Arch. Math. Logic (2007) 46:385–424
DOI 10.1007/s00153-007-0039-1

Mathematical Logic

Analytic proof systems for λ-calculus: the elimination of transitivity, and why it matters

Pierluigi Minari

Received: 22 September 2005 / Revised: 27 November 2006 / Published online: 27 February 2007

Abstract We introduce new proof systems $\mathbf{G}[\beta]$ and $\mathbf{G}_{\text{ext}}[\beta]$, which are equivalent to the standard equational calculi of $\lambda\beta$- and $\lambda\beta\eta$-conversion, and which may be qualified as 'analytic' because it is possible to establish, by purely proof-theoretical methods, that in both of them the transitivity rule admits effective elimination. This key feature, besides its intrinsic conceptual significance, turns out to provide a common logical background to new and comparatively simple demonstrations—rooted in nice proof-theoretical properties of transitivity-free derivations—of a number of well-known and central results concerning β- and $\beta\eta$-reduction. The latter include the *Church–Rosser* theorem for both reductions, the *Standardization* theorem for β-reduction, as well as the *Normalization* (*Leftmost reduction*) theorem and the *Postponement of η-reduction* theorem for $\beta\eta$-reduction.

Keywords Lambda-calculus · Extensionality · Elimination of transitivity · Equational proof systems · Lambda reduction

Mathematics Subject Classification (2000) 03B40 · 03F03 · 03F05 · 03F07

1 Introduction

A "standard" *equational proof system* $\mathbf{E}$ over an equational language $\mathcal{L}_\mathbf{E}$ consists of: (i) certain *specific axioms*, namely a set $A_\mathbf{E}$ of $\mathcal{L}_\mathbf{E}$-equations, closed under substitutions; (ii) the usual deductive apparatus of equational logic, which comprises the axiom schema of *reflexivity* of equality and the inference

P. Minari (✉)
Department of Philosophy, University of Florence, Via Bolognese 52, 50139 Firenze, Italy
e-mail: minari@unifi.it

rules of *symmetry*, *transitivity* and f-*congruence* of equality w.r. to each term constructor f of $\mathcal{L}_{\mathbf{E}}$—the substitution rule being superfluous in view of the closure requirement on $A_{\mathbf{E}}$. By Birkhoff's *Completeness theorem*, it holds that $\vdash_{\mathbf{E}} t = s \;\Leftrightarrow\; A_{\mathbf{E}} \models t = s$.

Such syntactic presentations of equational theories are, without saying, very convenient under most respects. Proof-theoretical investigations, however, are virtually made impossible because of the presence of the *transitivity* rule which, as a matter of fact, cannot be dispensed with (except for trivial cases). Suppose, for instance, that we are interested in finding a *syntactic* consistency proof of a given **E** for which we do not have a natural model ready at hand. We cannot, arguing by induction on the length of **E**-derivations, conclude that $\nvdash_{\mathbf{E}} x = y$ (x, y two distinct variables) by showing that $x = y$ neither is a specific axiom (assume it is not) nor it can be obtained as a conclusion of one of the inference rules: this simple argument trivially breaks down when the case of the transitivity rule is considered—and in fact in this case only.

Indeed, the transitivity rule shares with the rule of *modus ponens* in propositional and first-order Hilbert-style calculi an intrinsically *synthetic* character, which is responsible for the potential loss of relevant information (a term/a formula) along formal derivations and, ultimately, for the lack of any significant mathematical structure on the part of the latter.

Given that standard, 'synthetic' equational calculi do not lend themselves directly to proof-theoretical analysis, one might ask the question whether there are significant cases in which it is both *possible* and *useful* to turn a 'synthetic' equational proof system into an equivalent 'analytic' one, where the transitivity rule can be actually demonstrated being *redundant*. In [26] we addressed this question relatively to a specific class of equational theories including combinatory logic **CL** and more general systems $\mathbf{C}[\mathcal{X}]$ of combinators. The new formal systems $\mathbf{G}[\mathcal{C}]$, respectively $\mathbf{G}[\mathcal{X}]$ introduced to this aim were shown to be equivalent to their synthetic counterparts and to have the key feature—which qualifies them as *analytic*—that every derivation can effectively be transformed into a *transitivity-free* derivation of the same equation. As a first consequence, the *syntactic consistency* of $\mathbf{G}[\mathcal{C}]$ (in general of $\mathbf{G}[\mathcal{X}]$), and so of **CL** ($\mathbf{C}[\mathcal{X}]$) too, could be immediately established by the simple inductive argument alluded to above. But also, due to the nice proof-theoretical behavior of transitivity-free derivations, we were able to give new demonstrations of well known results concerning **CL** and *weak* reduction (e.g. the *Church–Rosser* theorem and the *leftmost reduction* theorem) and to generalize these results to arbitrary combinatory systems $\mathbf{C}[\mathcal{X}]$ as well.

In the present paper, our aim is to show how the "analytic" approach can be fruitfully extended to λ-*calculus*, namely to the equational theories $\boldsymbol{\lambda\beta}$ and $\boldsymbol{\lambda\beta\eta}$. This proves to be—perhaps not surprisingly—an intriguingly harder task, as we will see (a previous, only partially successful attempt of ours in this direction is sketched in [27]). In Sect. 2 we introduce the analytic proof systems $\mathbf{G}[\beta]$ and $\mathbf{G}_{\text{ext}}[\beta]$. Their distinguishing feature consists in the pair of symmetrical (*left* and *right*) β-rules

$$\frac{t[x/r]p_1 \cdots p_n = s}{(\lambda x.t)rp_1 \cdots p_n = s}\,[\beta_l] \quad \frac{t = s[x/r]p_1 \cdots p_n}{t = (\lambda x.s)rp_1 \cdots p_n}\,[\beta_r] \quad (n \geq 0)$$

which replace the β-*conversion* schema

$$(\lambda x.t)s = t[x/s]$$

characteristic of the corresponding synthetic proof systems. As far as the equational deductive apparatus of the latter is concerned, only the *reflexivity* schema $[\varrho]$ (restricted to variables) and the rules $[\xi]$ of *weak extensionality* (or *congruence w.r.t. to abstraction*) and $[\tau]$ of *transitivity* are retained as they are also in the analytic calculi; instead, the *symmetry* rule $[\sigma]$ is dropped, while the usual pair of *app-congruence* rules $[\mu]$ and $[\nu]$ is replaced by a single *parallel application* rule. Finally, to obtain $\mathbf{G}_{\mathrm{ext}}[\beta]$, the *extensionality* rule $[Ext]$ (or $[\zeta]$, as it is sometimes called in the literature, see e.g. [18]) is added. The equivalence between $\mathbf{G}[\beta]$ and $\boldsymbol{\lambda\beta}$, as well as between $\mathbf{G}_{\mathrm{ext}}[\beta]$ and $\boldsymbol{\lambda\beta\eta}$, is easily established.

Sections 3 and 5 are devoted to the demonstration (by purely proof-theoretical methods) of the central result of the paper, namely the *effective eliminability of the transitivity rule* in $\mathbf{G}[\beta]$, respectively $\mathbf{G}_{\mathrm{ext}}[\beta]$ (Theorems 3.11 and 5.8). Compared to the corresponding τ-eliminability problem for analytic combinatory systems $\mathbf{G}[\mathcal{X}]$ tackled in [26], this task turns out to be considerably more difficult, as we try to explain in Sect. 3, essentially due to the presence of the weak extensionality rule $[\xi]$ (and, in $\mathbf{G}_{\mathrm{ext}}[\beta]$, of $[Ext]$ as well). As a consequence, the proof strategy is going to require a considerable number of steps and technical tools, including one reminiscent of the *indexing* (or *marking*) technique familiar from the literature on lambda reduction (see e.g. [3], Chap. 11).

Just the very fact that analytic, transitivity-free equational proof systems adequate for λ-calculus can now be exhibited, has in our opinion a certain *conceptual* relevance. These systems, in particular, are easily seen to enjoy a kind of *subterm property* (Lemmas 2.3 and 2.4)—analogous, in a sense, to the *subformula-property* of cut-free sequent calculi—which immediately yields, *without any need to analyze convertibility* (*equality*) *through a reduction calculus*, the underivability of equations of the form $x = y$ with x and y distinct variables, hence also the *consistency* of the analytic proof systems $\mathbf{G}[\beta]$ and $\mathbf{G}_{\mathrm{ext}}[\beta]$ and, in turn, of their synthetic counterparts $\boldsymbol{\lambda\beta}$ and $\boldsymbol{\lambda\beta\eta}$.

More specific applications, presented in the remaining Sects. 4 and 6, are intended to try and motivate further *why analyticity and the elimination of transitivity matter*. The point is that τ-*free* derivations in $\mathbf{G}[\beta]$ and $\mathbf{G}_{\mathrm{ext}}[\beta]$ do have a non trivial mathematical structure and a number of nice properties, which may be analyzed and exploited by proof-theoretical methods. As a consequence, we find at our disposal a sort of unified logical background in which new and comparatively short proofs of well-known central results concerning β- and

$\beta\eta$-reduction can be developed just by reflecting on *τ-free provable β- and $\beta\eta$-equality*. This is the case, in particular, of the *Church–Rosser* property for both reductions (Theorems 4.3 and 6.3), of the (weak) *Standardization* theorem for β-reduction (Theorem 4.7), and finally of the *Normalization* (or *Leftmost reduction*) theorem and the *Postponement of η-reduction* theorem (Corollaries 6.8, 6.9) for $\beta\eta$-reduction.

Related work In the last two decades, the issue of *transitivity elimination* has been extensively investigated, mainly by computer scientists, in connection with *typed* λ-calculi, type theories and programming languages. It is, however, transitivity elimination *for the subtyping relation*. Subtyping enriches a type system by means of a preorder relation $\leq$ over types (the intended meaning of $S \leq T$ being that type S is a subtype of type T, i.e. that all expressions of type S can be used in any context where an expression of type T is expected) governed by a set of appropriate subtyping rules (a *subtyping system*). Usually combined with other type-theoretic constructs, subtyping gives rise to flexible and powerful formal environments, which are fruitfully applied to model notions such as e.g. object inheritance in object-oriented programming, or theory and proof reuse in proof assistants (see the informative survey [9]).

The explicit presence of the transitivity rule $\tau_{\leq}$ for the subtyping relation in a given subtyping system Σ, especially when combined with type-dependency, is responsible for serious difficulties in the verification of important meta-theoretical properties, such as subtyping-checking, subject reduction and decidability; on the other hand, the presence of certain other rules in Σ makes it impossible to eliminate $\tau_{\leq}$ outright. Hence the attempts to devise *equivalent* reformulations of Σ (e.g. by a suitable modification of the conflicting rules) which provably admit elimination of the transitivity rule (TE). A subtyping system for second-order λ-calculus **F** enjoying TE has been devised by Curien and Ghelli [12] by a clever reformulation of the rule for using primitive subtyping; in the related work of Longo, Milsted and Soloviev [23] subtyping for **F** is presented as logical entailment by means of a sequent calculus $Co^{\vdash}$ enjoying cut elimination, the latter implying TE. The problem of transitivity elimination in subtyping extensions of systems with (first or second order) *dependent types*—i.e., systems lying on the left and the rear face of Barendregt's *lambda-cube*, cf. [4]—is considerably more difficult. Indeed, once an equivalence on types (typically: β-conversion) coexists with the subtyping relation between types, further conflicts between transitivity elimination and other rules, notably the so-called 'conversion subtyping rule' (stating that any two convertible types are in the subtyping relation), do arise. Just to mention a few important results, around 1995 a *weak* form of TE (i.e., TE only at the level of *normalized* types) has been proved by Compagnoni (cf. [10]) for $\mathbf{F}^{\omega}_{\wedge}$, an extension of Girard's system $\mathbf{F}^{\omega}$ with subtyping, bounded quantification and intersection types, and by Aspinall and Compagnoni (cf. [2]) for $\lambda\mathbf{P}_{\leq}$, a subtyping extension of first-order dependent types system $\lambda\mathbf{P}$. More recently, a considerable improvement has been achieved with the work of Chen, proving *full* TE (for arbitrary, hence possibly non-normalized types) for the system $\lambda\boldsymbol{\Pi}_{\leq}$, equivalent to $\lambda\mathbf{P}_{\leq}$ ([6,7]), as well as for other type theories with subtyping among which $\lambda\mathbf{C}_{\leq}$, a subtyping extension of the strongest system in the lambda-cube, i.e. Coquand's Calculus of Constructions **CC** ([7]; see also [8], proving TE for an extension of $\lambda\mathbf{C}_{\leq}$ with coercions).

It is perhaps worth noticing here that one of the key steps in Chen's proof of TE for $\lambda\boldsymbol{\Pi}_{\leq}$ consists in the inclusion in the subtyping system of the two symmetrical rules

$$\frac{\Gamma \vdash B[x/M_1]M_2 \cdots M_n \leq C}{\Gamma \vdash (\Lambda x : A.B)M_1 \cdots M_n \leq C}\,\text{S-ApSL}, \qquad \frac{\Gamma \vdash C \leq B[x/M_1]M_2 \cdots M_n}{\Gamma \vdash C \leq (\Lambda x : A.B)M_1 \cdots M_n}\,\text{S-ApSR}$$

dealing with the subtyping derived from β-conversion at the type level ($A, B, \ldots$ vary over pre-types and $M, M_1 \ldots$ over pre-terms; Γ is a pre-context). These rules closely resemble our above mentioned β-rules.[1] But it has to be stressed that, working as we do with *untyped* λ-calculus, *we cannot rely on normalization*, whereas (strong) normalization is enjoyed by the above mentioned typed systems, a fact that also plays a crucial role in the proofs of TE (indeed, Chen's induction measure in the proof of admissibility of transitivity for $\lambda\boldsymbol{\Pi}_{\leq}$ is based on the maximum number of normalizing reduction steps of the involved type-expressions).

2 Synthetic versus analytic proof systems for λ-calculus

2.1 Preliminaries

In the following, Λ denotes the set of all pure λ-terms built over the countably infinite set $V = \{v_0, v_1, \ldots\}$ of variables:

$$\Lambda = V \mid (\lambda V.\Lambda) \mid (\Lambda\Lambda)\,.$$

The letters $x, y, z, \ldots$ and $t, s, r, \ldots$ vary over V, respectively Λ. The symbol $\equiv$ denotes syntactic identity between λ-terms. In writing λ-terms and in dealing with the syntactic operation $t, s, x \rightarrowtail t[x/s]$ of *substitution* we adopt throughout the standard conventions (see e.g. [3], Chap. 2, Sect. 1); in particular:

- outermost parentheses are not written, and missing ones (e.g. as in *tsr*) are to be restored on the left;
- *α-congruent* terms are *syntactically identified*;
- all bound variables occurring in the terms $t_1, t_2, \ldots$ taken into consideration in a certain context (e.g. a statement, a proof, etc.) are assumed to be different from the free variables occurring in them.

[1] Indeed, the idea of doing weak head expansions on the left or on the right side is not new; e.g. it can be found, although not so explicitly, also in [11], presenting an algorithm for testing $\beta\eta$-conversion in type theory.

Further notational conventions include:

- $\mathrm{FV}(t, s, \ldots) :=$ the set of all variables occurring free in the terms $t, s, \ldots$;
- $\|t\| :=$ the *depth* of t, inductively defined as follows:
  (a) $\|x\| = 0 \ (x \in V)$,
  (b) $\|sr\| = \max(\|s\|, \|r\|) + 1$,
  (c) $\|\lambda x.s\| = \|s\| + 1$.

An *application* (*abstraction*) term is a term of the form ts ($\lambda x.t$). A β- (η-) redex is an application (abstraction) term of the form $(\lambda x.t)s \ \ (\lambda x.tx,\ x \notin \mathrm{FV}(t))$.

2.2 Synthetic calculi

In the standard presentation, the *synthetic* equational calculus $\boldsymbol{\lambda\beta}$, or $\boldsymbol{\lambda K}$-calculus, consists of the following axiom-schema (*β-conversion*) and inference rules (*reflexivity*, which by convenience we treat as a 0-premises inference rule, *symmetry*, *transitivity*, *right* and *left app-congruence*, and *weak extensionality*):

$$(\lambda x.t)s = t[x/s]\,, \quad \frac{}{t = t}\,[\rho]\,, \quad \frac{t = s}{s = t}\,[\sigma]\,, \quad \frac{t = s \quad s = r}{t = r}\,[\tau]\,,$$
$$\frac{t = s}{rt = rs}\,[\mu]\,, \quad \frac{t = s}{tr = sr}\,[\nu]\,, \quad \frac{t = s}{\lambda x.t = \lambda x.s}\,[\xi]\,.$$

The calculus $\boldsymbol{\lambda\beta\eta}$, or *extensional* $\boldsymbol{\lambda K}$-calculus, results from the addition to $\boldsymbol{\lambda\beta}$ of the *η-conversion* schema

$$\lambda x.tx = t\,, \quad \text{provided } x \notin \mathrm{FV}(t)\,,$$

or, equivalently (see e.g. [3]), of the *extensionality* rule

$$\frac{tx = sx}{t = s}\,[Ext]\,, \quad \text{where } x \notin \mathrm{FV}(t, s)\,.$$

2.3 Analytic calculi

The *analytic* equational calculus $\mathbf{G}[\beta]$ is determined by the following inference rules:

(i) *Left* and *right* introduction rules (*β-rules*):

$$\frac{t[x/p_0]p_1 \cdots p_n = s}{(\lambda x.t)p_0 \cdots p_n = s}\,[\beta_l]\,, \quad \frac{t = s[x/p_0]p_1 \cdots p_n}{t = (\lambda x.s)p_0 \cdots p_n}\,[\beta_r] \ \ (n \geq 0)\,,$$

where the (possibly missing) terms $p_1, \ldots p_n$ are said to be the *side terms* of the inference;

(ii) *Structural* rules:
$[\rho]$ *restricted to variables*, $[\tau]$, $[\xi]$ and, instead of $[\mu]$ and $[\nu]$, the rule

$$\frac{t_1 = s_1 \quad t_2 = s_2}{t_1 t_2 = s_1 s_2}\ [App]$$

of *parallel* application.

