Towards a Semantic Measure of the Execution Time in Call-by-Value lambda-Calculus

Giulio Guerrieri
Towards a Semantic Measure of the Execution Time in Call-by-Value lambda-Calculus

Giulio Guerrieri
Dipartimento di Informatica – Scienze e Ingegneria (DISI), Università di Bologna, Bologna, Italy
giulio.guerrieri@unibo.it

We investigate the possibility of a semantic account of the execution time (i.e. the number of βv-steps leading to the normal form, if any) for the shuffling calculus, an extension of Plotkin’s call-by-value λ-calculus. For this purpose, we use a linear logic based denotational model that can be seen as a non-idempotent intersection type system: relational semantics. Our investigation is inspired by similar ones for linear logic proof-nets and untyped call-by-name λ-calculus. We first prove a qualitative result: a (possibly open) term is normalizable for weak reduction (which does not reduce under abstractions) if and only if its interpretation is not empty. We then show that the size of type derivations can be used to measure the execution time. Finally, we show that, differently from the case of linear logic and call-by-name λ-calculus, the quantitative information enclosed in type derivations does not lift to types (i.e. to the interpretation of terms). To get a truly semantic measure of execution time in a call-by-value setting, we conjecture that a refinement of its syntax and operational semantics is needed.

1 Introduction

Type systems enforce properties of programs, such as termination or deadlock-freedom. The guarantee provided by most type systems for the λ-calculus is termination.

Intersection types have been introduced as a way of extending simple types for the λ-calculus to “finite polymorphism”, by adding a new type constructor ∩ and new typing rules governing it. Contrarily to simple types, intersection types provide a sound and complete characterization of termination: not only typed programs terminate, but all terminating programs are typable as well (see [14, 15, 36, 30] where different intersection type systems characterize different notions of normalization). Intersection types are idempotent, that is, they verify the equation A ∩ A = A. This corresponds to an interpretation of a typed term t : A ∩ B as “t can be used both as data of type A and as data of type B”.

More recently [19, 29, 32, 11] (a survey can be found in [9]), non-idempotent variants of intersection types have been introduced: they are obtained by dropping the equation A ∩ A = A. In a non-idempotent setting, the meaning of the typed term t : A ∩ A ∩ B is refined as “t can be used twice as data of type A and once as data of type B”. This could give to programmers a way to keep control on the performance of their code and to count resource consumption. Finite multisets are the natural setting to interpret an associative, commutative and non-idempotent connective ∩: if A and B are non-idempotent intersection types, the multiset [A, A, B] represents the non-idempotent intersection type A ∩ A ∩ B.

Non-idempotent intersection types have two main features, both enlightened in de Carvalho’s system R [11]:

1. Bounds on the execution time: they go beyond simply qualitative characterisations of termination, as type derivations provide quantitative bounds on the execution time (i.e. on the number of β-steps to reach the β-normal form). Therefore, non-idempotent intersection types give intensional insights on programs, and seem to provide a tool to reason about complexity of programs. The approach is
defining a measure for type derivations and showing that the measure gives (a bound to) the length of the evaluation of typed terms.

2. Linear logic interpretation: non-idempotent intersection types are deeply linked to linear logic (LL) \[20\]. Relational semantics \[21\] — the category Rel of sets and relations endowed with the comonad of finite multisets — is a sort of “canonical” denotational model of LL; the Kleisli category Rel of the comonad is a CCC and then provides a denotational model of the ordinary (i.e. call-by-name) \(\lambda\)-calculus. Non-idempotent intersection types can be seen as a syntactic presentation of Rel: the semantics of a term \(t\) is the set of conclusions of all type derivations of \(t\).

These two facts together have a potential, fascinating consequence: denotational semantics may provide abstract tools for complexity analysis, that are theoretically solid, being grounded on LL.

Starting from \[11\], research on relational semantics/non-idempotent intersection types has proliferated: various works in the literature explore their power in bounding the execution time or in characterizing normalization \[12, 8, 6, 7, 5, 13, 34, 28, 9, 31\]. All these works study relational semantics/non-idempotent intersection types either in LL proof-nets (the graphical representation of proofs in LL), or in some variant of ordinary (i.e. call-by-name) \(\lambda\)-calculus. In the second case, the construction of the relational model \textbf{Rel}: sketched above essentially relies on Girard’s call-by-name translation \((\cdot)^n\) of intuitionistic logic into LL, which decomposes the intuitionistic arrow as \((A \Rightarrow B)^n = !(A^n \rightarrow B^n)\).

Ehrhard \[17\] showed that the relational semantics \textbf{Rel} of LL induces also a denotational model for the \textbf{call-by-value} \(\lambda\)-calculus that can still be viewed as a non-idempotent intersection type system. The syntactic counterpart of this construction is Girard’s (“boring”) call-by-value translation \((\cdot)^v\) of intuitionistic logic into LL \[20\], which decomposes the intuitionistic arrow as \((A \Rightarrow B)^v = !(A^v \rightarrow B^v)\). Just few works have started the study of relational semantics/non-idempotent intersection types in a call-by-value setting \[17, 16, 10, 18\], and no one investigates their bounding power on the execution time in such a framework. The aim of our paper is to fill this gap and to study the information enclosed in relational semantics/non-idempotent intersection types concerning the execution time in the call-by-value \(\lambda\)-calculus.

A difficulty arises immediately in the qualitative characterization of call-by-value normalization via the relational model. One would expect that the semantics of a term \(t\) is non-empty if and only if \(t\) is (strongly) normalizable for (some restriction of) the call-by-value evaluation \(\rightarrow_{\beta_v}\), but it is impossible to get this result in Plotkin’s original call-by-value \(\lambda\)-calculus \(\lambda_v\) \[35\]. Indeed, the terms \(t\) and \(u\) below are \(\beta_v\)-normal but their semantics in the relational model are empty:

\[
t := (\lambda_y.\Delta)(zI)\Delta \quad u := \Delta((\lambda_y.\Delta)(zI)) \quad \text{(where } \Delta := \lambda x.x \text{ and } I := \lambda x.x) \tag{1}
\]

Actually, \(t\) and \(u\) should behave like the famous divergent term \(\Delta\Delta\), since in \(\lambda_v\) they are observationally equivalent to \(\Delta\Delta\) with respect all closing contexts and have the same semantics as \(\Delta\Delta\) in all non-trivial denotational models of Plotkin’s \(\lambda_v\).

The reason of this mismatching is that in \(\lambda_v\) there are \textbf{stuck} \(\beta\)-\textbf{redexes} such as \((\lambda_y.\Delta)(zI)\) in Eq. (1), i.e. \(\beta\)-redexes that \(\beta_v\)-reduction will never fire because their argument is normal but not a value (nor will it ever become one). The real problem with stuck \(\beta\)-redexes is that they may prevent the creation of other \(\beta_v\)-redexes, providing “\textbf{premature}” \(\beta_v\)-\textbf{normal forms} like \(t\) and \(u\) in Eq. (1). The issue affects termination and thus can impact on the study of observational equivalence and other operational properties in \(\lambda_v\).

\[1\] In call-by-value evaluation \(\rightarrow_{\beta_v}\), function’s arguments are evaluated before being passed to the function, so that \(\beta\)-redexes can fire only when their arguments are values, i.e. abstractions or variables. Call-by-value evaluation is the most common parameter passing mechanism used by programming languages.
In a call-by-value setting, the issue of stuck β-redexes and then of premature βv-normal forms arises only with open terms (in particular, when the reduction under abstractions is allowed, since it forces to deal with “locally open” terms). Even if to model functional programming languages with a call-by-value parameter passing, such as OCaml, it is usually enough to just consider closed terms and evaluation not reducing under abstractions (i.e. function bodies are evaluated only when all parameters are supplied), the importance to consider open terms in a call-by-value setting can be found, for example, in partial evaluation (which evaluates a function when not all parameters are supplied, see [26]), or to reason about (denotational or operational) equivalences of terms in λv that are congruences, or about other theoretical properties of λv such as separability or solvability [33, 39, 8, 10].

In order to overcome this issue, we study relational semantics/non-idempotent intersection types in the shuffling calculus λsh, a conservative extension of Plotkin’s λv proposed in [10] and further studied in [23, 24, 2, 25]. It keeps the same term syntax as λv and adds to βv-reduction two commutation rules, σ1 and σ3, which “shuffle” constructors in order to move stuck β-redexes: they unblock βv-redexes that are hidden by the “hyper-sequential structure” of terms. These commutation rules (referred also as σ-reduction rules) are similar to Regnier’s σ-rules for the call-by-name λ-calculus [37, 38] and are inspired by the aforementioned (·)v translation of the λ-calculus into LL proof-nets.

Following the same approach used in [11] for the call-by-name λ-calculus and in [12] for LL proof-nets, we prove that in the shuffling calculus λsh:

1. (qualitative result) a possibly open term is normalizable for weak reduction (not reducing under λ’s) if and only if its interpretation in the relational semantics is not empty (Thm. 16); this result was already proven in [10] using different techniques;

2. (quantitative result) the size of type derivations can be used to measure the execution time, i.e. the number of βv-steps (and not σ-steps) to reach the normal form of the weak reduction (Prop. 21).

Finally, we show that, differently from the case of LL and call-by-name λ-calculus, we are not able to lift the quantitative information enclosed in type derivations to types (i.e. to the interpretation of terms) following the same technique used in [11, 12], as our Ex. 25 shows. In order to get a genuine semantic measure of execution time in a call-by-value setting, we conjecture that a refinement of its syntax and operational semantics is needed.

