:k}\!\leftarrow t = t_0t_1 \to_{\beta:k} u_0t_1 = u$ with $s_0 \;{}_{\beta:k}\!\leftarrow t_0 \to_{\beta:k} u_0$ and $s_0 \neq u_0$ and $k = \ell\ell(t) = \ell\ell(t_0) \leq \ell\ell(t_1) + 1$. By *i.h.*, there is $r_0$ such that $s_0 \to_{\beta:k} r_0 \;{}_{\beta:k}\!\leftarrow u_0$ where $\ell\ell(s_0) = k = \ell\ell(u_0)$. Thus, $s \to_{\beta:k} r_0t_1 \;{}_{\beta:k}\!\leftarrow u$. If $s_0$ or $u_0$ is an abstraction, then $\ell\ell(s) = 0$ or $\ell\ell(u) = 0$ and hence (by Prop. 23.2) $0 = \ell\ell(s) \geq \ell\ell(t)$ or $0 = \ell\ell(s) \geq \ell\ell(t)$, so in both cases $k = \ell\ell(t) = 0 = \ell\ell(s) = \ell\ell(u)$. Otherwise, $\ell\ell(s) = \min\{\ell\ell(s_0), \ell\ell(t_1) + 1\} = k = \min\{\ell\ell(u_0), \ell\ell(t_1) + 1\} = \ell\ell(u)$.
   - $s = t_0s_1 \;{}_{\beta:k}\!\leftarrow t = t_0t_1 \to_{\beta:k} t_0u_0 = u$ with $s_1 \;{}_{\beta:k-1}\!\leftarrow t_1 \to_{\beta:k-1} u_1$ and $k = \ell\ell(t) = \ell\ell(t_1) + 1$. Analogous to the previous case.
   - $s = t_2\{x{\leftarrow}t_1\} \;{}_{\beta:k}\!\leftarrow t = (\lambda x.t_2)t_1 \to_{\beta:k} (\lambda x.s_2)t_1 = u$ where $t_0 = \lambda x.t_2$ and $t_2 \to_{\beta:k} s_2$ and $k = \ell\ell(t_2) = \ell\ell(t) = 0 = \ell\ell(u)$. Thus, $t_2 \to_{\beta:0} s_2$ and $t \to_{\beta:0} t_2\{x{\leftarrow}t_1\}$. By substitutivity by level (Lemma 28), $t_2\{x{\leftarrow}t_1\} \to_{\beta:0} s_2\{x{\leftarrow}t_1\}$ and so $\ell\ell(t_2\{x{\leftarrow}t_1\}) = 0$ by Prop. 23.1. Therefore $t_2\{x{\leftarrow}t_1\} \to_{\beta:0} s_2\{x{\leftarrow}t_1\} \;{}_{\beta:0}\!\leftarrow u$. □

**Lemma 29 (Merge by level).** *If $t \Rightarrow_{\beta:n} s \to_{\beta:m} u$ with $n > m$, then $t \Rightarrow_\beta u$.*

*Proof.* By induction on the definition of $t \Rightarrow_{\beta:n} s$. Cases:

- *Variable*: $t = x \Rightarrow_{\beta:\infty} x = s$. Then, there is no $u$ such that $s \to_{\beta:m} u$ for any $m \in \mathbb{N}$.
- *Abstraction*: $t = \lambda x.t' \Rightarrow_{\beta:n} \lambda x.s' = s$ because $t' \Rightarrow_{\beta:n} s'$. According to the definition of $s \to_{\beta:m} u$, by necessity $u = \lambda x.u'$ with $s' \to_{\beta:m} u'$. By *i.h.*, $t' \Rightarrow u'$, thus

$$\frac{t' \Rightarrow u'}{t = \lambda x.t' \Rightarrow \lambda x.u' = u}.$$

- *Application*:

$$\frac{t_0 \Rightarrow_{\beta:n_0} s_0 \qquad t_1 \Rightarrow_{\beta:n_1} s_1}{t = t_0 t_1 \Rightarrow_{\beta:n} s_0 s_1 = s}$$

  where $n = \min\{n_0, n_1 + 1\}$. According to the definition of $s \to_{\beta:m} u$, there are the following sub-cases:

  1. $s = s_0 s_1 \to_{\beta:m} u_0 s_1 = u$ with $s_0 \to_{\beta:m} u_0$; since $m < n \leq n_0$, by *i.h.* applied to $t_0 \Rightarrow_{\beta:n_0} s_0 \to_{\beta:m} u_0$, we have $t_0 \Rightarrow_\beta u_0$, and so (as $\Rightarrow_{\beta:n_1} \subseteq \Rightarrow_\beta$)

$$\frac{t_0 \Rightarrow_\beta u_0 \qquad t_1 \Rightarrow_\beta s_1}{t = t_0 t_1 \Rightarrow_\beta u_0 s_1 = u};$$

  2. $s = s_0 s_1 \to_{\beta:m} u_0 s_1 = u$ with $s_1 \to_{\beta:m-1} u_1$; since $m - 1 < n - 1 \leq n_1$, by *i.h.* applied to $t_1 \Rightarrow_{\beta:n_1} s_1 \to_{\beta:m-1} u_1$, we have $t_1 \Rightarrow u_1$, and so (as $\Rightarrow_{\beta:n_0} \subseteq \Rightarrow$)

$$\frac{t_0 \Rightarrow s_0 \qquad t_1 \Rightarrow u_1}{t = t_0 t_1 \Rightarrow s_0 u_1 = u};$$

  3. $s = (\lambda x.s'_0) s_1 \to_{\beta:0} s_0\{x \leftarrow s_1\} = u$ with $s_0 = \lambda x.s'_0$ and $m = 0$; as $n > 0$ then, according to the definition of $t \Rightarrow_{\beta:n} s$,

$$\frac{\dfrac{t_0 \Rightarrow_{\beta:n_0} s_0}{\lambda x.t_0 \Rightarrow_{\beta:n_0} \lambda x.s_0} \qquad t_1 \Rightarrow_{\beta:n_1} s_1}{t = (\lambda x.t_0) t_1 \Rightarrow_{\beta:n} (\lambda x.s_0) s_1 = s}$$

  where $n = \min\{n_0, n_1 + 1\}$; therefore (as $\Rightarrow_{\beta:k} \subseteq \Rightarrow$)

$$\frac{t_0 \Rightarrow s_0 \qquad t_1 \Rightarrow s_1}{t = (\lambda x.t_0) t_1 \Rightarrow s_0\{x \leftarrow s_1\} = u}.$$

□

**Proposition 25** ($\ell\ell$ macro-step system)**.** *See p. 19*

1. Merge*: if $t \Rightarrow_{\neg\ell\ell} s \to_{\ell\ell} u$, then $t \Rightarrow_\beta u$.*
2. Indexed split*: if $t \overset{n}{\Rightarrow}_\beta s$ then $t \Rightarrow_{\neg\ell\ell} s$, or $n > 0$ and $t \to_{\ell\ell} \cdot \overset{n-1}{\Rightarrow}_\beta s$.*

*3.* Split*: if* $t \Rightarrow_\beta s$ *then* $t \to^*_{\ell\ell} \cdot \Rightarrow_{\neg\ell\ell} s$*.*

*That is,* $(\Lambda, \{\to_{\ell\ell}, \to_{\neg\ell\ell}\})$ *is a macro-step system with respect to* $\Rightarrow_\beta$ *and* $\Rightarrow_{\neg\ell\ell}$*.*

*Proof.*

1. *Merge*: As $t \Rightarrow_{\neg\ell\ell} s \to_{\ell\ell} u$, then $t \Rightarrow_{\beta:n} s \to_{\beta:m} u$ for some $n \in \mathbb{N} \cup \{\infty\}$ and $m \in \mathbb{N}$ such that $n > \ell\ell(t)$ and $m = \ell\ell(s)$. Since $\Rightarrow_{\neg\ell\ell} \subseteq \to^*_{\neg\ell\ell}$ and $\to_{\neg\ell\ell}$ cannot change the least level (Prop. 23.3), $\ell\ell(t) = \ell\ell(s)$ and so $n > m$. By Merge by Level (Lemma 29), $t \Rightarrow_\beta u$.
2. *Indexed split*: By induction on the definition of $t \overset{n}{\Rightarrow}_\beta s$. We freely use the fact that if $t \overset{n}{\Rightarrow}_\beta s$ then $t \Rightarrow_\beta s$. Cases:
   - *Variable*: $t = x \overset{0}{\Rightarrow}_\beta x = s$. Then, $t = x \Rightarrow_{\neg\ell\ell} x = s$ since $x \Rightarrow_{\beta:\infty} x$.
   - *Abstraction*: $t = \lambda x.t' \overset{n}{\Rightarrow}_\beta \lambda x.s' = s$ because $t' \overset{n}{\Rightarrow}_\beta s'$. It follows by the *i.h.*.
   - *Application*:

$$\frac{r \overset{n_1}{\Rightarrow}_\beta r' \qquad p \overset{n_2}{\Rightarrow}_\beta p'}{t = rp \overset{n_1+n_2}{\Rightarrow}_\beta r'p' = s} \tag{2}$$

     with $n = n_1 + n_2$. There are only two cases:
     - either $rp \Rightarrow_{\neg\ell\ell} r'p'$, and then the claim holds;
     - or $rp \not\Rightarrow_{\neg\ell\ell} r'p'$ and hence any derivation with conclusion $rp \Rightarrow_{\beta:d} r'p'$ is such that $d = \ell\ell(rp) \in \mathbb{N}$. Let us rewrite the derivation (2) replacing $\overset{n}{\Rightarrow}_\beta$ with $\Rightarrow_{\beta:k}$: we have[11]

$$\frac{r \Rightarrow_{\beta:d_r} r' \qquad p \Rightarrow_{\beta:d_p} p'}{t = rp \Rightarrow_{\beta:d} r'p' = s}$$

       where $d = \min\{d_r, d_p + 1\}$. Thus, there are two sub-cases:
       (a) $d = d_r \leq d_p + 1$ and then $d = \ell\ell(rp) \leq \ell\ell(r) \leq d_r = d$ (the first inequality holds by definition of $\ell\ell(rp)$), hence $\ell\ell(r) = d_r$; we apply the *i.h.* to $r \overset{n_1}{\Rightarrow}_\beta r'$ and we have that $r \Rightarrow_{\neg\ell\ell} r'$, or $n_1 > 0$ and $r \to_{\ell\ell} u_1 \overset{n_1-1}{\Rightarrow}_\beta r'$; but $r \Rightarrow_{\neg\ell\ell} r'$ is impossible because otherwise $rp \Rightarrow_{\neg\ell\ell} r'p'$ (as $d_r \leq d_p + 1$of s); therefore, $n_1 > 0$ and $r \to_{\ell\ell} u_1 \overset{n_1-1}{\Rightarrow}_\beta r'$, and so $n > 0$ and $t = rp \to_{\ell\ell} u_1 p \overset{n_1-1+n_2}{\Rightarrow}_\beta r'p' = s$.
       (b) $d = d_p + 1 \leq d_r$ and then $d = \ell\ell(rp) \leq \ell\ell(p) + 1 \leq d_p + 1 = d$, hence $\ell\ell(p) = d_p$; we conclude analogously to thee previous sub-case.
   - $\beta$ *step*:

$$\frac{u \overset{n_1}{\Rightarrow}_\beta u' \qquad r \overset{n_2}{\Rightarrow}_\beta r'}{t = (\lambda x.u)r \overset{n_1+|u'|_x \cdot n_2+1}{\Rightarrow}_\beta u'\{x \leftarrow r'\} = s}$$

[11] This is possible because the inference rules for $\overset{n}{\Rightarrow}_\beta$ and $\Rightarrow_{\beta:k}$ are the same except for the way they manage their own indexes $n$ and $k$.

With $n = n_1 + |u'|_x \cdot n_2 + 1 > 0$. We have $t = (\lambda x.u)r \to_{\ell\ell} u\{x \leftarrow r\}$ and by substitutivity of $\overset{n}{\Rightarrow}_\beta$ (Lemma 9) $u\{x \leftarrow r\} \overset{n_1+|u'|_x \cdot n_2}{\Rightarrow}_\beta u'\{x \leftarrow r'\} = s$.

3. *Split*: Exactly as in the head case (Prop. 10.3), using the Indexed Split property for $\ell\ell$ (Point 2 above). □