The *analytic* equational calculus $\mathbf{G}_{\text{ext}}[\beta]$ is obtained from $\mathbf{G}[\beta]$ by adding the (non structural) extensionality rule $[Ext]$.

Note that the reflexivity (0-premises) rule is *restricted* to variables. It is easily verified that all equations of the form $t = t$ are derivable by means of *structural* rules alone. Also, the *symmetry* rule is missing; however, it is readily seen that

Fact 2.1 $\mathbf{G}[\beta]$ *and* $\mathbf{G}_{\text{ext}}[\beta]$ *are closed under* $[\sigma]$.

In fact, due to the left/right symmetry of the β-rules, to each derivation $\mathfrak{D}$ in $\mathbf{G}_{(\mathbf{ext})}[\beta]$ we may associate the *dual* derivation $\widetilde{\mathfrak{D}}$ by effecting the following transformations: (i) each node $t = s$ is replaced by $s = t$; (ii) labels $[\beta_r]$ are changed into $[\beta_l]$ and conversely; (iii) the premises of a τ-inference, if any, are interchanged. We clearly have

$$\mathfrak{D} \vdash t = s \quad \text{iff} \quad \widetilde{\mathfrak{D}} \vdash s = t\,.$$

Now, it is easily verifiable that the analytic and synthetic calculi are *equivalent*.

Proposition 2.2 *For every equation* $t = s$:

$$\boldsymbol{\lambda\beta} \vdash t = s \;\Leftrightarrow\; \mathbf{G}[\beta] \vdash t = s\,; \quad \boldsymbol{\lambda\beta\eta} \vdash t = s \;\Leftrightarrow\; \mathbf{G}_{\text{ext}}[\beta] \vdash t = s\,.$$

2.4 Consistency of τ-free analytic calculi

Let $\mathbf{G}^-[\beta]$ ($\mathbf{G}^-_{\text{ext}}[\beta]$) be $\mathbf{G}[\beta]$ ($\mathbf{G}_{\text{ext}}[\beta]$) *minus the transitivity rule* $[\tau]$. In view of the peculiar form of the β-rules, $\mathbf{G}^-[\beta]$-derivations enjoy the following simple yet useful *weak subterm property*.

Lemma 2.3 *Let* $\mathfrak{D}$ *be a* τ*-free derivation of* $t = s$ *in* $\mathbf{G}[\beta]$.

(i) *If* $\mathfrak{D}$ *contains* at least one *occurrence of a* left (right) β*-rule, then the term* t (*respectively*: s) *contains* at least one *occurrence of a* β-redex.
(ii) *If both t and s are* β*-normal forms, then* $t \equiv s$.

Proof (i): straightforward, by induction on the length (number of nodes) of $\mathfrak{D}$. (ii): under the assumptions, it follows by (i) that the only inference rules possibly occurring in $\mathfrak{D}$ are $[\varrho]$, $[\xi]$ and $[App]$, and clearly the end–equation of every such derivation must be of the form $r = r$. □

As far as $\mathbf{G}^-_{\text{ext}}[\beta]$ is concerned, property (i) does no longer hold (because β-redex occurrences may be destroyed by the extensionality rule). Yet, we can easily prove the weaker

Lemma 2.4 *Let $\mathfrak{D}$ be a τ-free derivation of $t = s$ in $\mathbf{G}_{\text{ext}}[\beta]$.*

(i) *If $\mathfrak{D}$ contains* at least one *occurrence of a* left (right) *β-rule, then the term t (resp.: s) contains* at least one *occurrence of an* abstraction *term.*
(ii) *If t and s do not contain occurrences of λ, then $t \equiv s$.*

As an immediate consequence of (ii) of Lemmas 2.3 and 2.4, no equation $x = y$ with x distinct from y can be proved in $\mathbf{G}^-[\beta]$ or $\mathbf{G}^-_{\text{ext}}[\beta]$; that is:

Proposition 2.5 *The τ-free analytic calculi $\mathbf{G}^-[\beta]$ and $\mathbf{G}^-_{\text{ext}}[\beta]$ are consistent.*

So, the *transitivity elimination* theorems we are going to prove in Sect. 3 (Theorem 3.11) and Sect. 5 (Theorem 5.8), entailing the equivalence between $\mathbf{G}[\beta]$ and $\mathbf{G}^-[\beta]$, resp. $\mathbf{G}_{\text{ext}}[\beta]$ and $\mathbf{G}^-_{\text{ext}}[\beta]$, together with Propositions 2.2 and 2.5 will provide a *purely proof-theoretical* demonstration of:

Proposition 2.6 *The synthetic equational proof systems* $\boldsymbol{\lambda\beta}$ *and* $\boldsymbol{\lambda\beta\eta}$ *are consistent.*

Remark 2.7 Let $\mathbf{G}[\beta\eta]$ be the calculus obtained from $\mathbf{G}[\beta]$ by adding the following (*left* and *right*) η-introduction rules:

$$\frac{t = s}{\lambda x.tx = s}\,[\eta_l] \qquad \frac{s = t}{s = \lambda x.tx}\,[\eta_r] \qquad (x \notin \mathrm{FV}(t))\,.$$

It is easy to verify that $\mathbf{G}_{\text{ext}}[\beta]$ and $\mathbf{G}[\beta\eta]$ are equivalent. However, differently from $\mathbf{G}_{\text{ext}}[\beta]$, $\mathbf{G}[\beta\eta]$ *doesn't admit τ-elimination*. For instance, the equation

$$\lambda x.y((\lambda u.u)x) = y$$

is derivable in $\mathbf{G}[\beta\eta]$:

$$\dfrac{\dfrac{\dfrac{y = y \qquad \dfrac{x = x}{(\lambda u.u)x = x}\,\beta_l}{y((\lambda u.u)x) = yx}\,App}{\lambda x.y((\lambda u.u)x) = \lambda x.yx}\,\xi \qquad \dfrac{y = y}{\lambda x.yx = y}\,\eta_l}{\lambda x.y((\lambda u.u)x) = y}\,\tau$$

but is obviously *not* derivable in $\mathbf{G}[\beta\eta]$ *minus* $[\tau]$. By contrast,

$$\dfrac{\dfrac{\dfrac{y = y \qquad \dfrac{z = z}{(\lambda u.u)z = z}\,\beta_l}{y((\lambda u.u)z) = yz}\,App}{(\lambda x.y((\lambda u.u)x))z = yz}\,\beta_l}{\lambda x.y((\lambda u.u)x) = y}\,Ext$$

is a *τ-free* derivation of the same equation in $\mathbf{G}_{\text{ext}}[\beta]$.

2.5 Preliminaries to τ-elimination

We collect here some notions, notations and auxiliary lemmas which will be needed in the rest of the paper.

Both in $\mathbf{G}[\beta]$ and $\mathbf{G}_{\text{ext}}[\beta]$, derivations $\mathfrak{D}$ will be measured according to their

- *size*: $\mathtt{s}(\mathfrak{D}) :=$ the number of β- and $[Ext]$- inferences occurring in $\mathfrak{D}$;
- *height*: $\mathtt{h}(\mathfrak{D}) :=$ the maximum length (number of nodes) of the branches in $\mathfrak{D}$, minus 1.

Also, for $x \in V$, we let

- $\mathtt{ax}(\mathfrak{D}, x) :=$ the number of occurrences of $[\varrho]$-inferences $x = x$ in $\mathfrak{D}$.

Clearly, w.r. to the dual transformation $\mathfrak{D} \mapsto \widetilde{\mathfrak{D}}$ introduced in 2.3, we have

$$\mathtt{s}(\mathfrak{D}) = \mathtt{s}(\widetilde{\mathfrak{D}}),\ \mathtt{h}(\mathfrak{D}) = \mathtt{h}(\widetilde{\mathfrak{D}}) \quad \textit{and} \quad \mathtt{ax}(\mathfrak{D}, x) = \mathtt{ax}(\widetilde{\mathfrak{D}}, x)\ \textit{for}\ x \in V.$$

A derivation $\mathfrak{D}$ is said to be a *left* (*right*) derivation provided that no $[\beta_r]$ ($[\beta_l]$) inference occurs in $\mathfrak{D}$. We will usually write $\mathfrak{D} \vdash_L t = s$ ($\mathfrak{D} \vdash_R t = s$) to mean that $\mathfrak{D}$ is a left (right) derivation of $t = s$. Obviously,

$$\mathfrak{D} \vdash_{L/R} t = s \quad \text{iff} \quad \widetilde{\mathfrak{D}} \vdash_{R/L} s = t\,.$$

We will also write $\mathfrak{D} \vdash^- t = s$ ($\mathfrak{D} \vdash^-_L t = s, \mathfrak{D} \vdash^-_R t = s$) to mean that $\mathfrak{D}$ is a (left, right) τ-free derivation.

Lemma 2.8 (Parallel substitution) *To each pair of derivations*

$$\mathfrak{D}_1 \vdash t = s \quad \textit{and} \quad \mathfrak{D}_2 \vdash p = q$$

in $\mathbf{G}[\beta]$ (*in* $\mathbf{G}_{\text{ext}}[\beta]$) *and each variable* $x \in V$ *we can effectively associate a derivation*

$$\mathsf{sub}_x(\mathfrak{D}_1, \mathfrak{D}_2) \vdash t[x/p] = s[x/q]$$

in $\mathbf{G}[\beta]$ (*in* $\mathbf{G}_{\text{ext}}[\beta]$) *which is a left* (*right, τ-free*) *derivation provided both* $\mathfrak{D}_1, \mathfrak{D}_2$ *are left* (*right, τ-free*) *derivations. Moreover, for* $\mathfrak{D}^* \equiv \mathsf{sub}_x(\mathfrak{D}_1, \mathfrak{D}_2)$:

$$\mathtt{s}(\mathfrak{D}^*) \leq \mathtt{s}(\mathfrak{D}_1) + \mathtt{ax}(\mathfrak{D}_1, x) \cdot \mathtt{s}(\mathfrak{D}_2) \quad \textit{and} \quad \mathtt{h}(\mathfrak{D}^*) \leq \mathtt{h}(\mathfrak{D}_1) + \mathtt{h}(\mathfrak{D}_2)\,.$$

Proof By straightforward induction on the height of $\mathfrak{D}_1$, taking cases according to its final inference. □

Corollary 2.9 *To each derivation* $\mathfrak{D} \vdash t = s$ *in* $\mathbf{G}[\beta]$ (*in* $\mathbf{G}_{\text{ext}}[\beta]$), *each variable* $x \in V$ *and term* $p \in \Lambda$, *we can effectively associate a derivation*

$$\mathsf{sub}_{x,p}(\mathfrak{D}) \vdash t[x/p] = s[x/p]$$

in $\mathbf{G}[\beta]$ (*in* $\mathbf{G}_{\text{ext}}[\beta]$) *which is a left* (*right,* τ*-free*) *derivation provided* $\mathfrak{D}$ *is a left* (*right,* τ*-free*) *derivation. Moreover, for* $\mathfrak{D}^* \equiv \mathsf{sub}_{x,p}(\mathfrak{D})$:

$$\mathtt{s}(\mathfrak{D}^*) \leq \mathtt{s}(\mathfrak{D}) \quad \textit{and} \quad \mathtt{h}(\mathfrak{D}^*) \leq \mathtt{h}(\mathfrak{D}) + \|p\| \ .$$

Proof For every term p we can easily construct a derivation $\mathfrak{D}_p \vdash^- p = p$ containing structural rules only (so with $\mathtt{s}(\mathfrak{D}_p) = 0$) and such that $\mathtt{h}(\mathfrak{D}_p) = \|p\|$. Then, using Lemma 2.8, we may set $\mathsf{sub}_{x,p}(\mathfrak{D}) := \mathsf{sub}_x(\mathfrak{D}, \mathfrak{D}_p)$. □

Remark 2.10 In view of the above Corollary, precisely of the case in which the term p is a variable, we observe that:

(i) there is *no limitation* in assuming, given a derivation $\mathfrak{D}$ and an [*Ext*]-inference R occurring in it, that the *eigenvariable* of R is distinct from those in an arbitrarily chosen finite set of variables (henceforth, we will sometimes make a tacit use of this fact);
(ii) the *structural* rule [ξ] is trivially eliminable in $\mathbf{G}_{\text{ext}}[\beta]$—without making use of transitivity, but at the cost of using *non-structural* rules:

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\\ t = s\end{matrix}}{t[x/u] = s[x/u]}\ \text{Cor. 2.9, } u \text{ fresh}}{(\lambda x.t)u = (\lambda x.s)u}\ \beta_l, \beta_r}{\lambda x.t = \lambda x.s}\ Ext\ ,$$

and might therefore be dropped. We will nonetheless keep [ξ] as a primitive rule of $\mathbf{G}_{\text{ext}}[\beta]$, since this is *essential* (as the reader will easily realize) to make the proof-strategies in Sects. 5 and 6 work.

Finally, we state here a further special case of parallel substitution, which however will not be used until Sect. 6.

Lemma 2.11 *To each pair of* τ-free $\mathbf{G}_{\text{ext}}[\beta]$*-derivations*

$$\mathfrak{D}_1 \vdash^-_L t = s \quad \textit{and} \quad \mathfrak{D}_2 \vdash^- p = q\,,$$

with $\mathfrak{D}_1$ *a* left *derivation, and to any variable x having* at most one *free occurrence in the term s, we can associate a* τ-free $\mathbf{G}_{\text{ext}}[\beta]$*-derivation*

$$\mathsf{sb}_x(\mathfrak{D}_1, \mathfrak{D}_2) \vdash^- t[x/p] = s[x/q]$$

such that $\mathtt{s}(\mathsf{sb}_x(\mathfrak{D}_1, \mathfrak{D}_2)) \leq \mathtt{s}(\mathfrak{D}_1) + \mathtt{s}(\mathfrak{D}_2)$, *and which moreover is a* left *derivation provided* $\mathfrak{D}_2$ *is such.*

Proof It is easily verified that (i) if $\mathfrak{D} \vdash^-_L t = s$ and the variable x has at most n free occurrences in s, then $\mathtt{ax}(\mathfrak{D}, x) \leq n$ (for $n = 0$, we make use of the fact pointed out in (i) of Remark 2.10). The conclusion follows taking $n = 1$, together with Lemma 2.8.

Note that (i) is not in general true for derivations in which $[\beta_r]$ or $[\tau]$ do occur. □

3 Elimination of transitivity for G[β]

We intend to prove that any given derivation $\mathfrak{D} \vdash t = s$ in $\mathbf{G}[\beta]$ can effectively be transformed into a *τ-free* derivation $\mathsf{e}(\mathfrak{D})$ of the same end-equation. To establish this claim, it would be clearly sufficient to demonstrate—by some kind of inductive argument—that *topmost* applications of $[\tau]$ can be always removed, i.e.:

(◇) *for any given derivation* $\mathfrak{D} \equiv \dfrac{\mathfrak{D}_1\left\{\begin{array}{c}\vdots\\ t = r\end{array}\right. \qquad \left.\begin{array}{c}\vdots\\ r = s\end{array}\right\}\mathfrak{D}_2}{t = s}\,\tau$, *where* $\mathfrak{D}_1$ *and* $\mathfrak{D}_2$ *are τ-free, we can construct a τ-free derivation* $\mathsf{el}(\mathfrak{D}) \vdash^{-} t = s$.

This very natural strategy is indeed successful in proving the eliminability of transitivity for the analytic counterpart $\mathbf{G}[\mathcal{C}]$ of combinatory logic **CL**, see [26], where (◇) is established by transfinite induction on $\omega^2 \cdot \mathtt{h}^{\mathtt{c}}(\mathfrak{D}) + \omega \cdot \mathtt{s}(\mathfrak{D}) + \|r\|$ ($\mathtt{h}^{\mathtt{c}}(\mathfrak{D})$ being a suitable variant of $\mathtt{h}(\mathfrak{D})$), taking cases according to the possible combinations of the final inferences of $\mathfrak{D}_1$ and $\mathfrak{D}_2$.