Even if our main goal has not yet been achieved, this investigation leaded to new interesting results:

1. all normalizing weak reduction sequences (if any) in λsh from a given term have the same number of βv-steps (Cor. 22); this is not obvious, as we shall explain in Ex. 23;

2. terms whose weak reduction in λsh ends in a value has an elegant semantic characterization (Prop. 18), and the number of βv-steps needed to reach their normal form can be computed in a simple way from a specific type derivation (Thm. 24).

Omitted proofs are in Appendix A together with a list of notations and terminology used here.

2 The shuffling calculus

In this section we introduce the shuffling calculus λsh, namely the call-by-value λ-calculus defined in [10] and further studied in [23, 24, 2, 25]: it adds two commutation rules — the σ1- and σ3-reductions — to Plotkin’s pure (i.e. without constants) call-by-value λ-calculus λv [35]. The syntax for terms of λsh is the same as Plotkin’s λv and then the same as the ordinary (i.e. call-by-name) λ-calculus, see Fig. 1.
we define the \( \beta \)-contractum \( v \mapsto_{\beta} u \), \( \therefore \rightarrow_{\beta} \) from \( u \), \( \Delta \) different \( \text{kind may} \) \( \text{capture-allowing substitution of the term} \ t \) \( \text{of} \) \( \text{terms endowed with the reduction} \ r \text{-renaming. For any} \ r \in \{ \beta, \sigma_1, \sigma_3, \sigma, \text{sh} \}, \) \( \text{if} \ t \rightarrow_{r} t' \) then \( t \) is a \( \beta \)-redex and \( t' \) is its \( r \)-contractum. A term of the shape \( (\lambda x.t)u \) is a \( \beta \)-redex. Clearly, any \( \beta \)-redex is a \( \beta \)-redex but the converse does not hold: \( (\lambda x.2)(y1) \) is a \( \beta \)-redex but not a \( \beta \)-redex. Redexes of different kind may overlap: for instance, the term \( \Delta I \Lambda \sigma \) is a \( \sigma_1 \)-redex and contains the \( \beta_\gamma \)-redex \( \Delta I \Lambda \); the term \( \Delta(I\Lambda)(x1) \) is a \( \sigma_1 \)-redex and contains the \( \sigma_3 \)-redex \( \Delta I \Lambda \), which contains in turn the \( \beta_\gamma \)-redex \( I \Lambda \).

From definitions in Fig. 1 it follows that \( \rightarrow_{\text{sh}} = \rightarrow_{\beta} \cap \rightarrow_{\sigma} \) and \( \rightarrow_{\sigma} = \rightarrow_{\alpha} \cap \rightarrow_{\sigma_1} \), as well as \( \rightarrow_{\text{sh}} = \rightarrow_{\beta} \cap \rightarrow_{\sigma} \cap \rightarrow_{\sigma_1} \). The shuffling (resp. balanced shuffling) calculus \( \lambda_{\text{sh}} \) (resp. \( \lambda^3_{\text{sh}} \)) is the set \( \Lambda \) of terms endowed with the reduction \( \rightarrow_{\text{sh}} \) (resp. \( \rightarrow_{\text{sh}} \)). The set \( \Lambda \) endowed with the reduction \( \rightarrow_{\beta} \) is Plotkin’s pure call-by-value \( \lambda \)-calculus \( \lambda_{\beta} \) [35], a sub-calculus of \( \lambda_{\text{sh}} \).

**Proposition 1** (Basic properties of reductions, [35] [10]). The \( \sigma \)- and \( \sigma^3 \)-reductions are confluent and strongly normalizing. The \( \beta_\gamma \), \( \beta_\delta \), \( \text{sh}- \) and \( \text{sh}^3 \)-reductions are confluent.

**Example 2.** Recall the terms \( t \) and \( u \) in Eq. 1: \( t = (\lambda y.\Delta)(x1) \Delta \rightarrow_{\sigma_1} (\lambda y.\Delta)(x1) \rightarrow_{\beta_1} (\lambda y.\Delta)(x1) \rightarrow_{\beta_1} \ldots \) and \( u = (\lambda y.\Delta)(x1) \Delta \rightarrow_{\sigma_1} (\lambda y.\Delta)(x1) \rightarrow_{\beta_1} (\lambda y.\Delta)(x1) \rightarrow_{\beta_1} \ldots \) are the only possible sh-reduction paths from \( t \) and \( u \) respectively: \( t \) and \( u \) are not sh-normalizable and \( t \equiv_{\text{sh}} u \). But \( t \) and \( u \) are \( \beta_\gamma \)-normal \( ((\lambda y.\Delta)(x1)) \) is a stuck \( \beta_\gamma \)-redex and different, hence \( t \nless_{\beta_\gamma} u \) by confluence of \( \rightarrow_{\beta_\gamma} \) (Prop. 1).

Example 2 shows how \( \sigma \)-reduction shuffles constructors and moves stuck \( \beta_\gamma \)-redexes in order to unblock \( \beta_\gamma \)-redexes which are hidden by the “hyper-sequential structure” of terms, avoiding “premature” normal
forms. An alternative approach to circumvent the issue of stuck \(\beta\)-redexes is given by \(\lambda_{\text{sub}}\), the call-by-value \(\lambda\)-calculus with explicit substitutions introduced in [3], where hidden \(\beta\)-redexes are reduced using rules acting at a distance. In [2] it has been shown that \(\lambda_{\text{sub}}\) and \(\lambda_{\text{sh}}\) can be embedded in each other preserving termination and divergence. Interestingly, both calculi are inspired by an analysis of Girard’s “boring” call-by-value translation of \(\lambda\)-terms into linear logic proof-nets [20,11] according to the linear recursive type \(o = !o \rightarrow !o\), or equivalently \(o = !(o \rightarrow o)\). In this translation, \(sh\)-reduction corresponds to cut-elimination, more precisely \(\beta\)-steps (resp. \(\sigma\)-steps) correspond to exponential (resp. multiplicative) cut-elimination steps; \(sh\)-reduction corresponds to cut-elimination at depth 0.

Consider the two subsets of terms defined by mutual induction (notice that \(\Lambda_a \subseteq \Lambda_o \supseteq \Lambda_n\)):

\[
a := xv \mid xa \mid an \quad \text{(set: } \Lambda_a) \quad n := v \mid (\lambda x.n)a \quad \text{(set: } \Lambda_n).
\]

Any \(t \in \Lambda_a\) is neither a value nor a \(\beta\)-redex, but an open applicative term with a free “head variable”.

**Proposition 3** (Syntactic characterization on \(sh^3\)-normal forms). Let \(t\) be a term: \(Proof p. 14\)

- \(t\) is \(sh^3\)-normal iff \(t \in \Lambda_a\);
- \(t\) is \(sh^3\)-normal and is neither a value nor a \(\beta\)-redex iff \(t \in \Lambda_n\).

Stuck \(\beta\)-redexes correspond to \(sh^3\)-normal forms of the shape \((\lambda x.n)a\). As a consequence of Prop. 3, the behaviour of closed terms with respect to \(sh^3\)-reduction is quite simple: either they diverge or they \(sh^3\)-normalize to a (closed) value. Indeed:

**Corollary 4** (Syntactic characterization of closed \(sh^3\)-normal forms). Let \(t\) be a closed term: \(t\) is \(sh^3\)-normal iff \(t = \lambda x.u\) for some term \(u\) with \(fv(u) \subseteq \{x\}\). \(Proof p. 15\)

### 3 A non-idempotent intersection type system

We aim to define a non-idempotent intersection types system in order to characterize the (strong) normalizable terms for the reduction \(\rightarrow_{sh}\).

**Types** are positive or negative types, defined in turn by mutual induction as follows:

- **Negative Types**: \(M, N ::= P \rightarrow Q\)
- **Positive Types**: \(P, Q ::= [N_1, \ldots, N_n] \quad \text{(with } n \in \mathbb{N})\)

where \([N_1, \ldots, N_n]\) is a (possibly empty) finite multiset of negative types; in particular the empty multiset \([]\) (obtained for \(n = 0\)) is the atomic (positive) type. A positive type \([N_1, \ldots, N_n]\) has to be intended as a conjunction \(N_1 \land \cdots \land N_n\) of negative types \(N_1, \ldots, N_n\), for a commutative and associative conjunction connective \(\land\) that is not idempotent and whose neutral element is \([]\).

The derivation rules for the non-idempotent intersection types system are in Fig. 2. In this typing system, **judgments** have the shape \(\Gamma \vdash t : P\) where \(t\) is a term, \(P\) is a positive type and \(\Gamma\) is an environment (that is, a partial function from variables to positive types whose domain is finite). The **sum of environments** \(\Gamma \uplus \Delta\) is defined pointwise using the multiset sum, when \(\Gamma\) and \(\Delta\) have the same domain: if \(x \in \text{dom}(\Gamma) = \text{dom}(\Delta)\), then \((\Gamma \uplus \Delta)(x) = \Gamma(x) \uplus \Delta(x)\). An environment \(\Gamma\) such that \(\text{dom}(\Gamma) = \{x_1, \ldots, x_n\}\) with \(x_i \neq x_j\) and \(\Gamma(x_i) = m_i\) for all \(1 \leq i \neq j \leq n\) is often written as \(\Gamma = x_1 : m_1, \ldots, x_n : m_n\). Note that the sum of environments \(\uplus\) is commutative, associative and, given an environment \(\Gamma\) with \(\text{dom}(\Gamma) = \{x_1, \ldots, x_n\}\) (for some \(n \in \mathbb{N}\)), one has \(\Gamma \uplus \Gamma_0 = \Gamma\) iff \(\Gamma_0 = x_1 : [], \ldots, x_n : []\) (i.e. \(\Gamma_0\) is the neutral element for \(\uplus\)). The notation \(\pi \triangleright \Gamma \vdash t : P\) means that \(\pi\) is a derivation with conclusion the judgment \(\Gamma \vdash t : P\). We write \(\pi \triangleright t\) if \(\pi\) is such that \(\pi \triangleright \Gamma \vdash t : P\) for some environment \(\Gamma\) and positive type \(P\).
Towards a Semantic Measure of the Execution Time in Call-by-Value lambda-Calculus

\[
\frac{\Gamma_0, x : P \vdash x : P}{\text{ax}} \quad \frac{\Gamma \vdash t : [P \rightarrow Q]}{\Gamma \uplus t : P \vdash t : Q}\]

\[
\frac{\Gamma_1, x : P_1 \vdash t : Q_1}{\Gamma_0 \uplus \Gamma_1 \vdots \vdots \Gamma_n \vdash \lambda x : [ \lambda x_1 : [\ldots \lambda x_n : [\ldots Q_n \rightarrow Q_n] \ldots] \ldots]}
\]

Figure 2: Non-idempotent intersection type system for the shuffling calculus. In the rules \text{ax} and \text{\lambda}, \Gamma_0 = x_1 : [\ldots, x_n : [\ldots (w i f x \neq x_i \text{ for all } 1 \leq i \leq n), and in the rule \text{\lambda}, \text{fv}(t) \subseteq \text{dom}(\Gamma_0).