Unfortunately, such a strategy doesn't seem to be directly feasible in the present context, as one can easily guess by considering the critical case

$$\mathfrak{D} \equiv \dfrac{\dfrac{\dfrac{\mathfrak{D}'_1\left\{\begin{array}{c}\vdots\\ t = r\end{array}\right.}{\lambda x.t = \lambda x.r}\,\xi \qquad \left.\begin{array}{c}\vdots\\ p = q\end{array}\right\}\mathfrak{D}''_1}{(\lambda x.t)p = (\lambda x.r)q}\,App \qquad \dfrac{\mathfrak{D}'_2\left\{\begin{array}{c}\vdots\\ r[x/q] = s\end{array}\right.}{(\lambda x.r)q = s}\,\beta_l}{(\lambda x.t)p = s}\,\tau \tag{i}$$

and the symmetrical one.

On the one side, the only practicable way to get *inductively* a τ-free derivation of $(\lambda x.t)p = s$ out of the given $\mathfrak{D}$ in case (i) (and analogously in the symmetric case) appears to consist first in applying the induction hypothesis to the derivation

$$\mathfrak{D}^* \equiv \dfrac{\mathsf{sub}(\mathfrak{D}'_1, \mathfrak{D}''_1)\left\{\begin{array}{c}\vdots\\ t[x/p] = r[x/q]\end{array}\right. \qquad \left.\begin{array}{c}\vdots\\ r[x/q] = s\end{array}\right\}\mathfrak{D}'_2}{t[x/p] = s}\,\tau\,,$$

whose construction makes an essential use of *parallel substitution* (Lemma 2.8), and then in setting:

$$\mathsf{el}(\mathfrak{D}) := \quad \frac{\mathsf{el}(\mathfrak{D}^*)\left\{\begin{array}{c}\vdots\\ t[x/p] = s\end{array}\right.}{(\lambda x.t)p = s}\,\beta_l\,.$$

On the other side, as a matter of fact it is hard, if not impossible at all, to single out a complexity measure to reason inductively on, one under which $\mathfrak{D}^*$ is "simpler" than the original derivation $\mathfrak{D}$. Note in fact that, w.r. to the most basic complexity parameters such as *rank* (the depth of the *cut-term*), *size* and *height*, it may well happen here that $\|r[x/q]\| > \|(\lambda x.r)q\|$ and, at the same time, $\mathtt{s}(\mathfrak{D}^*) > \mathtt{s}(\mathfrak{D})$ as well as $\mathtt{h}(\mathfrak{D}^*) > \mathtt{h}(\mathfrak{D})$—see the upper bounds given in Lemma 2.8.

In consideration of these obstacles, we are forced to follow a more elaborate and less direct approach. First of all, we will show that (i) the rules $[\beta_l]$ and $[\beta_r]$ admit a *general* form of transitivity-free *inversion* (Theorem 3.4); next, with the aid of (i), we will prove (ii) the analogue of $(\diamond)$ *restricted to the case in which* $\mathfrak{D}_1$ *is a left derivation* (Lemma 3.6), having as an immediate consequence (iii) the τ-eliminability for *left* derivations (Theorem 3.7); further, using (iii), we will show that (iv) a natural generalization of the two β-introduction rules is transitivity-free admissible (Theorem 3.9). Finally, by making essential use of (i) and (iv), we will prove (Lemma 3.10) the *unrestricted* analogue of $(\diamond)$ in which however the original transitivity rule is replaced by the more general rule

$$\frac{t = s \quad \Phi[\![s]\!] = r}{\Phi[\![t]\!] = r}\,\tau^*\,, \quad \text{where } \Phi \text{ is a context with one hole}\,.$$

Whence our initial claim follows as an immediate consequence.

In order to carry out step (i) above, we preliminarily have to introduce *β-marked* λ-terms and other related notions. Intuitively, a β-marked term is a term t in which a number (possibly zero) of occurrences of β-redexes have been marked in some way, so that it makes sense to speak of the (unmarked) term t^* obtained from t by *simultaneously* replacing each marked β-redex $(\lambda x.s)r$ by its corresponding contractum $s[x/r]$. Formally, it is convenient to treat β-marked terms as expressions $M, N, \ldots$ taken from the set $\Lambda' \supseteq \Lambda$, which is defined inductively as follows over the extension of the original alphabet $\mathcal{A}$ by the symbols $\rceil$ and $\lceil$:

- $V \subseteq \Lambda'$
- $M \in \Lambda', x \in V \Rightarrow (\lambda x.M) \in \Lambda'$
- $M, N \in \Lambda' \Rightarrow (MN) \in \Lambda'$
- $M, N \in \Lambda', x \in V \Rightarrow (\lceil(\lambda x.M)N\rceil) \in \Lambda'$.

The usual conventions concerning free and bound variables, α-congruence and substitution (see Sect. 2) shall apply also to Λ'. Note that β-marked terms of the form $\ulcorner(\lambda x.M)N\urcorner$ *are not* application terms (we will call them *red$_\beta$-terms*).

Given a β-marked term M, we denote by $|M|$ the term in Λ which is obtained from M by leaving out every occurrence of the markers '$\ulcorner$' and '$\urcorner$'. In case $t \in \Lambda$ and $|M| \equiv t$, we also say that M is *a β-marking of t*. Clearly, if $|M|$ is a variable, respectively an abstraction term then M is a variable, respectively an abstraction term, while if $|M|$ is an application term then M is not necessarily an application term NP, since it might also be a red$_\beta$-term $\ulcorner(\lambda x.N)P\urcorner$.

Given a β-marked term M, the term $M^* \in \Lambda$ is defined inductively as expected:

- $x^* := x$, for $x \in V$,
- $(\lambda x.M)^* := \lambda x.M^*$,
- $(MN)^* := M^*N^*$,
- $(\ulcorner(\lambda x.M)N\urcorner)^* := M^*[x/N^*]$.

One can easily verify, by induction on the construction of β-marked terms (see e.g. [3], p. 281), that

Fact 3.1 *For every M, N and x:* $(M[x/N])^* \equiv M^*[x/N^*]$.

Also, we clearly have

Fact 3.2 *For every $M \in \Lambda'$:* $FV(M^*) \subseteq FV(|M|) = FV(M)$.

Further to marked terms, we shall also make use of *contexts with one hole*, i.e. expressions $\Phi, \Psi, \ldots$ belonging to the set

$$\mathbb{C} = \{*\} \mid V \mid (\lambda V.\mathbb{C}) \mid (\mathbb{C}\mathbb{C})$$

of unary contexts and containing *at most one* occurrence of $*$.

Given $t \in \Lambda$, $\Phi[\![t]\!]$ denotes the λ-term obtained from the one-hole context Φ by replacing the symbol $*$ with t (caution: variables which are free in t may become bound in $\Phi[\![t]\!]$).

Lemma 3.3 *Every τ-free* $\mathbf{G}[\beta]$*-derivation $\mathfrak{D} \vdash^- t = s$ can effectively be transformed, given an arbitrary marking M of t, into a τ-free* $\mathbf{G}[\beta]$*-derivation*

$$\mathsf{red}_M(\mathfrak{D}) \vdash^- M^* = s$$

which, moreover, is a right *derivation provided $\mathfrak{D}$ is a* right *derivation.*

Proof We argue by main induction on $\mathrm{s}(\mathfrak{D})$ and secondary induction on $\|t\|$, taking cases according to the last inference R of $\mathfrak{D}$.

Case A: $R = [\varrho]$. Trivially, $\mathsf{red}_M(\mathfrak{D}) := \mathfrak{D}$.

Case B: $R = [\xi]$. Then $t \equiv \lambda x.t'$, $s \equiv \lambda x.s'$ and $M \equiv \lambda x.P$ with $|P| \equiv t'$. Also, $M^* \equiv \lambda x.P^*$ and

$$\mathfrak{D} \equiv \quad \frac{\mathfrak{D}_1 \left\{ \begin{array}{c} \vdots \\ t' = s' \end{array} \right.}{\lambda x.t' = \lambda x.s'} \, \xi \, .$$

Since $\mathtt{s}(\mathfrak{D}_1) = \mathtt{s}(\mathfrak{D})$ and $\|t'\| < \|t\|$, we can apply the I.H. to $\mathfrak{D}_1$ and set:

$$\mathsf{red}_M(\mathfrak{D}) := \quad \frac{\mathsf{red}_P(\mathfrak{D}_1) \left\{ \begin{array}{c} \vdots \\ P^* = s' \end{array} \right.}{\lambda x.P^* = \lambda x.s'} \, \xi \, .$$

Case C: $R = [\beta_r]$. Immediate by the (main) I.H.

Case D: $R = [\beta_l]$. Then $t \equiv (\lambda x.p)q_0 \cdots q_n$ $(n \geq 0)$ and

$$\mathfrak{D} \equiv \quad \frac{\mathfrak{D}_1 \left\{ \begin{array}{c} \vdots \\ p[x/q_0]q_1 \cdots q_n = s \end{array} \right.}{(\lambda x.p)q_0 \cdots q_n = s} \, \beta_l \, .$$

On the other side, M is either $(\lambda x.P)Q_0 \cdots Q_n$ or $\lceil (\lambda x.P)Q_0 \rceil \, Q_1 \cdots Q_n$, with $|P| \equiv p$ and $|Q_i| \equiv q_i$ for $0 \leq i \leq n$. In the first case, $M^* \equiv (\lambda x.P^*)Q_0^* \cdots Q_n^*$; in the second case, $M^* \equiv P^*[x/Q_0^*]Q_1^* \cdots Q_n^*$.

Since $\mathtt{s}(\mathfrak{D}_1) < \mathtt{s}(\mathfrak{D})$ and $(P[x/Q_0]Q_1 \cdots Q_n)^* \equiv P^*[x/Q_0^*]Q_1^* \cdots Q_n^*$ by Fact 3.1, by the I.H. we can take:

$$\mathsf{red}_M(\mathfrak{D}) := \quad \frac{\mathsf{red}_{P[x/Q_0]Q_1 \cdots Q_n}(\mathfrak{D}_1) \left\{ \begin{array}{c} \vdots \\ P^*[x/Q_0^*]Q_1^* \cdots Q_n^* = s \end{array} \right.}{(\lambda x.P^*)Q_0^* \cdots Q_n^* = s} \, \beta_l$$

in the first case, and

$$\mathsf{red}_M(\mathfrak{D}) := \mathsf{red}_{P[x/Q_0]Q_1 \cdots Q_n}(\mathfrak{D}_1)$$

in the second one.

Case E: $R = [App]$. Then $|M| \equiv t$ is an application term (as well as $s \equiv s_1 s_2$), and so we have two subcases to consider.

Subcase E.1: $M \equiv PQ$ with $|P| \equiv p$, $|Q| \equiv q$, $t \equiv pq$. Then $M^* \equiv P^* Q^*$ and

$$\mathfrak{D} \equiv \quad \frac{\mathfrak{D}_1 \left\{ \begin{array}{c} \vdots \\ p = s_1 \end{array} \right. \qquad \left. \begin{array}{c} \vdots \\ q = s_2 \end{array} \right\} \mathfrak{D}_2}{pq = s_1 s_2} \, App \, .$$

Since $\mathtt{s}(\mathfrak{D}_1), \mathtt{s}(\mathfrak{D}_2) \leq \mathtt{s}(\mathfrak{D})$ and $\|p\|, \|q\| < \|t\|$, by the I.H. we may set:

$$\mathsf{red}_M(\mathfrak{D}) := \quad \dfrac{\mathsf{red}_P(\mathfrak{D}_1)\left\{\begin{matrix}\vdots\\ P^* = s_1\end{matrix}\right. \qquad \left.\begin{matrix}\vdots\\ Q^* = s_2\end{matrix}\right\}\mathsf{red}_Q(\mathfrak{D}_2)}{P^*Q^* = s_1 s_2}\, App \,.$$

<u>Subcase E.2</u>: $M \equiv \ulcorner(\lambda x.P)Q\urcorner$ with $|P| \equiv p$, $|Q| \equiv q$, and $t \equiv (\lambda x.p)q$. Then $M^* \equiv P^*[x/Q^*]$ and

$$\mathfrak{D} \equiv \quad \dfrac{\mathfrak{D}_1\left\{\dfrac{\vdots}{\lambda x.p = s_1}\, R'\right. \qquad \left.\begin{matrix}\vdots\\ q = s_2\end{matrix}\right\}\mathfrak{D}_2}{(\lambda x.p)q = s_1 s_2}\, App \,.$$

Necessarily, R' is either $[\xi]$ or $[\beta_r]$.

[E.2.a]: if R' is $[\xi]$, then $s_1 \equiv \lambda x.s_1'$ and

$$\mathfrak{D} \equiv \quad \dfrac{\dfrac{\mathfrak{D}_1'\left\{\begin{matrix}\vdots\\ p = s_1'\end{matrix}\right.}{\lambda x.p = \lambda x.s_1'}\, \xi \qquad \left.\begin{matrix}\vdots\\ q = s_2\end{matrix}\right\}\mathfrak{D}_2}{(\lambda x.p)q = (\lambda x.s_1')s_2}\, App \,.$$

Since $\mathtt{s}(\mathfrak{D}_1'), \mathtt{s}(\mathfrak{D}_2) \leq \mathtt{s}(\mathfrak{D})$ and $\|p\|, \|q\| < \|t\|$, we can apply the I.H. to $\mathfrak{D}_1'$ and $\mathfrak{D}_2$ and take, by making use also of the *Parallel substitution* Lemma 2.8:

$$\mathsf{red}_M(\mathfrak{D}) := \quad \dfrac{\dfrac{\mathsf{red}_P(\mathfrak{D}_1')\left\{\begin{matrix}\vdots\\ P^* = s_1'\end{matrix}\right. \qquad \left.\begin{matrix}\vdots\\ Q^* = s_2\end{matrix}\right\}\mathsf{red}_Q(\mathfrak{D}_2)}{P^*[x/Q^*] = s_1'[x/s_2]}\, SUB}{P^*[x/Q^*] = (\lambda x.s_1')s_2}\, \beta_r$$

[E.2.b]: if R' is $[\beta_r]$, then $s_1 \equiv (\lambda y.s'')s_0' \cdots s_n'$ $(n \geq 0)$ and

$$\mathfrak{D} \equiv \quad \dfrac{\dfrac{\mathfrak{D}_1'\left\{\begin{matrix}\vdots\\ \lambda x.p = s''[y/s_0']s_1' \cdots s_n'\end{matrix}\right.}{\lambda x.p = (\lambda y.s'')s_0' \cdots s_n'}\, \beta_r \qquad \left.\begin{matrix}\vdots\\ q = s_2\end{matrix}\right\}\mathfrak{D}_2}{(\lambda x.p)q = (\lambda y.s'')s_0' \cdots s_n' s_2}\, App \,.$$

In this case we consider the derivation

$$\mathfrak{D}' \equiv \quad \dfrac{\mathfrak{D}_1' \left\{ \begin{array}{c} \vdots \\ \lambda x.p = s''[y/s_0']s_1' \ldots s_n' \end{array} \right. \qquad \left. \begin{array}{c} \vdots \\ q = s_2 \end{array} \right\} \mathfrak{D}_2}{(\lambda x.p)q = s''[y/s_0']s_1' \ldots s_n' s_2} \, App \, .$$

Since $\mathtt{s}(\mathfrak{D}') < \mathtt{s}(\mathfrak{D})$, by the I.H. we may set:

$$\mathsf{red}_M(\mathfrak{D}) := \quad \dfrac{\mathsf{red}_M(\mathfrak{D}') \left\{ \begin{array}{c} \vdots \\ P^*[x/Q^*] = s''[y/s_0']s_1' \ldots s_n' s_2 \end{array} \right.}{P^*[x/Q^*] = (\lambda y.s'')s_0' \ldots s_n' s_2} \, \beta_r \, .$$

By inspecting the whole proof, it is easily verified that the transformation red_M maps *right* derivations into *right* derivations (whereas left derivations *need not* be transformed into left derivations: consider subcase E.2.a). □

Theorem 3.4 (generalized **β**-inversion) *Every τ-free* **G**[β]*-derivation*

$$\mathfrak{D} \vdash^- \Phi[\![(\lambda x.t)s_0 \ldots s_n]\!] = r \quad (\Phi \textit{ a context with one hole}; n \geq 0)$$

can effectively be transformed into a τ-free **G**[β]*-derivation*

$$\mathsf{inv}_\beta(\mathfrak{D}) \vdash^- \Phi[\![t[x/s_0]s_1 \cdots s_n]\!] = r$$

which, moreover, is a right *derivation provided* $\mathfrak{D}$ *is a* right *derivation.*

Proof Given $\mathfrak{D}$, we consider the marked term $M \equiv \Phi[\![\lceil(\lambda x.t)s_0\rceil s_1 \cdots s_n]\!]$ and take, using Lemma 3.3: $\mathsf{inv}_\beta(\mathfrak{D}) := \mathsf{red}_M(\mathfrak{D})$. □

Remark 3.5 Note that possibly $\mathtt{s}(\mathsf{inv}_\beta(\mathfrak{D})) > \mathtt{s}(\mathfrak{D})$, i.e. the *size* of derivations is not, and in fact cannot be, preserved in general under the inversion process; e.g., trivially, $(\lambda x.x)y = (\lambda x.x)y$ has a τ-free derivation of size 0, but there are no derivations of size 0 of $y = (\lambda x.x)y$.

We now exploit Theorem 3.4 to prove a *partial* elimination lemma.

Lemma 3.6 *Any given pair*

$$\mathfrak{D}_1 \vdash_L^- t = s \quad \textit{and} \quad \mathfrak{D}_2 \vdash^- s = r$$

of τ-free **G**[β]*-derivations,* with $\mathfrak{D}_1$ a left derivation, *can effectively be transformed into a τ-free* **G**[β]*-derivation*

$$\mathsf{el}_\mathsf{L}(\mathfrak{D}_1, \mathfrak{D}_2) \vdash^- t = r$$

which, moreover, is a left *derivation provided that* $\mathfrak{D}_2$ *is a* left *derivation.*

Proof Let R_1 and R_2 be the final inferences of $\mathfrak{D}_1$, respectively $\mathfrak{D}_2$. The proof runs by induction on $\omega^2 \cdot \mathrm{s}(\mathfrak{D}_2) + \omega \cdot \mathrm{s}(\mathfrak{D}_1) + \|s\|$, taking cases according to the possible combinations $\langle R_1, R_2\rangle$—under the key assumption that $\mathfrak{D}_1$ is a *left* derivation.

Case A: $R_1 = [\varrho]$. Trivially, we take $\mathsf{el}_\mathsf{L}(\mathfrak{D}_1, \mathfrak{D}_2) := \mathfrak{D}_2$.

Case B: $R_2 = [\beta_r]$. Then, for a suitable r', $\mathfrak{D}_2$ has the form

$$\dfrac{\mathfrak{D}'_2\left\{\begin{array}{c}\vdots\\ s = r'\end{array}\right.}{s = r}\,\beta_r, \quad \text{where } \mathrm{s}(\mathfrak{D}'_2) < \mathrm{s}(\mathfrak{D}_2).$$

So we can apply the (main) I.H. to $\mathfrak{D}_1$ and $\mathfrak{D}'_2$, and set:

$$\mathsf{el}_\mathsf{L}(\mathfrak{D}_1, \mathfrak{D}_2) := \qquad \dfrac{\mathsf{el}_\mathsf{L}(\mathfrak{D}_1, D'_2)\left\{\begin{array}{c}\vdots\\ t = r'\end{array}\right.}{t = r}\,\beta_r\,.$$

Case C: $R_1 = [\beta_l]$. Symmetrically (but as the induction measure is not symmetric, note that the secondary I.H. is used here).

Case D: $R_1 = [\xi]$. Necessarily, R_2 is either $[\beta_r]$ (see case B) or $[\xi]$, in which case $t \equiv \lambda x.t'$, $s \equiv \lambda x.s'$, $r \equiv \lambda x.r'$ and

$$\mathfrak{D}_1 \equiv \dfrac{\mathfrak{D}'_1\left\{\begin{array}{c}\vdots\\ t' = s'\end{array}\right.}{\lambda x.t' = \lambda x.s'}\,\xi\,, \quad \mathfrak{D}_2 \equiv \dfrac{\mathfrak{D}'_2\left\{\begin{array}{c}\vdots\\ s' = r'\end{array}\right.}{\lambda x.s' = \lambda x.r'}\,\xi\,,$$

where $\mathrm{s}(\mathfrak{D}'_1) = \mathrm{s}(\mathfrak{D}_1)$, $\mathrm{s}(\mathfrak{D}'_2) = \mathrm{s}(\mathfrak{D}_2)$.

Since $\|s'\| < \|s\|$, by the (ternary) I.H. we can take

$$\mathsf{el}_\mathsf{L}(\mathfrak{D}_1, \mathfrak{D}_2) := \qquad \dfrac{\mathsf{el}_\mathsf{L}(\mathfrak{D}'_1, \mathfrak{D}'_2)\left\{\begin{array}{c}\vdots\\ t' = r'\end{array}\right.}{\lambda x.t' = \lambda x.r'}\,\xi\,.$$

Case E: $R_1 = [App]$. Taking into account that Case B has been already discussed, we need to distinguish only two subcases depending on R_2.

Subcase E.1: $R_2 = [App]$. Then the conclusion immediately follows by the (ternary, in the worst case) I.H.

Subcase E.2: $R_2 = [\beta_l]$. Then $s \equiv (\lambda x.s')s_0 \ldots s_n$ $(n \geq 0)$ and

$$\mathfrak{D}_2 \equiv \qquad \dfrac{\mathfrak{D}'_2\left\{\begin{array}{c}\vdots\\ s'[x/s_0]s_1 \cdots s_n = r\end{array}\right.}{(\lambda x.s')s_0 \cdots s_n = r}\,\beta_l, \quad \text{where } \mathrm{s}(\mathfrak{D}'_2) < \mathrm{s}(\mathfrak{D}_2)\,.$$

On the other side, from

$$\mathfrak{D}_1 \vdash_L^- t = (\lambda x.s')s_0 \cdots s_n$$

it follows that

$$\widetilde{\mathfrak{D}_1} \vdash_R^- (\lambda x.s')s_0 \cdots s_n = t$$

and so, by Theorem 3.4 (with $\Phi \equiv *$)

$$\mathsf{inv}_\beta(\widetilde{\mathfrak{D}_1}) \vdash_R^- s'[x/s_0]s_1 \cdots s_n = t$$

i.e., letting $\mathfrak{D}_1' := \widetilde{\mathsf{inv}_\beta(\widetilde{\mathfrak{D}_1})}$,

$$\mathfrak{D}_1' \vdash_L^- t = s'[x/s_0]s_1 \cdots s_n.$$

Although in general $\mathrm{s}(\mathfrak{D}_1') \nless \mathrm{s}(\mathfrak{D}_1)$, the fact that $\mathrm{s}(\mathfrak{D}_2') < \mathrm{s}(\mathfrak{D}_2)$ allows us to apply the I.H. to the pair $\mathfrak{D}_1', \mathfrak{D}_2'$, and to set:

$$\mathsf{el_L}(\mathfrak{D}_1, \mathfrak{D}_2) := \mathsf{el_L}(\mathfrak{D}_1', \mathfrak{D}_2').$$

By inspecting the whole proof, it is easily verified that the transformation $\mathsf{el_L}$ maps pairs of τ-free *left* derivations into τ-free *left* derivations. □

Theorem 3.7 (Left τ-elimination) *Every* left $\mathbf{G}[\beta]$*-derivation* $\mathfrak{D}$ *can effectively be transformed into a* τ-free left $\mathbf{G}[\beta]$*-derivation* $\mathsf{e}_L(\mathfrak{D})$ *having the same end-equation as* $\mathfrak{D}$.

Proof By induction on the number of occurrences of rule $[\tau]$ in $\mathfrak{D}$, using Lemma 3.6 to eliminate a topmost occurrence of $[\tau]$. □

Remark 3.8 Note that the form of β-inversion which is actually used in the proof of Lemma 3.6 (see case E.2), on which Theorem 3.7 depends, is:

$$\mathfrak{D} \vdash_L^- t = (\lambda x.p_0)p_1 \cdots p_n \;\Rightarrow\; \exists \mathfrak{D}^* \,(\, \mathfrak{D}^* \vdash_L^- t = p_0[x/p_1]p_2 \cdots p_n\,).$$

This is much weaker than what is provided for by Theorem 3.4, and might be proved without making use of the marking technique. The full generality of Theorem 3.4 is instead needed in the proof of Lemma 3.10 below.

We can now prove:

Theorem 3.9 (generalized β-introduction) *The calculus* $\mathbf{G}^-[\beta]$ *is closed under the rules*:

$$\frac{\Phi[\![t[x/s_0]s_1 \cdots s_n]\!] = r}{\Phi[\![(\lambda x.t)s_0 \cdots s_n]\!] = r}\,\beta_l^+ \quad \textit{and} \quad \frac{r = \Phi[\![t[x/s_0]s_1 \cdots s_n]\!]}{r = \Phi[\![(\lambda x.t)s_0 \cdots s_n]\!]}\,\beta_r^+,$$

where Φ is a context with one hole and $n \geq 0$. Moreover, τ-free left (right) derivability is closed under β_l^+ (β_r^+).

Proof Given any context Φ, it is easily seen that a *left*, *τ-free* derivation

$$\mathfrak{D}_1 \vdash_L^- \Phi[\![(\lambda x.t)s_0 \cdots s_n]\!] = \Phi[\![t[x/s_0]s_1 \cdots s_n]\!]$$

with $\mathrm{s}(\mathfrak{D}_1) = 1$ can be constructed. So, for any given derivation

$$\mathfrak{D}_2 \vdash_{(L)}^- \Phi[\![t[x/s_0]s_1 \cdots s_n]\!] = r$$

we have, by Lemma 3.6,

$$\mathsf{el}_{\mathsf{L}}(\mathfrak{D}_1, \mathfrak{D}_2) \vdash_{(L)}^- \Phi[\![(\lambda x.t)s_0 \cdots s_n]\!] = r\,.$$

The rest of the claim follows by using the dual transformation $\tilde{\ }$. □

We are finally able to remove from Lemma 3.6 the limitation that $\mathfrak{D}_1$ be a *left* derivation.

Lemma 3.10 (main elimination lemma) *To each pair of τ-free* $\mathbf{G}[\beta]$*-derivations*

$$\mathfrak{D}_1 \vdash^- p = q \quad \textit{and} \quad \mathfrak{D}_2 \vdash^- \Phi[\![q]\!] = r$$

we can effectively associate a τ-free $\mathbf{G}[\beta]$*-derivation*

$$\mathsf{el}_{\Phi}(\mathfrak{D}_1, \mathfrak{D}_2) \vdash^- \Phi[\![p]\!] = r\,.$$

Proof By main induction on $\mathrm{s}(\mathfrak{D}_1)$ and secondary induction on $\|q\|$. We let R be the last inference in $\mathfrak{D}_1$, and consider all the possible cases.
Case A: $R = [\varrho]$. Trivially, we take $\mathsf{el}_{\Phi}(\mathfrak{D}_1, \mathfrak{D}_2) := \mathfrak{D}_2$.
Case B: $R = [\xi]$. Then $p \equiv \lambda x.p'$, $q \equiv \lambda x.q'$ and

$$\mathfrak{D}_1 \equiv \cfrac{\mathfrak{D}_1' \left\{ \begin{array}{c} \vdots \\ p' = q' \end{array} \right.}{\lambda x.p' = \lambda x.q'}\ \xi\,.$$

Consider the context $\Psi := \Phi[\![\lambda x. *]\!]$. Then $\Phi[\![q]\!] \equiv \Phi[\![\lambda x.q']\!] \equiv \Psi[\![q']\!]$, and so

$$\mathfrak{D}_2 \vdash^- \Psi[\![q']\!] = r.$$

Now $\mathrm{s}(\mathfrak{D}_1') = \mathrm{s}(\mathfrak{D}_1)$ and $\|q'\| < \|q\|$, so by applying the I.H. we can take

$$\mathsf{el}_{\Phi}(\mathfrak{D}_1, \mathfrak{D}_2) := \mathsf{el}_{\Psi}(\mathfrak{D}_1', \mathfrak{D}_2)\,,$$

since $\Psi[\![p']\!] \equiv \Phi[\![\lambda x.p']\!] \equiv \Phi[\![p]\!]$.

Case C: $R = [App]$. Then $p \equiv p_1 p_2,\ q \equiv q_1 q_2$, and

$$\mathfrak{D}_1 \equiv \cfrac{\mathfrak{D}_1' \left\{ \begin{matrix} \vdots \\ p_1 = q_1 \end{matrix} \right. \qquad \left. \begin{matrix} \vdots \\ p_2 = q_2 \end{matrix} \right\} \mathfrak{D}_1''}{p_1 p_2 = q_1 q_2}\ {\scriptstyle App}\,.$$

Let $\Psi := \Phi[\![* q_2]\!]$, so that $\Psi[\![q_1]\!] \equiv \Phi[\![q]\!]$ and

$$\mathfrak{D}_2 \vdash^- \Psi[\![q_1]\!] = r\,.$$

Since $\mathrm{s}(\mathfrak{D}_1') \leq \mathrm{s}(\mathfrak{D}_1)$ and $\|q_1\| < \|q\|$, we may apply the I.H. to get:

$$\mathsf{el}_\Psi(\mathfrak{D}_1', \mathfrak{D}_2) \vdash^- \Psi[\![p_1]\!] = r\,,$$

i.e.

$$\mathsf{el}_\Psi(\mathfrak{D}_1', \mathfrak{D}_2) \vdash^- \Phi[\![p_1 q_2]\!] = r\,.$$

Now, let $\Theta := \Phi[\![p_1 *]\!]$, so that $\Theta[\![q_2]\!] \equiv \Phi[\![p_1 q_2]\!]$ and $\Theta[\![p_2]\!] \equiv \Phi[\![p]\!]$.
Since $\mathrm{s}(\mathfrak{D}_1'') \leq \mathrm{s}(\mathfrak{D}_1)$ and $\|q_2\| < \|q\|$, by applying again the I.H. we may set:

$$\mathsf{el}_\Phi(\mathfrak{D}_1, \mathfrak{D}_2) := \quad \mathsf{el}_\Theta(\mathfrak{D}_1'', \mathsf{el}_\Psi(\mathfrak{D}_1', \mathfrak{D}_2)).$$

Case D: $R = [\beta_l]$. Then $p \equiv (\lambda x.p')p_0 \cdots p_n$, and $\mathfrak{D}_1$ has the form

$$\cfrac{\mathfrak{D}_1' \left\{ \begin{matrix} \vdots \\ p'[x/p_0]p_1 \cdots p_n = q \end{matrix} \right.}{(\lambda x.p')p_0 \cdots p_n = q}\ {\scriptstyle \beta_l}, \quad \text{where } \mathrm{s}(\mathfrak{D}_1') < \mathrm{s}(\mathfrak{D}_1).$$

By applying the I.H. and Theorem 3.9, we may set

$$\mathsf{el}_\Phi(\mathfrak{D}_1, \mathfrak{D}_2) := \quad \cfrac{\mathsf{el}_\Phi(\mathfrak{D}_1', \mathfrak{D}_2) \left\{ \begin{matrix} \vdots \\ \Phi[\![p'[x/p_0]p_1 \cdots p_n]\!] = r \end{matrix} \right.}{\Phi[\![(\lambda x.p')p_0 \cdots p_n]\!] = r}\ {\scriptstyle \beta_l^+}$$

Case E: $R = [\beta_r]$. Then $q \equiv (\lambda x.q')q_0 \cdots q_n$, and $\mathfrak{D}_1$ has the form

$$\cfrac{\mathfrak{D}_1' \left\{ \begin{matrix} \vdots \\ p = q'[x/q_0]q_1 \cdots q_n \end{matrix} \right.}{p = (\lambda x.q')q_0 \cdots q_n}\ {\scriptstyle \beta_r}, \quad \text{where } \mathrm{s}(\mathfrak{D}_1') < \mathrm{s}(\mathfrak{D}_1).$$

On the other side, by Theorem 3.4

$$\mathsf{inv}_\beta(\mathfrak{D}_2) \vdash^- \Phi[\![q'[x/q_0]q_1 \cdots q_n]\!] = r\,;$$

so by applying the I.H. we may take

$$\mathsf{el}_\Phi(\mathfrak{D}_1, \mathfrak{D}_2) := \mathsf{el}_\Phi(\mathfrak{D}'_1, \mathsf{inv}_\beta(\mathfrak{D}_2))\,.$$

□

Theorem 3.11 (τ-elimination for $\mathbf{G}[\beta]$) *Every* $\mathbf{G}[\beta]$*-derivation* $\mathfrak{D}$ *can effectively be transformed into a* τ-free $\mathbf{G}[\beta]$*-derivation* $\mathsf{e}(\mathfrak{D})$ *having the same end–equation as* $\mathfrak{D}$.

Proof By induction on the number of occurrences of rule $[\tau]$ in $\mathfrak{D}$, using the *Main elimination lemma* with $\Phi \equiv *$ to eliminate a topmost occurrence of $[\tau]$. □

4 Transitivity elimination in G[β] and β-reduction

Throughout this section, derivability ($\vdash$) means derivability in $\mathbf{G}[\beta]$. The symbols $\rightarrow$ and $\twoheadrightarrow$ denote the usual relations on Λ of *one-step* β*-reduction* and, respectively, β*-reduction*, the latter being the reflexive and transitive closure of $\rightarrow$. The letters $\sigma, \vartheta, \ldots$ vary over (possibly empty) *reduction paths*, i.e. finite or infinite sequences $t_0 \rightarrow t_1 \rightarrow t_2 \rightarrow \cdots$ of one-step reductions, and $\sigma : t \twoheadrightarrow s$ means that σ is a finite, possibly empty reduction path starting from t and ending with s. NF_β denotes the set of all terms in β*-normal form*; a term t *has* a β-normal form provided $t \twoheadrightarrow s$ for some $s \in \mathrm{NF}_\beta$.

We will show how two well known, central results concerning β-reduction, namely the *Church–Rosser* Theorem and the (weak) *Standardization* Theorem, can be quite easily obtained by exploiting the τ*-Elimination* Theorems 3.11 and 3.7.

Let us start with two simple lemmas, which do not depend on τ-elimination.

Lemma 4.1

(i) *If* $t \rightarrow s$ *then we can find a* τ*-free* left *derivation* $\mathfrak{D}_{t,s} \vdash^-_L t = s$.
(ii) *To each reduction path* $\sigma : t \twoheadrightarrow s$ *we can effectively associate a* left *derivation* $\mathsf{lder}(\sigma) \vdash_L t = s$.