It is worth noticing that the typing system in Fig. 2 is syntax oriented: for each type judgment \(J\) there is a unique derivation rule whose conclusion matches the judgment \(J\).

The size \(|\pi|\) of a type derivation \(\pi\) is just the number of \(\text{\@}\) rules in \(\pi\). Note that judgments play no role in the size of a derivation.

Example 5. Let \(I = \lambda x.x\) and \(x, x_1, \ldots, x_k\) be pairwise distinct variable (with \(k \in \mathbb{N}\)). Setting \(\Gamma = x_1 : [\ldots, x_k : [\ldots\), the derivations (typing \(I\) and \(I\) with same type and same context)

\[
\pi_I = \frac{\Gamma, x : [\ldots \vdash x : [\ldots]}{\Gamma \vdash I : [\ldots \rightarrow [\ldots\lambda}_{\text{\@}} \quad \pi_I = \frac{\Gamma \vdash I : [\ldots}{\Gamma \vdash I : [\ldots\lambda}\]
\]

are such that \(|\pi_I| = 1\) and \(|\pi_I| = 0\). Note that \(I \rightarrow_{sh} I\) and \(|\pi_I| = |\pi_I| + 1\).

The following lemma (whose proof is quite technical) will play a crucial role to prove the substitution lemma (Lemma 7) and the subject reduction (Prop. 8) and expansion (Prop. 10).

At the direction of Lemma 6 means that, given \(\pi \vartriangleright \Delta \vdash v : P\), for every \(p \in \mathbb{N}\) and every decomposition of the positive type \(P\) into a multiset sum of positive types \(P_1, \ldots, P_p\), there are environments \(\Delta_1, \ldots, \Delta_p\) such that \(\Delta_i \vdash v : P_i\) is derivable for all \(1 \leq i \leq p\).

Lemma 7 (Substitution). Let \(t \in \Lambda\) and \(v \in \Lambda_v\). If \(\pi \vartriangleright \Gamma, x : P \vdash t : Q\) and \(\pi' \vartriangleright \Delta \vdash v : P\), then there exists \(\pi'' \vartriangleright \Gamma \uplus \Delta \vdash t\langle v/x \rangle : Q\) such that \(|\pi''| = |\pi| + |\pi'|\).

We can now prove the subject reduction, with a quantitative flavour about the size of type derivations in order to extract information about the execution time.

Proposition 8 (Quantitative balanced subject reduction). Let \(t, t' \in \Lambda\) and \(\pi \vartriangleright \Gamma \vdash t : Q\).

1. Shrinkage under \(\beta^\text{sh}\) step: If \(t \rightarrow_{\text{\beta^\text{sh}}} t'\) then \(|\pi'| > 0\) and there exists a derivation \(\pi'\) with conclusion \(\Gamma \vdash t' : Q\) such that \(|\pi'| = |\pi| - 1\).

2. Size invariance under \(\sigma^\text{sh}\) step: If \(t \rightarrow_{\sigma^\text{sh}} t'\) then \(|\pi'| > 0\) and there exists a derivation \(\pi'\) with conclusion \(\Gamma \vdash t' : Q\) such that \(|\pi'| = |\pi|\).

In Prop. 8 the fact that \(\rightarrow_{\text{\beta^\text{sh}}}\) does not reduce under \(\lambda\)'s is crucial to get the quantitative information, otherwise one can have a term \(t\) such that every derivation \(\pi \vartriangleright \Gamma \vdash t : P\) is such that \(|\pi| = 0\) (and then there is no derivation \(\pi'\) with conclusion \(\Gamma \vdash t' : P\) such that \(|\pi'| = |\pi| - 1\)). this is the case, for example, for \(t = \lambda x. \delta \delta \rightarrow_{\beta^\text{sh}} t\).

In order to prove the quantitative subject expansion (Prop. 10), we first need the following technical lemma stating the commutation of abstraction with abstraction and application.
Lemma 9 (Abstraction commutation).  
1. Abstraction vs. abstraction: Let \( n \in \mathbb{N} \). If \( \pi \vdash \Delta \vdash \lambda.y.(\lambda.x.t)v : \bigcup_{i=1}^{n} [P'_i \to \top] \) and \( y \notin \text{fv}(v) \), then there is \( \pi' \vdash \Delta \vdash \lambda.x.(\lambda.y.t)\nu : \bigcup_{i=1}^{n} [P'_i \to \top] \) such that \( |\pi'| = |\pi| + 1 - n \).

2. Application vs. abstraction: If \( \pi \vdash \Delta \vdash ((\lambda.x.t)v)((\lambda.x.u)v) : P \) then there exists a derivation \( \pi' \vdash \Delta \vdash (\lambda.x.\mu) : P \) such that \( |\pi'| = |\pi| - 1 \).

Proposition 10 (Quantitative balanced subject expansion). Let \( t, t' \in \Lambda \) and \( \pi' \vdash \Gamma \vdash \pi \vdash t : Q \).

1. Enlargement under anti-\( \beta_0 \)-step: If \( t \rightarrow \beta_0 t' \) then there is \( \pi \vdash \Gamma \vdash t : Q \) with \( |\pi| = |\pi'| + 1 \).

2. Size invariance under anti-\( \sigma_0 \)-step: If \( t \rightarrow_{\sigma} t' \) then there exists \( \pi' \vdash \Gamma \vdash t : Q \) such that \( |\pi| = |\pi'| \).

Actually, subject reduction and expansion hold for the whole \( sh \)-reduction \( \rightarrow_{sh} \), not only for the balanced \( sh \)-reduction \( \rightarrow_{sh} \). The drawback for \( \rightarrow_{sh} \) is that the quantitative information about the size of the derivation is lost in the case of a \( \beta_0 \)-step.

Lemma 11 (Subject reduction). Let \( t, t' \in \Lambda \) and \( \pi \vdash \Gamma \vdash t : Q \).

1. Shrinkage under \( \beta_0 \)-step: If \( t \rightarrow \beta_0 t' \) then there is \( \pi \vdash \Gamma \vdash t : Q \) with \( |\pi| \geq |\pi'| \).

2. Size invariance under \( \sigma_0 \)-step: If \( t \rightarrow_{\sigma} t' \) then \( \pi \vdash \Gamma \vdash t : Q \) such that \( |\pi| = |\pi'| \).

Lemma 12 (Subject expansion). Let \( t, t' \in \Lambda \) and \( \pi' \vdash \Gamma \vdash t : Q \).

1. Enlargement under anti-\( \beta_0 \)-step: If \( t \rightarrow \beta_0 t' \) then there is \( \pi \vdash \Gamma \vdash t : Q \) with \( |\pi| \geq |\pi'| \).

2. Size invariance under anti-\( \sigma_0 \)-step: If \( t \rightarrow_{\sigma} t' \) then \( \pi \vdash \Gamma \vdash t : Q \) such that \( |\pi| = |\pi'| \).

In Lemmas 11 and 12 it is impossible to estimate more precisely the relationship between \( |\pi| \) and \( |\pi'| \). Indeed, Ex. 5 has shown that there are \( \pi_I \vdash y : [] \vdash I : [] \) and \( \pi_{II} \vdash y : [] \vdash II : [] \) such that \( |\pi_I| = 0 \) and \( |\pi_{II}| = 1 \) (where \( I = \lambda.x.x \)). So, given \( n \in \mathbb{N} \), consider the derivations \( \pi_n \vdash \lambda.y.II : [\cdots \to []] \) and \( \pi'_n \vdash \lambda.y.I : [\cdots \to []] \) below:

\[
\begin{align*}
\pi_n &= y : [] \vdash II : [] \quad \pi'_n &= y : [] \vdash I : [] \\
\lambda \downarrow y.II &\vdash [\cdots \to []] \\
\lambda \downarrow y.I &\vdash [\cdots \to []]
\end{align*}
\]

Clearly, \( \lambda.y.II \rightarrow_{sh} \lambda.y.I \) (but \( \lambda.y.II \not\rightarrow_{sh} \lambda.y.I \)) and the \( \pi'_n \) (resp. \( \pi_n \)) is the only derivation typing \( \lambda.y.I \) (resp. \( \lambda.y.II \)) with the same type and environment as \( \pi_n \) (resp. \( \pi'_n \)). One has \( |\pi_n| = n \cdot |\pi| = n \) and \( |\pi'_n| = n \cdot |\pi| = n \), thus the difference of size of the derivations \( \pi_n \) and \( \pi'_n \) can be arbitrarily large (since \( n \in \mathbb{N} \)); in particular \( |\pi_0| = |\pi'_0| \), so for \( n = 0 \) the size of derivations does not even strictly decrease.

4 Relational semantics: qualitative results

Lemmas 11 and 12 have an important consequence: the non-idempotent intersection type system of Fig. 2 defines a denotational model for the shuffling calculus \( \lambda_{sh} \) (Thm. 14 below).

Definition 13 (Suitable list of variables for a term, semantics of a term). Let \( t \in \Lambda \) and \( x_1, \ldots, x_n \) be pairwise distinct variables, for some \( n \in \mathbb{N} \).

If \( \text{fv}(t) \subseteq \{ x_1, \ldots, x_n \} \), then we say that the list \( \bar{x} = (x_1, \ldots, x_n) \) is suitable for \( t \).

If \( \bar{x} = (x_1, \ldots, x_n) \) is suitable for \( t \), the (relational) semantics, or interpretation, of \( t \) for \( \bar{x} \) is

\[
[t]_{\bar{x}} = \{ ((P_1, \ldots, P_n), Q) \mid \exists \pi \vdash x_1 : P_1, \ldots, x_n : P_n \vdash t : Q \}.
\]
Essentially, the semantics of a term $t$ for a suitable list $\bar{x}$ of variables is the set of judgments for $\bar{x}$ and $t$ that can be derived in the non-idempotent intersection type system of Fig. 2.

If we identify the type $P \rightarrow Q$ with the pair $(P, Q)$ and if we set $\mathcal{U} := \bigcup_{k \in \mathbb{N}} \mathcal{U}_k$ where:

$$\mathcal{U}_0 := \emptyset \quad \mathcal{U}_{k+1} := \mathcal{M}_!(\mathcal{U}_k) \times \mathcal{M}_!(\mathcal{U}_k) \quad (\mathcal{M}_!(X) \text{ is the set of finite multisets over the set } X)$$

then, for any $t \in \Lambda$ and any suitable list $\bar{x} = (x_1, \ldots, x_n)$ for $t$, one has $[t]_{\bar{x}} \subseteq \mathcal{M}_!(\mathcal{U})^k \times \mathcal{M}_!(\mathcal{U})^k$; in particular, if $t$ is closed and $\bar{x} = ()$, then $[t] = \{ Q \mid \exists \pi \vdash t : Q \} \subseteq \mathcal{M}_!(\mathcal{U})$ (up to an obvious isomorphism). Note that $\mathcal{U} = \mathcal{M}_!(\mathcal{U}) \times \mathcal{M}_!(\mathcal{U})$: [17][10] proved that the latter identity is enough to have a denotational model for $\lambda_{sh}$. We can also prove it explicitly using Lemmas 11 and 12.

**Proof.**

**Theorem 14** (Invariance under $sh$-equivalence). Let $t, u \in \Lambda$, let $n \in \mathbb{N}$ and let $\bar{x} = (x_1, \ldots, x_n)$ be a suitable list of variables for $t$ and $u$. If $t \equiv_{sh} u$ then $[t]_{\bar{x}} = [u]_{\bar{x}}$.

An interesting property of relational semantics is that all $sh^\flat$-normal forms have a non-empty interpretation (Lemma 15). To prove that we use the syntactic characterization of $sh^\flat$-normal forms (Prop. 3). Note that a stronger statement (Lemma 15.1) is required for $sh^\flat$-normal forms belonging to $\Lambda^d$, in order to handle the case where the $sh^\flat$-normal form is a $\beta$-redex.

**Lemma 15** (Semantics of $sh^\flat$-normal forms). Let $t$ be a term, let $k \in \mathbb{N}$ and let $\bar{x} = (x_1, \ldots, x_k)$ be a list of variables suitable for $t$.

1. If $t \in \Lambda_d$ then for every positive type $Q$ there exist positive types $P_1, \ldots, P_k$ and a derivation $\pi \vdash x_1 : P_1, \ldots, x_k : P_k \vdash t : Q$.
2. If $t \in \Lambda_n$ then there are positive types $Q, P_1, \ldots, P_k$ and a derivation $\pi \vdash x_1 : P_1, \ldots, x_k : P_k \vdash t : Q$.
3. If $t$ is $sh^\flat$-normal then $[t]_{\bar{x}} \neq \emptyset$.

A consequence of Prop. 8 and Thm. 14 and Lemma 15 is a qualitative result: a semantic and logical (if we consider our non-idempotent type system as a logical framework) characterization of (strong) $sh^\flat$-normalizable terms (Thm. 16). In this theorem, the main equivalences are between Points 1, 3 and 5, already proven in [10] using different techniques. Points 2, 4 and 5 can be seen as “intermediate stages” in the proof of the main equivalences, which are informative enough to deserve to be explicitly stated.

**Theorem 16** (Semantic and logical characterization of $sh^\flat$-normalization). Let $t \in \Lambda$ and let $\bar{x} = (x_1, \ldots, x_n)$ be a suitable list of variables for $t$. The following are equivalent:

1. $t$ is $sh^\flat$-normalizable;
2. $t \equiv_{sh} u$ for some $sh^\flat$-normal $u \in \Lambda$;
3. $[t]_{\bar{x}} \neq \emptyset$;
4. there exists a derivation $\pi \vdash x_1 : P_1, \ldots, x_n : P_n \vdash t : Q$ for some positive types $P_1, \ldots, P_n, Q$;
5. $t$ is strongly $sh^\flat$-normalizable.

Equivalence (5) implies (1) means that normalization and strong normalization are equivalent for $sh^\flat$-reduction, thus in studying the termination of $sh^\flat$-reduction no intricacy arises from its non-determinism. Equivalence (1) implies (2) says that $sh^\flat$-reduction is complete to get $sh^\flat$-normal forms; in particular, this entails that every $sh$-normalizable term is $sh^\flat$-normalizable. Equivalence (1) implies (2) is the analogue of a well-known theorem [4 Thm. 8.3.11] for ordinary (i.e. call-by-name) $\lambda$-calculus relating head $\beta$-reduction and $\beta$-equivalence: this corroborates the idea that $sh^\flat$-reduction is the “head reduction” in a call-by-value setting, despite its non-determinism.

The implication (5) implies (3) does not hold in Plotkin’s $\lambda_{sh}$: indeed, the terms $t$ and $u$ in Eq. (1) (see also Ex. 2) are $\beta$-normal (because of a stuck $\beta$-redex) but $[t]_{\bar{x}} = \emptyset = [u]_{\bar{x}}$. Equivalences such as the ones in Thm. 16 hold in a call-by-value setting provided that $\beta$-reduction is extended, e.g. by adding $\sigma$-reduction (see [2] for other equivalent approaches).
Lemma 17 (Uniqueness of the derivation with empty types; Semantic and logical characterization of values). Let \( t \in \Lambda \) be sh\(^b\)-normal and \( \vec{x} = (x_1, \ldots, x_k) \) be suitable for \( t \). 

1. For every \( \pi \vdash x_1 : [], \ldots, x_k : [] \vdash t : [] \) and \( \pi' \vdash x_1 : [], \ldots, x_k : [] \vdash t : [] \), one has \( t \in \Lambda_{\pi} \), \( |\pi| = 0 \) and \( \pi = \pi' \). More precisely, \( \pi \) consists only of the rule \( \exists x \) if \( t \) is a variable, otherwise \( t \) is an abstraction and \( \pi \) consists only of a 0-ary rule \( \lambda \).
2. The following are equivalent:
   
   \[
   (a) \quad t \text{ is a value}; \quad \quad (c) \quad \text{there exists } \pi \vdash x_1 : [], \ldots, x_k : [] \vdash t : []; \\
   (b) \quad (([], \ldots, []), []) \in [t]_{\vec{x}}; \quad \quad (d) \quad \text{there exists } \pi \vdash t \text{ such that } |\pi| = 0.
   \]

Qualitatively, Lemma 17 allows us to refine the semantic and logical characterization given by Thm. 16 for a specific class of terms: the valuable ones, i.e. the terms that sh\(^b\)-normalize to a value. Valuable terms are all only the terms whose semantics contains a specific element: the point with only empty types.

Proposition 18 (Logical and semantic characterization of valuability). Let \( t \) be a term and \( \vec{x} = (x_1, \ldots, x_k) \) be suitable for \( t \). The following are equivalent:

1. \( t \) is sh\(^b\)-normalizable and the sh\(^b\)-normal form of \( t \) is a value;
2. \( (([], \ldots, []), []) \in [t]_{\vec{x}}; \)
3. there exists \( \pi \vdash x_1 : [], \ldots, x_k : [] \vdash t : [] \).

5 The quantitative side of type derivations

From the quantitative subject reduction (Prop. 8) it follows immediately that the size of any derivation typing a (sh\(^b\)-normalizable) term \( t \) is an upper bound on the number of \( \beta^b \)-steps in any sh\(^b\)-normalizing reduction sequence from \( t \), since the size of a type derivation decreases by 1 after each \( \beta^b \)-step, and does not change after each \( \sigma^b \)-step.

Corollary 19 (Upper bound for the number of \( \beta^b \)-steps). Let \( t \) be a sh\(^b\)-normalizable term and \( t_0 \) be its sh\(^b\)-normal form. For any reduction sequence \( d : t \rightarrow^{*}_{\rightarrow^{b}_{\text{sh}}} t_0 \) and any \( \pi \vdash t \), \( \text{length}(d) \leq |\pi| \).

In order to extract from a type derivation the exact number of \( \beta^b \)-steps to reach the sh\(^b\)-normal form, we have to take into account that a sh\(^b\)-normal form could have only type derivations whose size is not 0, so we should consider also the size of (type derivations of) sh\(^b\)-normal forms.