Lemma 4.2 *To each* τ*-free derivation* $\mathfrak{D} \vdash^- t = s$ *we can associate a term* $r_{\mathfrak{D}}$ *such that* $t \twoheadrightarrow r_{\mathfrak{D}} \twoheadleftarrow s$.

The easy verification of (i) of Lemma 4.1 runs by induction on the generation of $\rightarrow_\beta$, while (ii) follows from (i) by making use of the transitivity rule. In turn, Lemma 4.2 is proved by a straightforward induction on the height of τ*-free* derivations.

And we are now in a position to get Church–Rosser's Theorem almost for free.

Theorem 4.3 (CR($\boldsymbol{\beta}$)) $\twoheadrightarrow$ *is Church–Rosser.*

Proof Suppose $s_1 \twoheadleftarrow t \twoheadrightarrow s_2$. By Lemma 4.1 both $t = s_1$ and $t = s_2$ are (left) derivable in $\mathbf{G}[\beta]$ and so also, by symmetry (which is admissible by Fact 2.1) and transitivity:

$$\vdash s_1 = s_2\,.$$

Then, by the Elimination Theorem 3.11 (notice: *Left* τ-elimination = Theorem 3.7 is not sufficient here!)

$$\vdash^{-} s_1 = s_2\,,$$

whence the conclusion immediately follows by Lemma 4.2. □

Remark 4.4 Uniqueness of β-normal forms

$$(t \twoheadrightarrow r,\ t \twoheadrightarrow s,\ r,s \in \mathrm{NF}_\beta) \ \Rightarrow\ r \equiv s. \qquad (\mathrm{UNF}_\beta)$$

is usually obtained as a corollary of CR(β). In the present context, also the following variant proof (which does not pass through CR) of UNF_β should be noticed.

Assume $t \twoheadrightarrow r$ and $t \twoheadrightarrow s$, with $r,s \in \mathrm{NF}_\beta$. Then, as above, there exists a τ-free derivation $\mathfrak{D}$ of $r = s$, and the conclusion follows by (ii) of Lemma 2.3.

As a second application of τ-elimination—actually of the weaker *Left* τ-elimination Theorem 3.7—a considerably short and simple proof of the (weak) *Standardization* theorem can be given. We recall that a β-reduction path σ from t to s is *standard* ($\sigma : t \twoheadrightarrow_{\mathfrak{s}} s$) provided it consists in a sequence of contractions proceeding *from left to right, possibly with some jumps*. The official ("geometric") definition (see e.g. [3], Chap. 11, Sect. 4) in terms of *residuals* runs as follows:

- $\sigma : t \twoheadrightarrow_{\mathfrak{s}} s \quad := \quad \sigma : t \equiv t_0 \to t_1 \to \cdots \to t_{n-1} \to t_n \equiv s \quad (n \geq 0)$, where in each step the contracted redex *is not* a residual of a redex to the *left* of a previously contracted one;
- $t \twoheadrightarrow_{\mathfrak{s}} s \quad := \quad \exists\sigma(\sigma : t \twoheadrightarrow_{\mathfrak{s}} s)$.

The (weak) *Standardization* theorem states that if a λ-term t β-reduces to a λ-term s, then there exists also a standard reduction from t to s. Starting from the original 1958 proof in Curry and Feys [13], a lot of interesting different proofs of this theorem (usually providing algorithms for extracting a standard $\sigma' : t \twoheadrightarrow_{\mathfrak{s}} s$ out of a given $\sigma : t \twoheadrightarrow s$) have been made available in the literature, among which a well known and very short one due to Mitschke [28] (using a technique combining *internal* reduction and *head* reduction) and, in the last decade, those by David [15] (where standardization is directly proved in a sharpened form which readily yields the *Finiteness of developments* theorem as a corollary), Takahashi [29] (exploiting Tait – Martin Löf's notion of *parallel* reduction), McKinna and Pollack [24] (formally developed in the LEGO Proof System), Xi [30] (providing a standardization algorithm ST together with an upper bound on the number of reduction steps in the standardization $\mathsf{ST}(\sigma)$ of a given reduction path σ) and Kashima [19] (avoiding the notion of residual, and using instead a clever inductive characterization of standard reduction paths). Further proofs, exploiting untyped variants of normalization-by-evaluation, can be found in Aehlig and Joachimski [1] (working in a syntactic, higher-order rewriting systems setting) and, at least implicitly, in Filinski and Rohde [16] (based on a semantical, domain-theoretical approach). Mention should be made also of the *strong* version of the Standardization theorem due to Lévy [22] and relying on the Berry – Lévy notion of *strong equivalence* between β-reduction paths (see e.g. [3], Chap. 12, Sect. 1). It states that *for every* $\sigma : t \twoheadrightarrow s$ *there is a unique standard reduction* $\sigma' : t \twoheadrightarrow_{s} s$ *which is strongly equivalent to* σ. Further proofs of this result have been given by Klop [21], Gonthier et al. [17], Melliès [25].

Let us now come to our proof of the weak version. All we need, besides Theorem 3.7, are just the following three properties of *standard β-reducibility* on Λ:

(S.1) $t \twoheadrightarrow_{s} s$ *implies* $\lambda x.t \twoheadrightarrow_{s} \lambda x.s$,
(S.2) $t_1 \twoheadrightarrow_{s} s_1$ *and* $t_2 \twoheadrightarrow_{s} s_2$ *implies* $t_1 t_2 \twoheadrightarrow_{s} s_1 s_2$,
(S.3) $p[x/p_0]p_1 \cdots p_k \twoheadrightarrow_{s} s$ *implies* $(\lambda x.p)p_0 \cdots p_k \twoheadrightarrow_{s} s$.

In order to easily verify properties (S.1)–(S.3) (indeed, only the second one is non trivial) one might e.g. employ Kashima's [19] *residual-free* inductive characterization of $\twoheadrightarrow_{s}$. Just to make our paper self-contained, we instead propose a further, convenient characterization of $\twoheadrightarrow_{s}$ which is still residual-free and is based on the marking technique.

Let $\Lambda' \supseteq \Lambda$ be the set of all terms (which we will still denote by $t, s, \ldots$) which are obtained from the terms in Λ by replacing some occurrences (possibly none) of the symbol λ with the symbol λ′. Next, given $t \in \Lambda'$, let

- t^+ be the term obtained by changing *every* λ occurring in t into λ′ ,
- t^- be the term obtained by changing *every* λ′ occurring in t into λ .

Clearly, $t^- \in \Lambda$ and $t^{++} \equiv t^+$, for every $t \in \Lambda'$.

We next define inductively the *one-step β′-reduction* relation $\rightarrowtail$ on Λ' by means of the clauses:

- $(\lambda x.t)s \rightarrowtail t[x/s]$;
- if $t \rightarrowtail s$, then $\lambda x.t \rightarrowtail \lambda' x.s,\ \lambda' x.t \rightarrowtail \lambda' x.s,\ tr \rightarrowtail sr,\ rt \rightarrowtail r^+ s$.

Finally, we let $\rightarrowtail\!\!\!\!\rightarrow$ be the reflexive and transitive closure of $\rightarrowtail$. It is then easily verified that

Fact 4.5 *For every term* t *and* $s \in \Lambda$:

$$t \twoheadrightarrow_{s} s \quad \text{if and only if} \quad t \rightarrowtail\!\!\!\!\rightarrow s' \text{ for some } s' \in \Lambda' \text{ such that } s'^{-} \equiv s.$$

Now, given two arbitrary β'-reduction paths

$$\sigma : t_0 \rightarrowtail t_1 \rightarrowtail \cdots \rightarrowtail t_n\,, \quad \vartheta : s_0 \rightarrowtail s_1 \rightarrowtail \cdots \rightarrowtail s_m\,,$$

where $n, m \geq 0$, $t_i, s_i \in \Lambda'$ and $t_0, s_0 \in \Lambda$, it is clear that the following are β'-reduction paths:

- $\lambda_x(\sigma) := \lambda x.t_0 \rightarrowtail \lambda' x.t_1 \rightarrowtail \cdots \rightarrowtail \lambda' x.t_n\,,$
- $\sigma \cdot \vartheta := t_0 s_0 \rightarrowtail t_1 s_0 \rightarrowtail \cdots \rightarrowtail t_n s_0 \rightarrowtail t_n^+ s_1 \rightarrowtail t_n^+ s_2 \rightarrowtail \cdots t_n^+ s_m$

and, for every *presentation* π of t_0 as $p[x/p_0]p_1 \cdots p_k$ with $k \geq 0$, also

- $\beta_\pi(\sigma) := (\lambda x.p)p_0 \cdots p_k \rightarrowtail p[x/p_0]p_1 \cdots p_k \rightarrowtail t_1 \rightarrowtail \cdots \rightarrowtail t_n\,.$

In view of Fact 4.5, it therefore follows that the properties (S.1)–(S.3) are satisfied by $\twoheadrightarrow_{\mathfrak{s}}$. Thus:

Lemma 4.6 *To any given τ-free* left *derivation $\mathfrak{D}$ of $t = s$ we can effectively associate a* standard *reduction path* $\mathsf{sp}(\mathfrak{D}) : t \twoheadrightarrow_{\mathfrak{s}} s$.

Proof By straightforward induction on $\mathtt{h}(\mathfrak{D})$, using (S.1)–(S.3) above; it is clear that $\mathsf{sp}(\mathfrak{D})$ may be explicitly presented by means of the operators $\lambda_x\,, \cdot$ and β_π, uniformly in $\mathfrak{D}$. □

Finally, for any finite β-reduction path σ, let

$$\mathsf{std}(\sigma) := \mathsf{sp}(\mathsf{e}_L(\mathsf{lder}(\sigma)))\,,$$

where the operators $\mathsf{sp}\,, \mathsf{e}_L$ and lder are given, respectively, by Lemma 4.6, Theorem 3.7 and Lemma 4.1. We have

Theorem 4.7 (Standardization) *If $\sigma : t \twoheadrightarrow s$ then* $\mathsf{std}(\sigma) : t \twoheadrightarrow_{\mathfrak{s}} s$.

Two important, almost immediate consequences of the (weak) *Standardization* Theorem 4.6, together with the Church–Rosser Theorem 4.3, are worth to be recalled here (cf. [3], Corollary 13.2.2 and Corollary 11.4.8):

(1) *The leftmost reduction strategy is normalizing (*Normalization *theorem).*
(2) *t has a head normal form iff the head reduction path of t terminates.*

5 Elimination of transitivity for $\mathbf{G}_{\mathrm{ext}}[\beta]$

In order to show the eliminability of the transitivity rule in $\mathbf{G}_{\mathrm{ext}}[\beta]$, we will follow a proof strategy which is structurally similar, although more complex, to the one we followed in Sect. 3 to show τ-eliminability for $\mathbf{G}[\beta]$.

First of all, we will prove that the statements obtained from those of Lemmas 3.3 and 3.6 and Theorems 3.4 (Generalized β-inversion), 3.7 (Left τ-elimination) and 3.9 (Generalized β-introduction) *by replacing throughout* $\mathbf{G}[\beta]$ *with* $\mathbf{G}_{\mathrm{ext}}[\beta]$ (and to which we shall refer from now on as to Lemmas 3.3* and 3.6* and

Theorems 3.4*, 3.7* and 3.9*) still do hold. In fact, as one can readily see, it will be sufficient to prove Lemmas 3.3* and 3.6*, and this by taking care of making the necessary integrations to the proofs of the corresponding Lemmas of Sect. 3.

*Proof of Lemma 3.3** The proof of Lemma 3.3 has to be supplemented as follows. First of all, in discussing subcase E.2 we must also consider the possibility that R' is [*Ext*]. So:
[E.2.c] $M \equiv \ulcorner(\lambda x.P)Q\urcorner$ with $|P| \equiv p$, $|Q| \equiv q$, and $t \equiv (\lambda x.p)q$. Moreover, $M^* \equiv P^*[x/Q^*]$ and $\mathfrak{D}$ has the form

$$\dfrac{\dfrac{\mathfrak{D}'_1\left\{\begin{array}{c}\vdots\\(\lambda x.p)u = s_1 u\end{array}\right.}{\lambda x.p = s_1}\,Ext \qquad \left.\begin{array}{c}\vdots\\ q = s_2\end{array}\right\}\mathfrak{D}_2}{(\lambda x.p)q = s_1 s_2}\,App \qquad \text{where } u \notin \mathrm{FV}(\lambda x.p, s_1).$$

Let $N := \ulcorner(\lambda x.P)u\urcorner$. Then $|N| \equiv (\lambda x.p)u$ and $N^* \equiv P^*[x/u]$.

We have $\mathtt{s}(\mathfrak{D}'_1) < \mathtt{s}(\mathfrak{D})$ and $\mathtt{s}(\mathfrak{D}_2) \le \mathtt{s}(\mathfrak{D})$, $\|q\| < \|(\lambda x.p)q\|$. So by the I.H. and the *Parallel substitution* Lemma 2.8 we can take

$$\mathsf{red}_M(\mathfrak{D}) := \dfrac{\mathsf{red}_N(\mathfrak{D}'_1)\left\{\begin{array}{c}\vdots\\ P^*[x/u] = s_1 u\end{array}\right. \quad \left.\begin{array}{c}\vdots\\ Q^* = s_2\end{array}\right\}\mathsf{red}_Q(\mathfrak{D}_2)}{P^*[x/u][u/Q^*] = (s_1 u)[u/s_2]}\,\mathrm{SUB}\,.$$

By the condition on u and Fact 3.2, $\mathsf{red}_M(\mathfrak{D}) \vdash^- P^*[x/Q^*] = s_1 s_2$.

Next, a totally new case has to be considered:

Case F: $R = [Ext]$. Then, for some variable $x \notin \mathrm{FV}(t,s)$,

$$\mathfrak{D} \equiv \dfrac{\mathfrak{D}_1\left\{\begin{array}{c}\vdots\\ tx = sx\end{array}\right.}{t = s}\,Ext.$$

Let $N := Mx$. Then $|N| \equiv tx$ and $N^* \equiv M^*x$; also, by Fact 3.2, $x \notin \mathrm{FV}(M^*)$.

Since $\mathtt{s}(\mathfrak{D}_1) < \mathtt{s}(\mathfrak{D})$, we may apply the I.H. to $\mathfrak{D}_1$, and set:

$$\mathsf{red}_M(\mathfrak{D}) := \dfrac{\mathsf{red}_N(\mathfrak{D}_1)\left\{\begin{array}{c}\vdots\\ M^*x = sx\end{array}\right.}{M^* = s}\,Ext\,.$$

□

*Proof of Lemma 3.6** The proof runs as that of the corresponding Lemma 3.6: since we now know that generalized β-inversion is admissible in $\mathbf{G}^-_{\mathrm{ext}}[\beta]$ (Theorem 3.4*, which is an immediate consequence of Lemma 3.3* above), only one new case has to be discussed.

Case F: $R_1 = [Ext]$ *or* $R_2 = [Ext]$. Then

$$\mathfrak{D}_1 \equiv \dfrac{\mathfrak{D}_1' \left\{ \begin{array}{c} \vdots \\ tx = sx \end{array} \right.}{t = s}\, Ext \quad \text{or} \quad \mathfrak{D}_2 \equiv \dfrac{\mathfrak{D}_2' \left\{ \begin{array}{c} \vdots \\ sx = rx \end{array} \right.}{s = r}\, Ext$$

where by Remark 2.10 we may assume, without limitations, that $x \notin \mathrm{FV}(t, s, r)$.

In the first case, we consider the derivation

$$\mathfrak{D}_2^* := \dfrac{\mathfrak{D}_2 \left\{ \begin{array}{c} \vdots \\ s = r \end{array} \right. \qquad x = x}{sx = rx}\, App$$

and observe that $\mathsf{s}(\mathfrak{D}_2^*) = \mathsf{s}(\mathfrak{D}_2)$ while $\mathsf{s}(\mathfrak{D}_1') < \mathsf{s}(\mathfrak{D}_1)$, so that by the I.H. we can take:

$$\mathsf{el_L}(\mathfrak{D}_1, \mathfrak{D}_2) := \dfrac{\mathsf{el_L}(\mathfrak{D}_1', \mathfrak{D}_2^*) \left\{ \begin{array}{c} \vdots \\ tx = rx \end{array} \right.}{t = r}\, Ext\,.$$

In the second case we proceed symmetrically (the main I.H. is used). □

To prove a Main elimination lemma for $\mathbf{G}_{\mathrm{ext}}[\beta]$ analogous to Lemma 3.10, we also need to show that $\mathbf{G}^-_{\mathrm{ext}}[\beta]$ is closed under a sort of *generalized η-rules* and their *inversions*.