The balanced size of a term \( t \), denoted by \( |t|_{\vec{y}} \), is defined by induction on \( t \) as follows (\( v \in \Lambda_{\vec{y}} \)):

\[
|v|_{\vec{y}} = 0 \quad \quad |tu|_{\vec{y}} = \left\{ \begin{array}{ll}
|s|_{\vec{y}} + |u|_{\vec{y}} + 1 & \text{if } t = \lambda x. s \\
|t|_{\vec{y}} + |u|_{\vec{y}} + 1 & \text{otherwise}
\end{array} \right.
\]

So, the balanced size of a term \( t \) is the number of applications occurring in \( t \) under a balanced context, i.e. the number of pairs \((u, s)\) such that \( t = B(u s) \) for some balanced context \( B \). For instance, \( |(\lambda x. y y)(z z)|_{\vec{y}} = 3 \) and \( |(\lambda x. \lambda x'. y y)(z z)|_{\vec{y}} = 2 \). The following lemma can be seen as a quantitative version of Lemma 15.

Lemma 20 (Relationship between sizes). Let \( t \in \Lambda \), \( k \in \mathbb{N} \) and \( \vec{x} = (x_1, \ldots, x_k) \) suitable for \( t \).

1. If \( t \) is sh\(^b\)-normal then \( |t|_{\vec{y}} = \inf \{ |\pi| \mid \pi \vdash x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \} \).
2. If \( t \) is a value then \( |t|_{\vec{y}} = \inf \{ |\pi| \mid \pi \vdash x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \} = 0 \).
Thus, the balanced size of a $\text{sh}^b$-normal form $n$ equals the minimal size of the type derivation of $n$.

**Proposition 21** (Exact number of $\beta^b_i$-steps). Let $t$ be a $\text{sh}^b$-normalizable term and $t_0$ be its $\text{sh}^b$-normal form. For every reduction sequence $d : t \to^*_{\text{sh}^b} t_0$ and every $\pi \triangleright t$ and $\pi_0 \triangleright t_0$ such that $|\pi| = \inf\{|\pi'| \mid \pi' \triangleright t\}$ and $|\pi_0| = \inf\{|\pi'_0| \mid \pi'_0 \triangleright t_0\}$, one has

$$\text{leng}_{\beta^b_i}(d) = |\pi| - |t_0| = |\pi| - |\pi_0|.$$  \hspace{1cm} (2)

If moreover $t_0$ is a value, then $\text{leng}_{\beta^b_i}(d) = |\pi|$.

Prop. 21 could seem slightly disappointing: it allows us to know the exact number of $\beta^b_i$-steps of a $\text{sh}^b$-normalizing reduction sequence from $t$ only if we already know the $\text{sh}^b$-normal form $t_0$ of $t$ (or the minimal derivation of $t_0$), which essentially means that we have to perform the reduction sequence in order to know the exact number of its $\beta^b_i$-steps. However, Prop. 21 says also that this limitation is circumvented in the case $t$ $\text{sh}^b$-reduces to a value. Moreover, a notable and immediate consequence of Prop. 21 is:

**Corollary 22** (Same number of $\beta^b_i$-steps). Let $t$ be a $\text{sh}^b$-normalizable term and $t_0$ be its $\text{sh}^b$-normal form. For all reduction sequences $d : t \to^*_{\text{sh}^b} t_0$ and $d' : t \to^*_{\text{sh}^b} t_0$, $\text{leng}_{\beta^b_i}(d) = \text{leng}_{\beta^b_i}(d')$.

Even if $\text{sh}^b$-reduction is weak, in the sense that it does not reduce under $\lambda$'s, Cor. 22 is not obvious at all, since the rewriting theory of $\text{sh}^b$-reduction is not quite elegant, in particular it does not enjoy any form of (quasi-)diamond property because of $\sigma$-reduction, as shown by the following example.

**Example 23.** Let $t := (\lambda y . z) (\Delta (y f)) I$; one has $u := (\lambda y . y') (\Delta (y f)) s_1 \leftarrow t \sigma_1 \leftarrow (\lambda z . (\lambda y . y') (zz)) (xf) I =: s$ and the only way to join this critical pair is by performing one $\sigma^b_i$-step from $u$ and two $\sigma^b_i$-steps from $s$, so that $u \to^{s_1} \leftarrow (\lambda z . (\lambda y . y') (zz)) (xf) s_1 \leftarrow (\lambda z . (\lambda y . y') (zz)) (xf) s_1 \leftarrow s$. Since each $\sigma^b_i$-step can create a new $\beta^b_i$-redex in a balanced context (as shown in Ex. 2), **a priori** there is no evidence that Cor. 22 should hold.

Cor. 22 allows us to define the following function $\text{leng}_{\beta^b_i} : \Lambda \to \mathbb{N} \cup \{\infty\}$

$$\text{leng}_{\beta^b_i}(t) = \begin{cases} 
\text{leng}_{\beta^b_i}(d) & \text{if there is a } \text{sh}^b \text{-normalizing reduction sequence } d \text{ from } t; \\
\infty & \text{otherwise}.
\end{cases}$$

In other words, in $\lambda_{sh}$ we can univocally associate with every term the number of $\beta^b_i$-steps needed to reach its $\text{sh}^b$-normal form, if any (the infinite is associated with the non-$\text{sh}^b$-normalizable terms). The characterization of $\text{sh}^b$-normalization given in Thm. 16 allows us to determine through semantic or logical means if the value of $\text{leng}_{\beta^b_i}(t)$ is a finite number or not.

Quantitatively, via Lemma 17 we can simplify the way to compute the number of $\beta^b_i$-steps to reach the $\text{sh}^b$-normal form of a valuable (i.e. that reduce to a value) term $t$, using only a specific type derivation of $t$.

**Theorem 24** (Exact number of $\beta^b_i$-steps for valuables). Let $t \to^*_{\text{sh}^b} v \in \Lambda_v$. For any $\vec{x} = (x_1, \ldots, x_k)$ suitable for $t$, and any $\pi \triangleright x_1 : [], \ldots, x_k : [] \vdash t : []$, one has $\text{leng}_{\beta^b_i}(t) = |\pi|$.

Prop. 18 and Thm. 24 provide a procedure to determine whether a term $t$ $\text{sh}^b$-normalizes to a value or not, and, in how many $\beta^b_i$-steps are needed to reach its $\text{sh}^b$-normal form (this number does not depend on the reduction strategy according to Cor. 22), considering only the term $t$ and without performing any $\text{sh}^b$-reduction step:

1. check if there is a derivation $\pi$ with empty types, i.e. $\pi \triangleright x_1 : [], \ldots, x_k : [] \vdash t : []$;
2. if it is so, compute the size $|\pi|$.

Remind that, according to Cor. 3, any closed term either is not $\text{sh}^b$-normalizable, or it $\text{sh}^b$-normalizes to a (closed) value. So, this procedure completely determines (qualitatively and quantitatively) the behavior of closed terms with respect to $\text{sh}^b$-reduction.

We do not have any evidence that this procedure is more efficient than effectively computing the $\text{sh}^b$-normal form by performing $\text{sh}^b$-reduction steps.
6 Conclusions

In order to get a truly semantic measure of the execution time in the shuffling calculus $\lambda_{sh}$, we should first be able to give an upper bound to the number of $\beta^3$-steps in a $sh^3$-reduction looking only at the semantics of terms. Therefore, we need to define a notion of size for the elements of the semantics of terms. The most natural approach is the following. For every positive type $P = \langle P_1 \to Q_1, \ldots, P_k \to Q_k \rangle \in \mathcal{M}(\mathcal{U})$ (with $k \in \mathbb{N}$), the size of $P$ is $|P| = k + \sum_{i=1}^{k} (|P_i| + |Q_i|)$. So, the size of a positive type $P$ is the number of occurrences of $\to$ in $P$; in particular, $|[\null]| = 0$. For any $((P_1, \ldots, P_k), Q) \in \mathcal{M}(\mathcal{U})^k \times \mathcal{M}(\mathcal{U})$ (with $k \in \mathbb{N}$), the size of $((P_1, \ldots, P_k), Q)$ is $|((P_1, \ldots, P_k), Q)| = |Q| + \sum_{i=1}^{k} |P_i|$.

The approach of [11] [12] relies on a crucial lemma to find an upper bound (and hence the exact length) of the execution time: it relates the size of a type derivation to the size of its conclusion, for a normal term/proof-net. In $\lambda_{sh}$ this lemma should claim that “For every $sh$-normal form $t$, if $\pi \triangledown x_1 : P_1, \ldots, x_k : P_k \vdash t : Q$ then $|\pi| \leq |((P_1, \ldots, P_k), Q)|$”. Unfortunately, in $\lambda_{sh}$ this property is false!

Example 25. Let $t := \lambda z. (\lambda y. (\lambda x. (yy))(zz))$, which is a $sh$-normal closed value. Consider the derivation

$$
\pi := \lambda \beta^3 \vdash x : [\null] \rightarrow P \vdash x : [\null] \rightarrow P \downarrow x: [\null] \rightarrow P @.
$$

where, for any variable $x$ and any positive type $P$,

$$
\pi^P_x := \lambda \beta^3 \vdash x : [\null] \rightarrow P \vdash x : [\null] \rightarrow P \downarrow x: [\null] \rightarrow P @.
$$

Then, $|\pi| = 2 + |\pi^P_x| = 4 > 3 = |((\null) \rightarrow [\null] \rightarrow [\null] \rightarrow [\null])|$.