Lemma 5.1

(i) *Every (left) τ-free $\mathbf{G}^-_{\mathrm{ext}}[\beta]$-derivation $\mathfrak{D} \vdash^- t = s$ can effectively be transformed, for every $x \notin FV(t)$, into a (left) τ-free $\mathbf{G}^-_{\mathrm{ext}}[\beta]$-derivation $\mathfrak{D}^* \vdash^- \lambda x.tx = s$.*

(ii) *Every (right) τ-free $\mathbf{G}^-_{\mathrm{ext}}[\beta]$-derivation $\mathfrak{D} \vdash^- \lambda x.tx = s$, with $x \notin FV(t)$, can effectively be transformed into a (right) τ-free $\mathbf{G}^-_{\mathrm{ext}}[\beta]$-derivation $\mathfrak{D}^* \vdash^- t = s$.*

Proof (i) Given $\mathfrak{D} \vdash^-_{(L)} t = s$, we take

$$\mathfrak{D}^* := \dfrac{\dfrac{\dfrac{\mathfrak{D} \left\{ \begin{array}{c} \vdots \\ t = s \end{array} \right. \qquad u = u}{tu = su}\, App}{(\lambda x.tx)u = su}\, \beta_l}{\lambda x.tx = s}\, Ext \qquad (u \text{ fresh}).$$

(ii) Given $\mathfrak{D} \vdash^{-}_{(R)} \lambda x.tx = s$, we consider

$$\mathfrak{D}' \equiv \quad \dfrac{\mathfrak{D}\left\{\begin{array}{c}\vdots\\ \lambda x.tx = s\end{array}\right. \qquad u = u}{(\lambda x.tx)u = su}\,App \qquad (u\ \mathit{fresh}).$$

By Theorem 3.4*, we have $\mathsf{inv}_\beta(\mathfrak{D}') \vdash^{-}_{(R)} tu = su$, so we may set:

$$\mathfrak{D}^* := \quad \dfrac{\mathsf{inv}_\beta(\mathfrak{D}')\left\{\begin{array}{c}\vdots\\ tu = su\end{array}\right.}{t = s}\,Ext$$

□

Theorem 5.2 (generalized η-introduction in $\mathbf{G}^{-}_{\text{ext}}[\beta]$) *The calculus* $\mathbf{G}^{-}_{\text{ext}}[\beta]$ *is closed under the rules*:

$$\frac{\Phi[\![t]\!] = r}{\Phi[\![\lambda x.tx]\!] = r}\,\eta_l^+ \qquad \frac{r = \Phi[\![t]\!]}{r = \Phi[\![\lambda x.tx]\!]}\,\eta_r^+ \qquad (x \notin FV(t))$$

where Φ is a context with one hole. Moreover, τ-free left (right) $\mathbf{G}^{-}_{\text{ext}}[\beta]$-derivability is closed under η_l^+ (η_r^+).

Proof By $\vdash^{-}_{L} t = t$ and (i) of Lemma 5.1 we get $\vdash^{-}_{L} \lambda x.tx = t$ whence, by structural rules only, a derivation:

$$\mathfrak{D} \vdash^{-}_{L} \Phi[\![\lambda x.tx]\!] = \Phi[\![t]\!].$$

Then, for any given derivation

$$\mathfrak{D}_2 \vdash^{-}_{(L)} \Phi[\![t]\!] = r$$

we have, by Lemma 3.6*,

$$\mathsf{el}_{\mathsf{L}}(\mathfrak{D}, \mathfrak{D}_2) \vdash^{-}_{(L)} \Phi[\![\lambda x.tx]\!] = r\,.$$

The rest of the claim follows by using the dual transformation $\tilde{\ }$. □

In order to prove an η-inversion lemma corresponding to Theorem 3.4, we have to introduce η-*marked* λ-terms, i.e. (in analogy to β-marked terms) expressions $A, B, \ldots$ taken from the set $\Lambda'' \supseteq \Lambda$, which is defined inductively as follows over the extended alphabet $\mathcal{A} \cup \{\rceil, \lceil\}$:

- $V \subseteq \Lambda''$,
- $A \in \Lambda'', x \in V \Rightarrow (\lambda x.A) \in \Lambda''$,
- $A, B \in \Lambda'' \Rightarrow (AB) \in \Lambda''$,
- $A \in \Lambda'', x \notin \mathrm{FV}(A) \Rightarrow (\lceil \lambda x.Ax \rceil) \in \Lambda''$.

Again, the usual conventions concerning free and bound variables, α-congruence and substitution shall apply also to Λ''. Note that η-marked terms of the form $\lceil \lambda x.Ax \rceil$ *are not* abstraction terms.

Given an η-marked term A, we denote by $|A|$ the unmarked term $\in \Lambda$ which is obtained from A by leaving out every occurrence of the markers '$\lceil$' and '$\rceil$'. In case $t \in \Lambda$ and $|A| \equiv t$, we also say that A is *an η-marking of t*. Finally, given an η-marked term A, the term $A^* \in \Lambda$ is defined inductively as expected:

- $x^* := x$, for $x \in V$,
- $(\lambda x.A)^* := \lambda x.A^*$,
- $(AB)^* := A^* B^*$,
- $(\lceil \lambda x.Ax \rceil)^* := A^*$.

The following is easily verified by induction on the construction of η-marked terms:

Fact 5.3 *For every $A, B \in \Lambda''$ and $x \in V$:* $(A[x/B])^* \equiv A^*[x/B^*]$.

Also, we clearly have

Fact 5.4 *For every $A \in \Lambda''$:* $FV(A) = FV(|A|) = FV(A^*)$.

Lemma 5.5 *Every τ-free $\mathbf{G}_{\text{ext}}[\beta]$-derivation $\mathfrak{D} \vdash^- t = s$ can effectively be transformed, given an arbitrary η-marking A of t, into a τ-free $\mathbf{G}_{\text{ext}}[\beta]$-derivation*

$$\mathsf{red}_A(\mathfrak{D}) \vdash^- A^* = s$$

which, moreover, is a right *derivation provided $\mathfrak{D}$ is a* right *derivation.*

Proof We argue by main induction on $\mathtt{s}(\mathfrak{D})$ and secondary induction on $\|t\|$, taking cases according to the last inference R of $\mathfrak{D}$.

Case A: $R = [\varrho]$. Trivially, $\mathsf{red}_A(\mathfrak{D}) := \mathfrak{D}$.
Case B: $R = [\xi]$. Then $t \equiv \lambda x.t'$, $s \equiv \lambda x.s'$ and

$$\mathfrak{D} \equiv \quad \cfrac{\mathfrak{D}_1 \left\{ \begin{array}{c} \vdots \\ t' = s' \end{array} \right.}{\lambda x.t' = \lambda x.s'} \, \xi \, ,$$

where $\mathtt{s}(\mathfrak{D}_1) = \mathtt{s}(\mathfrak{D})$ and $\left\|t'\right\| < \|t\|$.
Subcase B.1: $A \equiv \lambda x.B$ with $|B| \equiv t'$ (and so $A^* \equiv \lambda x.B^*$). By applying the I.H., we set:

$$\mathsf{red}_A(\mathfrak{D}) := \quad \cfrac{\mathsf{red}_B(\mathfrak{D}_1) \left\{ \begin{array}{c} \vdots \\ B^* = s' \end{array} \right.}{\lambda x.B^* = \lambda x.s'} \, \xi \, .$$

Subcase B.2: $A \equiv \lceil \lambda x.Cx \rceil$ with $|C| \equiv t''$, $t' \equiv t''x$, $x \notin \mathrm{FV}(t'')$. Also, $A^* \equiv C^*$. By applying the I.H. and (ii) of Lemma 5.1 (together with Fact 5.4), we may set:

$$\mathsf{red}_A(\mathfrak{D}) := \cfrac{\cfrac{\mathsf{red}_{Cx}(\mathfrak{D}_1)\left\{\begin{array}{c}\vdots\\ C^*x = s'\end{array}\right.}{\lambda x.C^*x = \lambda x.s'}\,\xi}{C^* = \lambda x.s'}\,5.1.\mathrm{(ii)}$$

Case C: $R = [App]$. Then $t \equiv t_1t_2$, $s \equiv s_1s_2$, $A \equiv A_1A_2$ with $|A_i| \equiv t_i$. Also, $A^* \equiv A_1^*A_2^*$. By applying the (secondary) I.H., we set

$$\mathsf{red}_A(\mathfrak{D}) := \cfrac{\mathsf{red}_{A_1}(\mathfrak{D}_1)\left\{\begin{array}{c}\vdots\\ A_1^* = s_1\end{array}\right. \qquad \left.\begin{array}{c}\vdots\\ A_2^* = s_2\end{array}\right\}\mathsf{red}_{A_2}(\mathfrak{D}_2)}{A_1^*A_2^* = s_1s_2}\,App\,,$$

where $\mathfrak{D}_1$ and $\mathfrak{D}_2$ are the subderivations of $\mathfrak{D}$ ending with $t_1 = s_1$, respectively $t_2 = s_2$.
Case D: $R = [Ext]$. Then

$$\mathfrak{D} \equiv \cfrac{\mathfrak{D}_1\left\{\begin{array}{c}\vdots\\ tu = su\end{array}\right.}{t = s}\,Ext \quad \text{where } u \notin \mathrm{FV}(t,s) \text{ and } \mathtt{s}(\mathfrak{D}_1) < \mathtt{s}(\mathfrak{D})\,.$$

Since $|A| \equiv t$, we have $|Au| \equiv tu$, so by the I.H. and Fact 5.4 we may take:

$$\mathsf{red}_A(\mathfrak{D}) := \cfrac{\mathsf{red}_{Au}(\mathfrak{D}_1)\left\{\begin{array}{c}\vdots\\ A^*u = su\end{array}\right.}{A^* = s}\,Ext\,.$$

Case E: $R = [\beta_r]$. Immediate by the I.H.
Case F: $R = [\beta_l]$. Then $t \equiv (\lambda x.p)q_0 \cdots q_n$ $(n \geq 0)$ and:
Subcase F.1: $A \equiv (\lambda x.B)C_0 \cdots C_n$ with $|B| \equiv p$, $|C_i| \equiv q_i$ for $0 \leq i \leq n$, and

$$\mathfrak{D} \equiv \cfrac{\mathfrak{D}_1\left\{\begin{array}{c}\vdots\\ p[x/q_0]q_1 \cdots q_n = s\end{array}\right.}{(\lambda x.p)q_0 \cdots q_n = s}\,\beta_l \quad \text{where } \mathtt{s}(\mathfrak{D}_1) < \mathtt{s}(\mathfrak{D}).$$

Since $(B[x/C_0]C_1 \cdots C_n)^* \equiv B^*[x/C_0^*]C_1^* \cdots C_n^*$ by Fact 5.3, by the I.H. we may take:

$$\mathsf{red}_A(\mathfrak{D}) := \frac{\mathsf{red}_{B[x/C_0]C_1\cdots C_n}(\mathfrak{D}_1)\left\{\begin{array}{c}\vdots\\ B^*[x/C_0^*]C_1^* \cdots C_n^* = s\end{array}\right.}{(\lambda x.B^*)C_0^* \cdots C_n^* = s}\,\beta_l$$

Subcase F.2: $A \equiv \ulcorner \lambda x.Dx \urcorner C_0 \cdots C_n$ with $|D| \equiv p'$, $|C_i| \equiv q_i$ $(0 \leq i \leq n)$, $p \equiv p'x$ and $x \notin \mathrm{FV}(p')$. Also, $A^* \equiv D^* C_0^* \cdots C_n^*$ and

$$\mathfrak{D} \equiv \frac{\mathfrak{D}_1\left\{\begin{array}{c}\vdots\\ (p'x)[x/q_0]q_1 \cdots q_n = s\end{array}\right.}{(\lambda x.p'x)q_0 \cdots q_n = s}\,\beta_l$$
$$\text{where } \mathrm{s}(\mathfrak{D}_1) < \mathrm{s}(\mathfrak{D}).$$

By the I.H. we may take:

$$\mathsf{red}_A(\mathfrak{D}) := \quad \mathsf{red}_{(Dx)[x/C_0]C_1\cdots C_n}(\mathfrak{D}_1)$$

since $((Dx)[x/C_0]C_1 \cdots C_n)^* \equiv (D^*x)[x/C_0^*]C_1^* \cdots C_n^* \equiv D^*C_0^* \cdots C_n^* \equiv A^*$.

By inspecting the whole proof, it is easily verified that the transformation red_A maps *right* derivations into *right* derivations (whereas left derivations *need not* be transformed into left derivations: consider subcase B.2 where we use (ii) of Lemma 5.1). □

As an immediate consequence, we have

Theorem 5.6 (generalized η-inversion in $\mathbf{G}^-_{\mathrm{ext}}[\beta]$) *Every τ-free $\mathbf{G}_{\mathrm{ext}}[\beta]$-derivation*

$$\mathfrak{D} \vdash^- \Phi[\![\lambda x.tx]\!] = r \quad (\Phi \textit{ a context with one hole; } x \notin FV(t))$$

can effectively be transformed into a τ-free $\mathbf{G}_{\mathrm{ext}}[\beta]$-derivation

$$\mathsf{inv}_\eta(\mathfrak{D}) \vdash^- \Phi[\![t]\!] = r$$

which, moreover, is a right *derivation provided $\mathfrak{D}$ is a* right *derivation.*

We can finally prove, along the lines of the proof of Lemma 3.10:

Lemma 5.7 (main elimination lemma for $\mathbf{G}_{\mathrm{ext}}[\beta]$) *To each pair*

$$\mathfrak{D}_1 \vdash^- p = q \quad \textit{and} \quad \mathfrak{D}_2 \vdash^- \Phi[\![q]\!] = r$$

of τ*-free* $\mathbf{G}_{\text{ext}}[\beta]$*-derivations we can effectively associate a* τ*-free* $\mathbf{G}_{\text{ext}}[\beta]$*-derivation*

$$\mathsf{el}_{\Phi}(\mathfrak{D}_1, \mathfrak{D}_2) \vdash^{-} \Phi[\![p]\!] = r.$$

Proof The proof runs as that of the corresponding Lemma 3.10 (using, where appropriate, Theorems 3.4* and 3.9* in place of Theorems 3.4 and 3.9), except that the following new case has to be considered:
Case F: $R = [Ext]$. Then $\mathfrak{D}_1$ has the form

$$\frac{\mathfrak{D}'_1 \left\{ \begin{array}{c} \vdots \\ px = qx \end{array} \right.}{p = q} \, Ext, \quad \text{where } x \notin \mathrm{FV}(p, q) \text{ and } \mathsf{s}(\mathfrak{D}'_1) < \mathsf{s}(\mathfrak{D}_1).$$

Let

$$\mathfrak{D}''_1 := \quad \frac{\mathfrak{D}'_1 \left\{ \begin{array}{c} \vdots \\ px = qx \end{array} \right.}{\lambda x.px = \lambda x.qx} \, \xi$$

and, by applying Theorem 5.2 to $\mathfrak{D}_2$, let

$$\mathfrak{D}'_2 \vdash^{-} \Phi[\![\lambda x.qx]\!] = r\,.$$

Since $\mathsf{s}(\mathfrak{D}''_1) < \mathsf{s}(\mathfrak{D}_1)$, by the I.H. we have

$$\mathsf{el}_{\Phi}(\mathfrak{D}''_1, \mathfrak{D}'_2) \vdash^{-} \Phi[\![\lambda x.px]\!] = r$$

whence, by Theorem 5.6, we may take:

$$\mathsf{el}_{\Phi}(\mathfrak{D}_1, \mathfrak{D}_2) := \mathsf{inv}_{\eta}(\mathsf{el}_{\Phi}(\mathfrak{D}''_1, \mathfrak{D}'_2))\,.$$

□

Theorem 5.8 (τ-elimination for $\mathbf{G}_{\text{ext}}[\beta]$) *Every derivation* $\mathfrak{D}$ *in* $\mathbf{G}_{\text{ext}}[\beta]$ *can effectively be transformed into a* τ-free $\mathbf{G}_{\text{ext}}[\beta]$*-derivation* $\mathsf{e}(\mathfrak{D})$ *having the same end–equation as* $\mathfrak{D}$.

Proof By induction on the number of occurrences of rule $[\tau]$ in $\mathfrak{D}$, using the *Main elimination lemma* 5.7 with $\Phi \equiv *$ to eliminate a topmost occurrence of $[\tau]$. □

Remark 5.9 Using Theorem 5.8 it is now possible to solve in the *positive* the problem left open in [26] (Problem 6.1) concerning the eliminability of the transitivity rule for the analytic version $\mathbf{G}_{\text{ext}}[\mathcal{C}]$ of extensional combinatory logic $\mathbf{CL} + ext$. The proof, which is rather complex, cannot be presented here.

We just stress that the proof is *indirect*, hinging on the *detour* through $\mathbf{G}_{\text{ext}}[\beta]$, and that up to now we have not been able to find a direct one.

6 Transitivity elimination in $\mathbf{G}_{\text{ext}}[\beta]$ and $\beta\eta$-reduction

Throughout this section, derivability ($\vdash$) means derivability in $\mathbf{G}_{\text{ext}}[\beta]$, while the symbols $\to$ and $\twoheadrightarrow$ without subscripts refer always to the usual relations on Λ of *one-step $\beta\eta$-reduction* and *$\beta\eta$-reduction*, respectively. Next, we use the symbol $\xrightarrow{l}$ to denote the relation of one-step *leftmost* $\beta\eta$-reduction ($t \xrightarrow{l} s$ iff s is obtained by replacing in t the leftmost (β- or η-) redex occurrence with the corresponding contractum) and the symbol $\xrightarrow{l}\!\!\!\!\to$ to denote its reflexive and transitive closure (*leftmost* $\beta\eta$-reduction). Note that it is possible to have terms like e.g. $z((\lambda x.ux)y)$ in which a β-redex and an η-redex occurrence share the leftmost λ; the result of both contractions is however the same, and so we fix conventionally that in these cases the β-redex is the leftmost one. $\mathrm{NF}_{\beta\eta}$ denotes the set of all terms in *$\beta\eta$-normal form*; a term t *has* a $\beta\eta$-normal form provided $t \twoheadrightarrow s$ for some $s \in \mathrm{NF}_{\beta\eta}$.