We conjecture that in order to overcome this counterexample (and to successfully follow the method of [11] [12] to get a purely semantic measure of the execution time) we should change the syntax and the operational semantics of our calculus, always remaining in a call-by-value setting equivalent (from the termination point of view) to $\lambda_{sh}$ and the other ones studied in [12]. Intuitively, in Ex. 25 $t$ contains two applications — $(\lambda x. (yy))$ and $(\lambda y. (\lambda x. (yy))(zz))$ — that are stuck $\beta$-redexes and are the source of two “useless” instances of the rule $@$ in $\pi$. The idea for the new calculus is to “fire” a stuck $\beta$-redex $(\lambda x. t)u$ without performing the substitution $t[u/x]$, but just creating an explicit substitution $t[u/x]$ that removes the application but “stores” the stuck $\beta$-redex.

Since $\lambda_{sh}$ is compatible with Girard’s call-by-value translation of $\lambda$-terms into LL proof-nets, it could seem surprising that some property holds in the general case of untyped LL proof-nets (as proven in [12]) but does not hold in the special case terms of $\lambda_{sh}$. Actually, there is no contradiction because LL-proof-nets in [12] always require an explicit constructor for dereliction, whereas $\lambda_{sh}$ is outside of this fragment since variables correspond to exponential axioms (which keep implicit the dereliction).

References

Towards a Semantic Measure of the Execution Time in Call-by-Value lambda-Calculus


A  Technical appendix: omitted proofs

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

A.1  Preliminaries and notations

The set of $\lambda$-terms is denoted by $\Lambda$. We set $I := \lambda x.x$ and $\Delta := \lambda x.xx$. Let $\Rightarrow$, $\subseteq \Lambda \times \Lambda$.

- The reflexive-transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$. The $r$-equivalence $\simeq_r$ is the reflexive-transitive and symmetric closure of $\Rightarrow_r$.

- Let $t$ be a term: $t$ is $r$-normal if there is no term $u$ such that $t \Rightarrow u$; $t$ is $r$-normalizable if there is a $r$-normal term $u$ such that $t \Rightarrow^* u$, and we then say that $u$ is a $r$-normal form of $t$; $t$ is strongly $r$-normalizable if it does not exist an infinite sequence of $r$-reductions starting from $t$. Finally, $\Rightarrow_r$ is strongly normalizing if every $u \in \Lambda$ is strongly $r$-normalizable.

- $\Rightarrow$ is confluent if $\Rightarrow^* \cdot \Rightarrow^* \subseteq \Rightarrow^*$; $\Rightarrow^*$ is confluent. From confluence it follows that: $t \simeq_r u$ iff $t \Rightarrow^* s \Rightarrow^* u$ for some term $s$; and any $r$-normalizable term has a unique $r$-normal form.

A.2  Omitted proofs and remarks of Section 2

See p.5

Proposition 3 (Syntactic characterization on $\mathsf{sh}^\ast$-normal forms). Let $t$ be a term:

- $t$ is $\mathsf{sh}^\ast$-normal iff $t \in \Lambda_n$;
- $t$ is $\mathsf{sh}^\ast$-normal and is neither a value nor a $\beta$-redex iff $t \in \Lambda_a$.

Proof.

$\Rightarrow$: We prove the left-to-right direction of both statements simultaneously by induction on $t \in \Lambda$.

If $t$ is a value then $t \in \Lambda_n$ by definition.

Otherwise $t = us$ for some terms $u, s$. By simple inspection of the rules of $\Rightarrow_{\mathsf{sh}^\ast}$, one can deduce that $u$ and $s$ $\mathsf{sh}^\ast$-normal, $u$ is not a $\beta$-redex (otherwise $t$ would be a $\sigma_1$-redex) and if $u$ is of the shape $\lambda x.u'$ then $u'$ is $\mathsf{sh}^\ast$-normal; furthermore $t$ is neither a $\beta_r$ nor a $\sigma_3$-redex, hence there are only three possibilities:

1. $u$ is not a value: by induction hypothesis $u \in \Lambda_a$ and $s \in \Lambda_n$, therefore $t \in \Lambda_a$.
2. $u$ is not an abstraction and $s$ is not a $\beta$-redex: either $u$ is a variable or $u \in \Lambda_a$ by induction hypothesis (since $u$ is neither a value nor a $\beta$-redex). If $s$ is not a value then $s \in \Lambda_a$ by induction hypothesis, so $t \in \Lambda_a$ because $t$ is either of the form $xa$ either of the form $a'a$ (with $\Lambda_a \subseteq \Lambda_n$). Otherwise $s$ is a value, thus $t \in \Lambda_a$ since $t$ is either of the form $xv$ either of the form $av$ (with $\Lambda_v \subseteq \Lambda_n$).
3. $s$ is neither a value nor a $\beta$-redex: by induction hypothesis $s \in \Lambda_a$; if $u$ is a variable then $t \in \Lambda_a$ because $t$ is of the form $xa$; if $u$ is an abstraction then $u = \lambda x.u'$ where $u'$ is $\mathsf{sh}^\ast$-normal, so $u' \in \Lambda_n$ by induction hypothesis and thus $t \in \Lambda_a$ since $t$ is of the form $(\lambda x.n)a$; finally, if $u$ is not a value then $u \in \Lambda_a$ by induction hypothesis, hence $t \in \Lambda_a$ because $t$ is of the form $a'a$ (with $\Lambda_a \subseteq \Lambda_n$).

$\Leftarrow$: The second statement follows from the first one, since $\Lambda_a \subseteq \Lambda_n$ and if $t \in \Lambda_a$ then $t$ is neither a value nor a $\beta$-redex. We prove the first statement by induction on $t \in \Lambda_n$.

If $t$ is a value then $t$ is $\mathsf{sh}^\ast$-normal (no rule of $\Rightarrow_{\mathsf{sh}^\ast}$ can be applied to $t$).
If \( t = xv \) for some variable \( x \) and value \( v \) then \( x \) and \( v \) are sh\(^{\downarrow}\)-normal and \( xv \) is not a sh-redex; therefore \( t \) is sh\(^{\downarrow}\)-normal.

If \( t = xa \) for some variable \( x \) and term \( a \in \Lambda_a \subseteq \Lambda_n \), then \( x \) and (by induction hypothesis) \( a \) are sh\(^{\downarrow}\)-normal, moreover \( a \) is not a \( \beta \)-redex (so \( t \) is not a \( \sigma_3 \)-redex), thus \( t \) is sh\(^{\downarrow}\)-normal.

If \( t = an \) or \( t = (\lambda x. n)a \) for some \( a \in \Lambda_a \subseteq \Lambda_n \) and \( n \in \Lambda_n \), then \( a \) and \( n \) are sh\(^{\downarrow}\)-normal by induction hypothesis; moreover \( a \) is neither a value nor a \( \beta \)-redex, thus \( t \) is not a sh-redex; therefore \( t \) is sh\(^{\downarrow}\)-normal. \( \square \)

**Corollary 4** (Syntactic characterization of closed sh\(^{\downarrow}\)-normal forms). Let \( t \) be a closed term: \( t \) is sh\(^{\downarrow}\)-normal iff \( t \) is a value iff \( t = \lambda x. u \) for some term \( u \) with \( \text{fv}(u) \subseteq \{x\} \).

**Proof.** By Prop. 3 and since \( t \) is closed, \( t \) is sh\(^{\downarrow}\)-normal iff \( t \) is a value (all terms in \( \Lambda_a \) are open). Since \( t \) is closed and variables are open, \( t \) is a value iff \( t = \lambda x. u \) for some \( u \) with \( \text{fv}(u) \subseteq \{x\} \). \( \square \)

### A.3 Omitted proofs and remarks of Section 3

**Lemma 26** (Free variables in environment). If the judgment \( \Gamma \vdash t : P \) is derivable then \( \text{fv}(t) \subseteq \text{dom}(\Gamma) \) and moreover \( x \in \text{dom}(\Gamma) \setminus \text{fv}(t) \) then \( \Gamma(x) = [] \).

**Proof.** By straightforward induction on \( t \in \Lambda \). \( \square \)

**Remark 27.** If \( t \) is an application and \( \pi \vdash t \), then \( |\pi| > 0 \).

**Lemma 6** (Judgment decomposition for values). Let \( v \in \Lambda_v \), \( \Delta \) be an environment, and \( P_1, \ldots, P_p \) be positive types (for some \( p \in \mathbb{N} \)). There is a derivation \( \pi \vdash \Delta \vdash v : P_1 \uplus \cdots \uplus P_p \) iff for all \( 1 \leq i \leq p \) there are an environment \( \Delta_i \) and a derivation \( \pi_i \vdash \Delta_i \vdash v : P_i \) such that \( \text{dom}(\Delta_1) = \ldots = \text{dom}(\Delta_p) = \text{dom}(\Delta) \) and \( \Delta = \biguplus_{i=1}^p \Delta_i \). Moreover, \( |\pi| = \sum_{i=1}^p |\pi_i| \).

**Proof.** Both directions are proved by cases, depending on whether \( v \) is a variable or an abstraction.

\( \Rightarrow: \) If \( v = y \), then the last rule of \( \pi \) is \( \text{ax} \) and thus \( \Delta = x_1 : [], \ldots, x_m : [], y : P_1 \uplus \cdots \uplus P_p \). So, for all \( 1 \leq i \leq p \), there are an environment \( \Delta_i = x_1 : [], \ldots, x_m : [], y : P_i \) and a derivation \( \pi_i = \Delta_i \vdash v : P_i \vdash \text{ax} \)

with \( \biguplus_{i=1}^p \Delta_i = \Delta \) and \( |\pi| = 0 = \sum_{i=1}^p |\pi_i| \).

If \( v = \lambda x.t \) then the last rule of \( \pi \) is \( \lambda \), so there are \( n \in \mathbb{N} \), positive types \( Q_1, Q_1', \ldots, Q_n, Q_n' \), environments \( \Gamma_1, \ldots, \Gamma_n \) such that \( \Delta = \biguplus_{i=1}^n \Gamma_i \), \( \biguplus_{i=1}^{n-1} P_i = [Q_1 \rightarrow Q_1', \ldots, Q_n \rightarrow Q_n'] \) and

\[
\pi = \frac{\vdash \pi_1'}{\Gamma_1, x : Q_1 \vdash t : Q_1'} \quad \frac{\vdash \pi_n'}{\Gamma_n, x : Q_n \vdash t : Q_n'}
\]

Thus, up to renumbering the \( Q_i \)'s and \( Q_i' \)'s, there are environments \( \Delta_1, \ldots, \Delta_p \), derivations \( \pi_1, \ldots, \pi_p \) and integers \( k_1 = 1 \leq k_2 \leq \cdots \leq k_p \leq k_{p+1} = p \), such that, for all \( 1 \leq i \leq p \), \( P_i = \biguplus_{j=k_i}^{k_{i+1}-1} [Q_j \rightarrow Q_j'] \) and

\[
\pi_i = \frac{\vdash \pi_{k_i}'}{\Gamma_{k_i}, x : Q_{k_i} \vdash t : Q_{k_i}'} \quad \frac{\vdash \pi_{k_i+1}'}{\Gamma_{k_i+1}, x : Q_{k_i+1} \vdash t : Q_{k_i+1}'}
\]

with \( \Delta_i = \biguplus_{j=k_i}^{k_{i+1}-1} \Gamma_j \).
where \(|\pi| = \sum_{j=k_i}^{k_i+1} |\pi'_j|\), hence \(|\pi| = \sum_{i=1}^{n} |\pi'_j| = \sum_{j=k_i}^{k_i+1} |\pi'_j| = \sum_{i=1}^{n} |\pi_i|\).

\[\pi = \frac{\Delta \vdash v : \bigcup_{i=1}^{p} P_i}{\Delta \vdash v : \bigcup_{i=1}^{p} P_i}^{\text{ax}}\]

where \(|\pi| = 0 = \sum_{i=1}^{p} |\pi_i|\).

\[\pi_i = \frac{\vdash : \pi_{1i}}{\Delta_i \vdash x : Q_{ij} \vdash t : Q'_{ij}}^{\text{Minimal derivation for values}} \frac{\vdash : \pi_{ki}}{\Delta_i \vdash x : Q_{ij} \vdash t : Q'_{ij}}^{\text{Weakening}} \lambda \]

where \(|\pi| = \sum_{j=1}^{k_i} |\pi_{ij}|\).

Corollary 28 (Minimal derivation for values). For any \(v \in \Lambda_v\) and any pairwise distinct variables \(x_1, \ldots, x_n\) with \(\text{fv}(v) \subseteq \{x_1, \ldots, x_n\}\), there is a \(\pi \vdash x_1 : [], \ldots, x_n : [] \vdash v : []\) with \(|\pi| = 0\).

Proof. Apply the right-to-left direction of Lemma 6 taking \(p = 0\). \(\square\)

Lemma 29 (Weakening). Let \(t \in \Lambda\) and \(\Gamma\) be an environment such that \(x \notin \text{dom}(\Gamma)\).

1. Adding weakening: If \(\pi \vdash \Gamma \vdash t : P\) then there is \(\pi' \vdash \Gamma, x : [] \vdash t : P\) such that \(|\pi| = |\pi'|\).

2. Removing weakening: If \(\pi' \vdash \Gamma, x : [] \vdash t : P\) and \(x \notin \text{fv}(t)\), then there exists \(\pi \vdash \Gamma \vdash t : P\) such that \(|\pi| = |\pi'|\).

Proof. Both points are proved by straightforward induction on \(t \in \Lambda\). \(\square\)

Lemma 29 means that, given two judgments \(\Gamma \vdash t : P\) and \(\Delta \vdash u : Q\), we can suppose without loss of generality that \(\text{dom}(\Gamma) = \text{dom}(\Delta)\). In Lemma 29(1) the hypothesis \(x \notin \text{dom}(\Gamma)\) implies \(x \notin \text{fv}(t)\), by Lemma 26. Lemma 29(2)—a weakened converse of Lemma 29(1)—fails when \(x \in \text{fv}(t)\), indeed the judgment \(x : [] \vdash x : []\) is derivable but the judgment \(\vdash x : []\) is not.

See p.6 Lemma 7 (Substitution). Let \(t \in \Lambda\) and \(v \in \Lambda_v\). If \(\pi \vdash \Gamma, x : P \vdash t : Q\) and \(\pi' \vdash \Delta \vdash v : P\) where \(\text{dom}(\Gamma) = \text{dom}(\Delta)\), then there exists \(\pi'' \vdash \Gamma \cup \Delta \vdash t\{v/x\} : Q\) such that \(|\pi''| = |\pi| + |\pi'|\).
Proof. By induction on \( t \in \Lambda \).

If \( t = x \), then \( t\{v/x\} = v \) and the last rule of \( \pi \) is \( \text{ax} \) with \( P = Q \) and \( \Gamma = y_1 : [\cdot], \ldots, y_n : [\cdot] \) (\( y_i \neq x \) for all \( 1 \leq i \leq n \)), whence \( \Gamma \cup \Delta = \Delta \) and \( |\pi| = 0 \). We conclude by setting \( \pi'' = \pi' \).

If \( t = y \neq x \), then \( t\{v/x\} = y \) and the last rule of \( \pi \) is \( \text{ax} \) with \( P = [\cdot] \) (since \( x \neq y \)), whence \( |\pi| = 0 \) and \( \Gamma = z_1 : [\cdot], \ldots, z_k : [\cdot] : y. Q \). By Lemma 6 from \( \Delta \vdash v :: [\cdot] \) it follows that \( |\pi'| = 0 \) and (since \( \text{dom}(\Gamma) = \text{dom}(\Delta) \)) \( \Delta = z_1 : [\cdot], \ldots, z_k : [\cdot] : y. [\cdot] \), therefore \( \Gamma \cup \Delta = \Gamma \). So, the derivation

\[
\pi'' = \Gamma \vdash y : Q \quad \text{ax}
\]

has conclusion \( \Gamma \cup \Delta \vdash t\{v/x\} : Q \) and is such that \( |\pi''| = 0 = |\pi| + |\pi'| \).

If \( t = us \), then \( t\{v/x\} = u\{v/x\} \) and

\[
\begin{align*}
\pi &= \Gamma_1, x : P_1 \vdash u : [Q_2 \rightarrow Q] & \Gamma_2, x : P_2 \vdash s : Q_2 \\
\end{align*}
\]

with \( |\pi| = |\pi_1| + |\pi_2| + 1 = |\pi_1| + |\pi_1'| + |\pi_2| + 1 = |\pi| + |\pi'| + 1 \).

If \( t = \lambda y. u \), then we can suppose without loss of generality that \( y \notin \text{fv}(v) \cup \{x\} \cup \text{dom}(\Delta) \), therefore \( t\{v/x\} = \lambda y. u\{v/x\} \) and there are \( n \in \mathbb{N} \), environments \( \Gamma_1, \ldots, \Gamma_n \), positive types \( P_1, Q_1, Q_1', \ldots, P_n, Q_n, Q_n' \) such that \( \Gamma = \bigcup_{i=1}^{n} \Gamma_i, P = \bigcup_{i=1}^{n} P_i, Q = \bigcup_{i=1}^{n} (Q_i \rightarrow Q_i') \) and

\[
\begin{align*}
\pi &= \Gamma_1, y : Q_1, x : P_1 \vdash u : Q_1 \\
&\quad \ldots \\
&\quad \Gamma_n, y : Q_n, x : P_n \vdash u : Q_n' \\
\end{align*}
\]

with \( |\pi| = \sum_{i=1}^{n} |\pi_i| \). According to Lemma 6, since \( y \notin \text{dom}(\Delta) \), there is a derivation \( \pi_0 \) with conclusion \( \Delta, y : [\cdot] \vdash v : P \) such that \( |\pi_0| = |\pi'| \). By applying Lemma 6 to \( \pi_0 \) (as \( P = \bigcup_{i=1}^{n} P_i \)), for all \( 1 \leq i \leq n \) there are an environment \( \Delta_i \) and a derivation \( \pi_i \vdash \Delta_i, y : [\cdot] \vdash v : P_i \) such that \( \Delta = \bigcup_{i=1}^{n} \Delta_i \) and \( |\pi_i| = \sum_{i=1}^{n} |\pi_i'| \). By induction hypothesis, for all \( 1 \leq i \leq n \), there is a derivation \( \pi''_i \vdash \Gamma_i \cup \Delta_i, y : Q_i \vdash u\{v/x\} : Q_i' \) such that \( |\pi''_i| = |\pi_i| + |\pi_i'| \). Since \( \Gamma \cup \Delta = \bigcup_{i=1}^{n} \Gamma_i \cup \Delta_i \), there is a derivation

\[
\begin{align*}
\pi'' &= \Gamma_1 \cup \Delta_1, y : Q_1 \vdash u\{v/x\} : Q_1' \\
&\quad \ldots \\
&\quad \Gamma_n \cup \Delta_n, y : Q_n \vdash u\{v/x\} : Q_n' \\
\end{align*}
\]

where \( |\pi''| = \sum_{i=1}^{n} |\pi''_i| = \sum_{i=1}^{n} |\pi_i| + \sum_{i=1}^{n} |\pi_i'| = |\pi| + |\pi_0'| = |\pi| + |\pi'| \).