As we did in Sect. 4 for β-reduction, we are now going to apply the *Full* and the *Left τ-Elimination theorems* 5.8 and 3.7* for $\mathbf{G}_{\text{ext}}[\beta]$ to the study of $\beta\eta$-reduction. In particular, we will present new purely proof-theoretical demonstrations of the *Church–Rosser* Theorem on the one side, and of the *Normalization* or *Leftmost reduction* Theorem [21,29] as well as of the so-called *η-postponement* Theorem [3,14,29], on the other side. Notice that whereas for β-reduction the Normalization theorem is an immediate corollary of the Standardization theorem (see Sect. 4), the same doesn't hold for $\beta\eta$-reduction, since e.g. there are standard reductions $\sigma : t \twoheadrightarrow_{s} s$ with $s \in \mathrm{NF}_{\beta\eta}$ (under an appropriate definition of "standard") which are not leftmost; see the discussion in [21], Ch. IV. A nice proof of the Standardization theorem for $\beta\eta$-reduction has been recently given by Kashima [20].

Preliminarily, we collect below some basic facts concerning β-, η- and $\beta\eta$-reduction, as well as leftmost $\beta\eta$-reduction, which we are going to use later. To this aim, it is convenient to introduce the abbreviation

$$\Lambda_x := \{qx \mid q \in \Lambda \quad \text{and} \quad x \notin \mathrm{FV}(q)\} \quad (\text{for } x \in V).$$

As a motivation for (ii) below, observe that whereas for leftmost β-reduction we can infer $\lambda x.t \xrightarrow{l}\!\!\!\!\to \lambda x.s$ from $t \xrightarrow{l}\!\!\!\!\to s$, this is no longer true for leftmost $\beta\eta$-reduction; for instance,

$$(\lambda u.u)vx \xrightarrow{l}\!\!\!\!\to vx$$

but

$$\lambda x.((\lambda u.u)vx) \xrightarrow{l} (\lambda u.u)v \xrightarrow{l} v\,,$$

and so $\lambda x.((\lambda u.u)vx) \xrightarrow{l}\!\!\!\!\to \lambda x.vx$ is false.

Lemma 6.1

(i) *Let the λ-terms* $t_0, \ldots, t_n$, $s_0, \ldots, s_n$ $(n \geq 0)$ *be such that* $t_i \overset{l}{\twoheadrightarrow} s_i$ *and* $s_i \in \mathrm{NF}_{\beta\eta}$ *for* $0 \leq i \leq n$. *Then, for every variable x,*

$$xt_0 \cdots t_n \overset{l}{\twoheadrightarrow} xs_0 \cdots s_n\,.$$

(ii) *Assume that* $t \overset{*}{\twoheadrightarrow} r$, *where* $\overset{*}{\twoheadrightarrow} \in \{\twoheadrightarrow, \overset{l}{\twoheadrightarrow}\}$. *Then, exactly one of the following cases does hold*:
(a) $t \equiv t'x \in \Lambda_x$ *and* $r \equiv r'x \in \Lambda_x$ *and* $\lambda x.t \overset{*}{\twoheadrightarrow} t' \overset{*}{\twoheadrightarrow} r'$,
(b) $t \equiv t'x \in \Lambda_x$ *and* $r \notin \Lambda_x$ *and* $\lambda x.t \overset{*}{\twoheadrightarrow} t' \overset{*}{\twoheadrightarrow} \lambda x.r$,
(c) $t \notin \Lambda_x$ *and* $r \equiv r'x \in \Lambda_x$ *and* $\lambda x.t \overset{*}{\twoheadrightarrow} r'$,
(d) $t \notin \Lambda_x$ *and* $r \notin \Lambda_x$ *and* $\lambda x.t \overset{*}{\twoheadrightarrow} \lambda x.r$.

(iii) *If* $tx \twoheadrightarrow_\beta r$ *and* $x \notin FV(t)$ *then*:
(a) *if* $r \equiv r'x \in \Lambda_x$ *then* $t \twoheadrightarrow_\beta r'$ *or* $t \twoheadrightarrow_\beta \lambda x.r$,
(b) *if* $r \notin \Lambda_x$ *then* $t \twoheadrightarrow_\beta \lambda x.r$.

(iv) *If* $tx \twoheadrightarrow_\eta s$ *and* $x \notin FV(t)$ *then* $s \equiv s'x \in \Lambda_x$ *and* $t \twoheadrightarrow_\eta s'$.

Proof (i) and (iv): straightforward.
(ii): we assume that $\overset{*}{\twoheadrightarrow}$ is $\overset{l}{\twoheadrightarrow}$ (a similar argument works for $\twoheadrightarrow$), arguing by induction on the number n of steps of a given leftmost reduction $\sigma : t \overset{l}{\twoheadrightarrow} r$. The conclusion is obvious if $n = 0$; so let $n = k + 1$. Then, for some term p,

$$(1)\ t \overset{l}{\to} p \quad \text{and} \quad (2)\ p \overset{l}{\twoheadrightarrow} r \ \textit{in k steps}.$$

If $t \equiv t'x \in \Lambda_x$ and $t'x$ is not a redex then, by (1), $p \equiv p'x$ and $\lambda x.t \overset{l}{\to}_\eta t' \overset{l}{\to} p'$, whence also $x \notin \mathrm{FV}(p')$ and $p \in \Lambda_x$. So we can apply the I.H. to (2), and conclude that either (a) or (b) holds.

If $t \equiv t'x \in \Lambda_x$ and $t'x$ is a redex, then the latter is necessarily a β-redex: so $t' \equiv \lambda u.t''$ and by (1) $p \equiv t''[u/x]$, whence

$$\lambda x.t \overset{l}{\to}_\eta t' \equiv \lambda u.t'' \equiv_\alpha \lambda x.t''[u/x] \equiv \lambda x.p\,.$$

And the conclusion (a) or (b) follows by applying the I.H. to (2).

Finally, if $t \notin \Lambda_x$ then $\lambda x.t \overset{l}{\to} \lambda x.p$, and by applying the I.H. to (2) we have that either (c) or (d) holds.
(iii): the easy verification, similar to that of (ii), is left to the reader. Note the difference between point (a) here and point (a) of (ii). □

In perfect analogy with Lemmas 4.1 and 4.2, we can now establish the following links between $\beta\eta$-reductions and $\mathbf{G}_{\mathrm{ext}}[\beta]$-derivations:

Lemma 6.2

(i) *If* $t \to s$ *then we can find a* τ*-free* left *derivation* $\mathfrak{D}_{t,s} \vdash_L^- t = s$.
(ii) *To each reduction path* $\sigma : t \twoheadrightarrow s$ *we can effectively associate a* left *derivation* $\mathsf{lder}(\sigma) \vdash_L t = s$.

(iii) *To each* τ*-free derivation* $\mathfrak{D} \vdash^{-} t = s$ *we can associate a term* $r_{\mathfrak{D}}$ *such that* $t \twoheadrightarrow r_{\mathfrak{D}} \twoheadleftarrow s$.

Proof (i): by induction on the generation of $\rightarrow$, using Theorem 5.2 to construct a derivation $\mathfrak{D}^{t,x} \vdash_L^{-} \lambda x.tx = t$ for any given t and $x \notin \mathrm{FV}(t)$.
(ii): follows from (i) and the transitivity rule.
(iii): is easily verified by induction on the height of the given τ*-free* derivation $\mathfrak{D} \vdash^{-} t = s$, taking cases according to the last inference R. (ii) of Lemma 6.1 shows which term one has to take for $r_{\mathfrak{D}}$ when $R = [Ext]$. □

Theorem 6.3 **(CR($\beta\eta$))** $\twoheadrightarrow$ *is Church–Rosser.*

Proof Immediate, like that of the corresponding Theorem 4.3, using the Elimination Theorem 5.8 and (ii), (iii) of Lemma 6.2. □

Let us now turn to applications of *Left* τ-eliminability.

Definition 6.4 *A derivation* $\mathfrak{D}$ *in* $\mathbf{G}_{\mathrm{ext}}[\beta]$ *is* normal *provided*:

(i) $\mathfrak{D}$ *is* τ*-free;*
(ii) *the* left *premise of each occurrence of an* $[App]$*-inference in* $\mathfrak{D}$ *is the conclusion of a* structural *rule* (*i.e.* $[\varrho]$, $[\xi]$, $[App]$).

Lemma 6.5 *To each* left τ-free *derivation* $\mathfrak{D} \vdash_L^{-} t = s$ *we can effectively associate a* left normal *derivation*

$$\mathsf{nor}(\mathfrak{D}) \vdash_L^{-} t = s$$

satisfying $\mathtt{s}(\mathsf{nor}(\mathfrak{D})) \leq \mathtt{s}(\mathfrak{D})$.

Proof By induction on $\omega \cdot \mathtt{s}(\mathfrak{D}) + \mathtt{h}(\mathfrak{D})$, taking cases according to the final inference R of $\mathfrak{D}$.
The conclusion is trivial if $R = [\varrho]$, while it follows easily by the I.H. if $R = [\xi], [\beta_l], [Ext]$: simply, we normalize the subderivation of the premise of R, and then apply R again.

If $R = [App]$, we consider the *leftmost* branch π of $\mathfrak{D}$, letting $t' = s'$ be the lowermost node in π which is the conclusion of an inference rule R' *different from* $[App]$. Such a node, which of course necessarily exists, is located strictly above the conclusion $t = s$ of $\mathfrak{D}$. We so have $t \equiv t' t_0 \cdots t_n$, $s \equiv s' s_0 \cdots s_n$ $(n \geq 0)$, and:

$$\mathfrak{D} \equiv \dfrac{\dfrac{\mathfrak{D}'\left\{\dfrac{\vdots\ \}\mathfrak{D}''}{t' = s'}\,R' \right. \qquad \left.\begin{matrix}\vdots \\ t_0 = s_0\end{matrix}\right\}\mathfrak{D}_0}{t' t_0 = s' s_0}\,App \quad \begin{matrix} \ddots \\ t' t_0 \cdots t_{n-1} = s' s_0 \cdots s_{n-1} \end{matrix} \qquad \left.\begin{matrix}\vdots \\ t_n = s_n\end{matrix}\right\}\mathfrak{D}_n}{t' t_0 \cdots t_n = s' s_0 \cdots s_n}\,R{=}App\ .$$

We now consider all the possible subcases depending on R':

- $R' = [\varrho]$ or $[\xi]$. By applying the I.H. we may replace in $\mathfrak{D}$ the subderivations $\mathfrak{D}_0, \ldots, \mathfrak{D}_n$ by $\mathsf{nor}(\mathfrak{D}_0), \ldots, \mathsf{nor}(\mathfrak{D}_n)$, and also $\mathfrak{D}''$ by $\mathsf{nor}(\mathfrak{D}'')$ if $R' = [\xi]$: the resulting derivation satisfies the conclusion.
- $R' = [\beta_l]$. Then $t' \equiv (\lambda y.p_0)p_1 \cdots p_k$ $(k \geq 1)$ and the premise of R', i.e. the conclusion of $\mathfrak{D}''$, is $p_0[y/p_1]p_2 \cdots p_k = s'$. By replacing in the original derivation $\mathfrak{D}$ the subderivation $\mathfrak{D}'$ with $\mathfrak{D}''$, we get a derivation

$$\mathfrak{D}^* \vdash_L^- p_0[y/p_1]p_2 \cdots p_k t_0 \cdots t_n = s's_0 \cdots s_n .$$

Now $\mathsf{s}(\mathfrak{D}^*) < \mathsf{s}(\mathfrak{D})$, so by applying the (main) I.H. we can take

$$\mathsf{nor}(\mathfrak{D}) := \cfrac{\mathsf{nor}(\mathfrak{D}^*)\left\{\begin{array}{c} \vdots \\ p_0[y/p_1]p_2 \cdots p_k t_0 \cdots t_n = s's_0 \cdots s_n \end{array}\right.}{t't_0 \cdots t_n = s's_0 \cdots s_n}\ \beta_l .$$

- $R' = [Ext]$. Then the premise of R', i.e. the conclusion of $\mathfrak{D}''$, is $t'u = s'u$ for some $u \notin \mathrm{FV}(t', s')$. The fact that $\mathfrak{D}''$ is a *left* τ-free derivation and that the variable u has *exactly one* occurrence in $s'u$ allows us to apply the special *substitution* Lemma 2.11, giving a derivation

$$\mathsf{sb}_u(\mathfrak{D}'', \mathfrak{D}_0) \vdash_L^- t't_0 = s's_0$$

with $\mathsf{s}(\mathsf{sb}_u(\mathfrak{D}'', \mathfrak{D}_0)) \leq \mathsf{s}(\mathfrak{D}'') + \mathsf{s}(\mathfrak{D}_0)$.
Hence, by replacing in the original derivation $\mathfrak{D}$ the subderivation ending with $t't_0 = s's_0$ by the derivation $\mathsf{sb}_u(\mathfrak{D}'', \mathfrak{D}_0)$, we get a derivation

$$\mathfrak{D}^* \vdash_L^- t't_0 \cdots t_n = s's_0 \cdots s_n$$

such that

$$\mathsf{s}(\mathfrak{D}^*) = \mathsf{s}(\mathsf{sb}_u(\mathfrak{D}'', \mathfrak{D}_0)) + \sum_{i=1}^{n} \mathsf{s}(\mathfrak{D}_i) \; < \; \mathsf{s}(\mathfrak{D}'') + 1 + \mathsf{s}(\mathfrak{D}_0) + \sum_{i=1}^{n} \mathsf{s}(\mathfrak{D}_i) = \mathsf{s}(\mathfrak{D}).$$

And the conclusion follows by the I.H. □

Remark 6.6 With the above Lemma (note that the proof, because of the use of Lemma 2.11, depends *essentially* on the assumption that $\mathfrak{D}$ be a *left* τ-free derivation) we have confined ourselves to state *just* what we are going to need for the applications we have in mind. Indeed, it is possible to establish—at the

cost, however, of a quite lengthy and complex proof—the following general result:
Every τ-free derivation $\mathfrak{D}$ in $\mathbf{G}_{\text{ext}}[\beta]$ can effectively be transformed into a normal *derivation* $\mathsf{nor}(\mathfrak{D})$ *with* $\mathtt{s}(\mathsf{nor}(\mathfrak{D})) \leq \mathtt{s}(\mathfrak{D})$, *and with the additional property that no β-inference is immediately preceded by an* $[Ext]$*-inference.*

The following key result shows that a *left normal* derivation of $t = s$ "encodes" a reduction path from t to s consisting of a number of β-contractions followed by a number of η-contractions (the dividing term being in NF_β whenever $s \in \mathrm{NF}_{\beta\eta}$), as well as a leftmost reduction path from t to s in case $s \in \mathrm{NF}_{\beta\eta}$.

Theorem 6.7 (Extraction) *From any given* left normal *derivation* $\mathfrak{D} \vdash_L^- t = s$ *we can effectively extract*:

(i) *a term* $r_\mathfrak{D}$ *such that* $t \twoheadrightarrow_\beta r_\mathfrak{D} \twoheadrightarrow_\eta s$ *and* $s \in \mathrm{NF}_{\beta\eta} \Rightarrow r_\mathfrak{D} \in \mathrm{NF}_\beta$;
(ii) *a* leftmost *reduction path* $t \overset{l}{\twoheadrightarrow} s$, provided $s \in \mathrm{NF}_{\beta\eta}$.

Proof We argue by induction on $\omega^2 \cdot \#_\lambda(s) + \omega \cdot \mathtt{s}(\mathfrak{D}) + \mathtt{h}(\mathfrak{D})$, where $\#_\lambda(r)$ denotes the number of occurrences of the symbol 'λ' in the term r.
Let R be the final inference in $\mathfrak{D}$; since the latter is by assumption a *left normal* derivation, the cases to be considered are (apart from the trivial $R = [\varrho]$) the following four.
Case A: $R = [Ext]$. Then

$$\mathfrak{D} \equiv \qquad \cfrac{\mathfrak{D}'\left\{\begin{array}{c}\vdots\\ tu = su\end{array}\right.}{t = s}\,{\scriptstyle Ext}\,, \quad \text{with } u \notin \mathrm{FV}(t, s).$$

<u>Subcase A.1:</u> either $s \notin \mathrm{NF}_{\beta\eta}$, or $s \in \mathrm{NF}_{\beta\eta}$ *and* s is not an abstraction term. Then

$$s \in \mathrm{NF}_{\beta\eta} \Rightarrow su \in \mathrm{NF}_{\beta\eta}\,, \qquad (*)$$

and since $\#_\lambda(su) = \#_\lambda(s)$, we can apply the (secondary) I.H. to $\mathfrak{D}'$, giving
(a) a term $r' \equiv r_{\mathfrak{D}'}$ s.t. $tu \twoheadrightarrow_\beta r' \twoheadrightarrow_\eta su$ and $su \in \mathrm{NF}_{\beta\eta} \Rightarrow r' \in \mathrm{NF}_\beta$;
(b) a leftmost reduction $tu \overset{l}{\twoheadrightarrow} su$, provided $su \in \mathrm{NF}_{\beta\eta}$.
(i) If $r' \notin \Lambda_u$ then by the first half of (a) and (iii) of Lemma 6.1

$$t \twoheadrightarrow_\beta \lambda u.r' \twoheadrightarrow_\eta \lambda u.su \rightarrow_\eta s\,,$$

and we may take $r_\mathfrak{D} := \lambda u.r'$ since by $(*)$ and the second half of (a)

$$s \in \mathrm{NF}_{\beta\eta} \Rightarrow su \in \mathrm{NF}_{\beta\eta} \Rightarrow r' \in \mathrm{NF}_\beta \Rightarrow \lambda u.r' \in \mathrm{NF}_\beta\,.$$

If $r' \equiv r''u \in \Lambda_u$ then by the first half of (a) and (iii), (iv) of Lemma 6.1

$$t \twoheadrightarrow_\beta r'' \twoheadrightarrow_\eta s \quad \text{or} \quad t \twoheadrightarrow_\beta \lambda u.r''u \rightarrow_\eta r'' \twoheadrightarrow_\eta s\,,$$

and we may take $r_{\mathfrak{D}} := r''$ in the first case, $r_{\mathfrak{D}} := \lambda u.r'$ in the second one. Indeed, as above, we have $s \in \mathrm{NF}_{\beta\eta} \Rightarrow r_{\mathfrak{D}} \in \mathrm{NF}_{\beta}$ in both cases.
(ii) if $s \in \mathrm{NF}_{\beta\eta}$ then $t \overset{l}{\twoheadrightarrow} s$ follows by $(*)$, (b) and (ii) of Lemma 6.1.
Subcase A.2: $s \in \mathrm{NF}_{\beta\eta}$ and s is an abstraction term, say $s \equiv \lambda y.s'$. Then by dualization (cf. 2.3), Theorem 3.4* and Lemma 6.5 we may transform $\mathfrak{D}'$ into a *left normal* derivation

$$\mathfrak{D}^* \vdash_L^- tu = s'[y/u]\,, \quad \text{where } \mathfrak{D}^* \equiv \widetilde{\mathsf{nor}(\mathsf{inv}_\beta(\widetilde{\mathfrak{D}'}))}\,.$$

By the assumptions on $s \equiv \lambda y.s'$, we have

$$s', s'[y/u] \in \mathrm{NF}_{\beta\eta} \quad \text{and} \quad s'[y/u] \notin \Lambda_u\,. \tag{$**$}$$

Indeed, were $s'[y/u] \in \Lambda_u$ then we would have (in view of $u \notin \mathrm{FV}(s)$) $s' \equiv s''y$ and $s \equiv \lambda y.s''y \notin \mathrm{NF}_{\beta\eta}$: contradiction.

Now, by $\#_\lambda\big(s'[y/u]\big) < \#_\lambda(s)$, we can apply the *main* I.H. to $\mathfrak{D}^*$ (note that, because of the use of inv_β in the construction of $\mathfrak{D}^*$, possibly $\mathrm{s}(\mathfrak{D}^*) > \mathrm{s}(\mathfrak{D})$!) and get, together with $(**)$:
(a′) a term $r' \equiv r_{\mathfrak{D}^*}$ such that $tu \twoheadrightarrow_\beta r' \twoheadrightarrow_\eta s'[y/u]$ and $r' \in \mathrm{NF}_\beta$,
(b′) a leftmost reduction $tu \overset{l}{\twoheadrightarrow} s'[y/u]$.
(i) by $r' \twoheadrightarrow_\eta s'[y/u]$, (iv) of Lemma 6.1 and $(**)$ we have: $r' \notin \Lambda_u$. Therefore, by (a′) and (iii) of Lemma 6.1,

$$t \twoheadrightarrow_\beta \lambda u.r' \twoheadrightarrow_\eta \lambda u.(s'[y/u]) \equiv_\alpha \lambda y.s' \equiv s \quad \text{and} \quad \lambda u.r' \in \mathrm{NF}_\beta\,,$$

so we may set $r_{\mathfrak{D}} := \lambda u.r'$.
(ii) by (b′), $(**)$ and (ii) of Lemma 6.1:

$$t \overset{l}{\twoheadrightarrow} \lambda u.(s'[y/u]) \equiv_\alpha \lambda y.s' \equiv s\,.$$

Case B: $R = [\xi]$. Then $t \equiv \lambda x.t'$, $s \equiv \lambda x.s'$ and so $s \in \mathrm{NF}_{\beta\eta} \Rightarrow s' \in \mathrm{NF}_{\beta\eta}$. By applying the (main) I.H. to the subderivation $\mathfrak{D}'$ of the premise $t' = s'$ of R, we find a term $r' \equiv r_{\mathfrak{D}'}$ such that

$$t' \twoheadrightarrow_\beta r' \twoheadrightarrow_\eta s' \quad \text{and} \quad s' \in \mathrm{NF}_{\beta\eta} \Rightarrow r' \in \mathrm{NF}_\beta\,,$$

and, under the assumption that $s' \in \mathrm{NF}_{\beta\eta}$, a leftmost reduction path

$$t' \overset{l}{\twoheadrightarrow} s'\,.$$

(i) clearly, we can take $r_{\mathfrak{D}} := \lambda x.r'$.
(ii) assuming $s \in \mathrm{NF}_{\beta\eta}$ we have that $s' \in \mathrm{NF}_{\beta\eta} \setminus \Lambda_x$, so we get the conclusion

$$t \equiv \lambda x.t' \overset{l}{\twoheadrightarrow} \lambda x.s' \equiv s$$

by (ii) of Lemma 6.1.

Case C: $R = [\beta_l]$. We have $t \equiv (\lambda x.t')t_0 \cdots t_n$ with $n \geq 0$ and

$$\mathfrak{D} \equiv \quad \dfrac{\mathfrak{D}' \left\{ \begin{array}{c} \vdots \\ t'[x/t_0]t_1 \cdots t_n = s \end{array} \right.}{(\lambda x.t')t_0 \cdots t_n = s} \; \beta_l \, .$$

Then both (i), with $r_{\mathfrak{D}} := r_{\mathfrak{D}'}$, and (ii) readily follow by the (secondary) I.H. applied to $\mathfrak{D}'$, together with

$$t \equiv (\lambda x.t')t_0 \cdots t_n \xrightarrow{l}_{\beta} t'[x/t_0]t_1 \cdots t_n \, .$$

Case D: $R = [App]$. Then, for some $n \geq 0$, $t \equiv t't_0 \cdots t_n$, $s \equiv s's_0 \cdots s_n$, and $\mathfrak{D}$ has the form

$$\dfrac{\dfrac{\mathfrak{D}' \left\{ \dfrac{\vdots}{t' = s'} \, R' \right. \qquad \left. \begin{array}{c} \vdots \\ t_0 = s_0 \end{array} \right\} \mathfrak{D}_0}{t't_0 = s's_0} \; App \qquad \begin{array}{c} \ddots \\ \end{array} \qquad \left. \begin{array}{c} \vdots \\ \end{array} \right.}{\dfrac{t't_0 \cdots t_{n-1} = s's_0 \cdots s_{n-1} \qquad \left. t_n = s_n \right\} \mathfrak{D}_n}{t't_0 \cdots t_n = s's_0 \cdots s_n} \; App} \, ,$$

where the inference R' is *different from* $[App]$.
Note that by *normality* of $\mathfrak{D}$ it must be either $R' = [\varrho]$ or $R' = [\xi]$, and that:

$$s \in \mathrm{NF}_{\beta\eta} \Rightarrow s', s_0, \ldots, s_n \in \mathrm{NF}_{\beta\eta} \text{ and } s' \textit{ is not } \text{an abstraction term.} \qquad (*)$$

By the (ternary, in the worst case) I.H. applied to the subderivations $\mathfrak{D}_0, \ldots, \mathfrak{D}_n$, together with $(*)$, we get for each i with $0 \leq i \leq n$:
(a) a term $r_i \equiv r_{\mathfrak{D}_i}$ such that $t_i \twoheadrightarrow_\beta r_i \twoheadrightarrow_\eta s_i$ and $s_i \in \mathrm{NF}_{\beta\eta} \Rightarrow r_i \in \mathrm{NF}_\beta$,
and, under the assumption $s \in \mathrm{NF}_{\beta\eta}$,
(b) a leftmost reduction path $t_i \overset{l}{\twoheadrightarrow} s_i$.
<u>Subcase D.1:</u> $R' = [\varrho]$, and so $t' \equiv x \equiv s'$ for some variable x.
(i): by (a) and $(*)$, the term

$$r_{\mathfrak{D}} := xr_1 \ldots r_n$$

clearly satisfies both the required conditions.
(ii): by (b) and $(*)$, together with (i) of Lemma 6.1, we may conclude, under $s \in \mathrm{NF}_{\beta\eta}$:

$$t \equiv xt_0 \cdots t_n \overset{l}{\twoheadrightarrow} xs_0 \cdots s_n \equiv s \, .$$

Subcase D.2: $R' = [\xi]$, and so $t' \equiv \lambda x.t''$, $s' \equiv \lambda x.s''$. Moreover, by (∗), it is $s \notin \mathrm{NF}_{\beta\eta}$, which means that in this case only claim (i)—actually, its first half—has to be verified. To this aim, we apply the (ternary, in the worst case) I.H. also to the subderivation $\mathfrak{D}''$ of the premise $t'' = s''$ of R', to get a term $r'' \equiv r_{\mathfrak{D}''}$ s.t.:

$$t'' \twoheadrightarrow_\beta r'' \twoheadrightarrow_\eta s'' .$$

This, combined with (a), gives

$$\begin{aligned} t \equiv (\lambda x.t'')t_0 \cdots t_n &\twoheadrightarrow_\beta (\lambda x.r'')r_0 \cdots r_n \\ &\twoheadrightarrow_\eta (\lambda x.s'')s_0 \cdots s_n , \end{aligned}$$

so that we may take $r_{\mathfrak{D}} := (\lambda x.r'')r_0 \cdots r_n$. □

Corollary 6.8 (Leftmost $\beta\eta$-reduction) *The leftmost $\beta\eta$-reduction strategy is* normalizing, *i.e.: for every term t, if t has the $\beta\eta$-normal form s, then $t \overset{l}{\twoheadrightarrow} s$.*

Proof Given $\sigma : t \twoheadrightarrow s \in \mathrm{NF}_{\beta\eta}$, by combining (ii) of Lemma 6.2, Theorem 3.7* (*Left* τ-elimination for $\mathbf{G}_{\mathrm{ext}}[\beta]$) and Lemma 6.5 we obtain the *left normal* derivation

$$\mathsf{nor}(\mathsf{e}_L(\mathsf{lder}(\sigma))) \vdash^{-}_{L} t = s$$

whence, by (ii) of Theorem 6.7, a leftmost reduction path $t \overset{l}{\twoheadrightarrow} s$ can be extracted. □

Corollary 6.9 (Postponement of η-reductions) *If $\sigma : t \twoheadrightarrow s$ then there is a term r such that $t \twoheadrightarrow_\beta r \twoheadrightarrow_\eta s$.*

Proof r is extracted from $\mathsf{nor}(\mathsf{e}_L(\mathsf{lder}(\sigma))) \vdash^{-}_{L} t = s$ by means of the first half of (i) of Theorem 6.7. □

Corollary 6.10 ([5,14]) *t has a β normal form ⇔ t has a $\beta\eta$ normal form.*

Proof From right to left: as above, this time using also the second half of (i) of Theorem 6.7. From left to right: trivial. □

References

1. Aehlig, K., Joachimski, F.: Operational aspects of untyped normalization by evaluation. Math. Struct. Comput. Sci. **14**, 587–611 (2004)
2. Aspinall, D., Compagnoni, A.: Subtyping dependent types. Theor. Comput. Sci. **266**, 273–309 (2001)
3. Barendregt, H.P.: The Lambda Calculus, its Syntax and Semantics. Revised edition. North-Holland, Amsterdam (1984)
4. Barendregt, H.P.: Lambda Calculi with Types. In: Abramsky, S., Gabbay, D.M., Maibaum, T.S. (eds.) Handbook of Logic in Computer Science, vol. II, pp. 117–309. Clarendon Press, Oxford (1992)

5. Barendregt, H.P., Bergstra, J., Klop, J.W., Volken, A.: Some notes on lambda reduction. In: Degrees, reductions and representability in the lambda calculus, pp. 13–53. Preprint n. 22, University of Utrecht, Department of Mathematics (1976)
6. Chen, G.: Dependent type system with subtyping. Type level transitivity elimination. Technical Report LIENS-96-27, Laboratoire d'Informatique, Ecole Normale Supérieure, Paris (1996)
7. Chen, G.: Subtyping, type conversions and elimination of transitivity. PhD thesis, Université Paris VII (1998)
8. Chen, G.: Coercive subtyping for the Calculus of Constructions. In: Conference record of POPL 2003. The 30th SIGPLAN-SIGACT Symposium on Principles in Programming Languages, New Orleans, January 2003, pp. 150–159. SIGPLAN Notices **38/1** (2003)
9. Chen, G., Longo, G.: Subtyping parametric and dependent types. An introduction. http://www.di.ens.fr/~longo/download.html (1999)
10. Compagnoni, A.: Higher-order subtyping and its decidability. Inf. Comput. **191**, 41–113 (2004)
11. Coquand, T.: An algorithm for testing conversion in type theory. In: Huet, G., Plotkin, G. (eds.) Logical Frameworks, pp. 255–279. Cambridge University Press, Cambridge (1991)
12. Curien, P.-L., Ghelli, G.: Coherence and subsumption, minimum typing and type-checking in $F_{\leq}$. Math. Struct. Comput. Sci. **2**, 55–91 (1992)
13. Curry, H.B., Feys, R.: Combinatory Logic. vol. I, North-Holland, Amsterdam (1958)
14. Curry, H.B., Hindley, J.R., Seldin, J.P.: Combinatory Logic. vol. II, North-Holland, Amsterdam (1972)
15. David, R.: Une preuve simple de résultats classiques en λ-calcul. C. R. Acad. Sci. Paris Sér. I Math. **320**, 1401–1406 (1995)
16. Filinski, A., Rohde, H.K.: Denotational aspects of untyped normalization by evaluation. Theor. Inform. Appl. **39**, 423–453 (2005)
17. Gonthier, G., Lévy, J.-J., Melliès, P.-A.: An abstract standardisation theorem. In: Proceedings of the 7th Annual IEEE Symposium on Logic in Computer Science (LICS'92), Santa Cruz, California, August 1992, pp. 72–81. IEEE Computer Society Press (1992)
18. Hindley, J.R., Seldin, J.P.: Introduction to Combinators and λ–Calculus. Cambridge University Press, London (1986)
19. Kashima, R.: A proof of the standardization theorem in λ-calculus. RIMS Kokyuoroku **1217**, 37–44 (2001)
20. Kashima, R.: On the standardization theorem for $\lambda\beta\eta$-calculus. Contributed paper, International workshop on Rewriting in Proof and Computation (RPC'01), Sendai, Japan, October 2001.
21. Klop, J.W.: Combinatory reduction systems. Mathematical Center Tracts, 127. Mathematisch Centrum, Amsterdam (1980)
22. Lévy, J.-J.: Réductions correctes et optimales dans le λ-calcul. PhD thesis, Université Paris VII (1978)
23. Longo, G., Milsted, K., Soloviev, S.: Coherence and transitivity of subtyping as entailment. J. Log. Comput. **10**, 493–526 (2000)
24. McKinna, J., Pollack, R.: Some Lambda-Calculus and Type Theory formalized. J. Autom. Reasoning **23**, 373–409 (1999)
25. Melliès, P.-A.: Axiomatic rewriting theory I. A diagrammatic standardization theorem. In: Middeldorp, A., van Oostrom, V., van Raamsdonk, F., de Vrijer, R. (eds.) Processes, Terms and Cycles: steps on the road to infinity, pp. 554–638. Lectures Notes in Computer Science 3838, Springer, Berlin (2005)
26. Minari, P.: Analytic combinatory calculi and the elimination of transitivity. Arch. Math. Logic **43**, 159–191 (2004)
27. Minari, P.: Proof-theoretical methods in combinatory logic and λ-calculus. In: Cooper, S.B., Löwe, B., Torenvliet, L. (eds.) CiE 2005: New Computational Paradigms, pp. 148–157. ILLC X-2005-01, Amsterdam (2005)
28. Mitschke, G.: The standardization theorem for the λ-calculus. Z. Math. Logik Grundlagen Math. **25**, 29–31 (1979)
29. Takahashi, M.: Parallel Reductions in λ-Calculus. Inf. Comput. **118**, 120–127 (1995)
30. Xi, H.: Upper bounds for standardizations and an application. J. Symb. Log. **64**, 291–303 (1999)