\( \square \)
Proposition 8 (Quantitative balanced subject reduction). Let $t, t' \in \Lambda$ and $\pi \vdash \Gamma \vdash t : Q$.

1. Shrinkage under $\beta_v$-step: If $t \rightarrow^{\beta_v} t'$ then $|\pi| > 0$ and there exists a derivation $\pi'$ with conclusion $\Gamma \vdash t' : Q$ such that $|\pi'| = |\pi| - 1$.

2. Size invariance under $\sigma^p$-step: If $t \rightarrow^\sigma t'$ then $|\pi| > 0$ and there exists a derivation $\pi'$ with conclusion $\Gamma \vdash t' : Q$ such that $|\pi'| = |\pi|$.

Proof. 1. Since $t$ is not $\beta_v$-normal, $t$ is not a value and thus $|\pi| > 0$ according to Remark 27. The proof that there exists a derivation $\pi' \vdash \Gamma \vdash t' : Q$ such that $|\pi'| = |\pi| - 1$ is by induction on $t \in \Lambda$.

Cases:

- **Step at the root**, i.e. $t = (\lambda x.u)v \rightarrow^{\beta_v} u\{v/x\} = t'$: then,

$$
\pi = \frac{\pi_1}{\Gamma_1, x : P \vdash u : Q} \quad \frac{\pi_2}{\Gamma_2 \vdash v : P} \quad \frac{\lambda \pi_1}{\Gamma_1 \vdash \lambda x. u : [P \rightarrow Q]} \quad \frac{\pi_2}{\Gamma_2 \vdash t : Q} \quad \frac{\pi}{\Gamma \vdash t : Q}
$$

where $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi| = |\pi_1| + |\pi_2| + 1$. By the substitution lemma (Lemma 7), there exists a derivation $\pi' \vdash \Gamma \vdash t' : Q$ such that $|\pi'| = |\pi_1| + |\pi_2| = |\pi| - 1$.

- **Application Left**, i.e. $t = us \rightarrow^{\beta_v} u's = t'$ with $u \rightarrow^{\beta_v} u'$: then,

$$
\pi = \frac{\pi_1}{\Gamma_1 \vdash u : [P \rightarrow Q]} \quad \frac{\pi_2}{\Gamma_2 \vdash s : P} \quad \frac{\pi}{\Gamma \vdash t : Q} \quad \frac{\pi_1'}{\Gamma_1 \vdash u' : [P \rightarrow Q]} \quad \frac{\pi_2}{\Gamma_2 \vdash s : P} \quad \frac{\pi}{\Gamma \vdash t' : Q}
$$

where $|\pi'| = |\pi_1'| + |\pi_2| + 1 = |\pi_1| + |\pi_2| + 1 - 1 = |\pi| - 1$.

- **Application Right**, i.e. $t = us \rightarrow^{\beta_v} us' = t'$ with $s \rightarrow^{\beta_v} s'$: then,

$$
\pi = \frac{\pi_1}{\Gamma_1 \vdash u : [P \rightarrow Q]} \quad \frac{\pi_2}{\Gamma_2 \vdash s : P} \quad \frac{\pi}{\Gamma \vdash t : Q} \quad \frac{\pi_1'}{\Gamma_1 \vdash u' : [P \rightarrow Q]} \quad \frac{\pi_2'}{\Gamma_2 \vdash s' : P} \quad \frac{\pi}{\Gamma \vdash t' : Q}
$$

where $|\pi'| = |\pi_1| + |\pi_2'| + 1 = |\pi_1| + |\pi_2| + 1 - 1 = |\pi| - 1$.
• Step inside a \( \beta \)-redex, i.e. \( t = (\lambda x.u)s \rightarrow_{\beta} (\lambda x.u')s = t' \) with \( u \rightarrow_{\beta} u' \): then,

\[
\pi = \frac{\pi_1}{\Gamma_1, x : P \vdash u : Q} \quad \frac{\pi_2}{\Gamma_2 \vdash s : P} \quad \frac{\Gamma \vdash t : Q}{\Gamma \vdash t' : Q}
\]

where \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) \) and \( \pi = |\pi_1| + |\pi_2| + 1 > 0 \). By induction hypothesis, there exists a derivation \( \pi' \vdash \Gamma_1 \vdash t' : [P \rightarrow Q] \) such that \( |\pi'| = |\pi_1| - 1 \). Therefore, there is a derivation

\[
\pi' = \frac{\pi_1}{\Gamma_1, x : P \vdash u' : Q} \quad \frac{\pi_2}{\Gamma_2 \vdash s : P} \quad \frac{\Gamma \vdash t' : Q}{\Gamma \vdash t' : Q}
\]

where \( |\pi'| = |\pi_1| + |\pi_2| + 1 = |\pi_1| + |\pi_2| + 1 - 1 = |\pi| - 1 \).

2. Since \( t \) is not \( \sigma \)-normal, \( t \) is not a value and thus \( |\pi| > 0 \) according to Remark 27. The proof that there exists a derivation \( \pi' \) with conclusion \( \Gamma \vdash t' : Q \) such that \( |\pi'| = |\pi| \) is by induction on \( t \in \Lambda \).

Cases:

• Step at the root: there are two sub-cases:
  - \( t = (\lambda x.u)s r \mapsto_{\sigma_1} (\lambda x.ur)s = t' \) with \( x \notin \text{fv}(r) \): then,

\[
\pi = \frac{\pi_1}{\Gamma_1, x : P \vdash u : Q' \rightarrow Q} \quad \frac{\pi_2}{\Gamma_2 \vdash s : P} \quad \frac{\Gamma_3 \vdash r : Q'}{\Gamma_3 \vdash t : [Q' \rightarrow Q]} \quad \frac{\Gamma \vdash t : Q}{\Gamma \vdash t : Q}
\]

with \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) = \text{dom}(\Gamma_3) \), \( \Gamma = \Gamma_1 \uplus \Gamma_2 \uplus \Gamma_3 \) and \( |\pi| = |\pi_1| + |\pi_2| + |\pi_3| + 2 \). According to Lemma 26 and Lemma 27, we can suppose without loss of generality that \( x \notin \text{dom}(\Gamma_3) \), since \( x \notin \text{fv}(r) \). By Lemma 29, there exists \( \pi_3' \vdash \Gamma_3, x : [] \vdash r : Q' \) such that \( |\pi_3'| = |\pi_3| \). Therefore, there is a derivation

\[
\pi' = \frac{\pi_1}{\Gamma_1, x : P \vdash u : Q' \rightarrow Q} \quad \frac{\pi_3'}{\Gamma_3, x : [] \vdash r : Q'} \quad \frac{\Gamma_2 \vdash s : P}{} \quad \frac{\Gamma_3 \vdash t : Q'}{\Gamma \vdash t' : Q}
\]

where \( |\pi'| = |\pi_1| + |\pi_3'| + 1 + |\pi_2| + 1 = |\pi_1| + |\pi_3| + 1 + |\pi_2| + 1 = |\pi| \).

- \( t = v((\lambda x.u)s) r \mapsto_{\sigma_3} (\lambda x.uv)s = t' \) with \( x \notin \text{fv}(v) \): then,

\[
\pi = \frac{\pi_2}{\Gamma_2, x : P \vdash u' : Q'} \quad \frac{\pi_3'}{\Gamma_3 \vdash s : P} \quad \frac{\Gamma \vdash t : Q}{\Gamma \vdash t : Q}
\]
with $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) = \text{dom}(\Gamma_3)$, $\Gamma = \Gamma_1 \uplus \Gamma_2 \uplus \Gamma_3$ and $|\pi| = |\pi_1| + |\pi_2| + |\pi_3| + 1$. According to Lemma 26 and Lemma 29, we can suppose without loss of generality that $x \notin \text{dom}(\Gamma_1)$, since $x \notin \text{fv}(v)$. By Lemma 29, there exists $\pi'_1 \triangleright \Gamma_1, x: [\cdot] \vdash [Q' \rightarrow Q]$ such that $|\pi'_1| = |\pi_1|$. So, there is a derivation

$$
\pi' = \frac{\pi_1' \triangleright \Gamma_1, x: [\cdot] \vdash [Q' \rightarrow Q]}{\text{dom} \pi_1'} \quad \frac{\pi_3 \triangleright \Gamma_3, x: P \vdash u: Q' \quad \Gamma_1 \uplus \Gamma_3, x: P \vdash vu: Q}{\text{dom} \pi_3} \quad \frac{\pi_2 \triangleright \Gamma_2 \vdash s: P \quad \Gamma_1 \uplus \Gamma_3 \vdash \lambda x. u': [P \rightarrow Q]}{\text{dom} \pi_2}
$$

where $|\pi' stan| = |\pi'_1| + |\pi_3| + 1 + |\pi_2| + 1 = |\pi_1| + |\pi_3| + 1 + |\pi_2| + 1 = |\pi|$.

- **Application Left**, i.e. $t = us \rightarrow_{\sigma'} us' = t'$ with $u \rightarrow_{\sigma'} u'$: then,

$$
\pi = \frac{\pi_1 \triangleright \Gamma_1 \vdash u: [P \rightarrow Q] \quad \Gamma_2 \vdash s: P}{\text{dom} \pi_1} \quad \frac{\pi_2 \triangleright \Gamma_2 \vdash t: Q}{\text{dom} \pi_2}
$$

where $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi| = |\pi_1| + |\pi_2| + 1$. By induction hypothesis, there exists a derivation $\pi'_1 \triangleright \Gamma_1 \vdash u': [P \rightarrow Q]$ such that $|\pi'_1| = |\pi_1|$. Therefore, there is a derivation

$$
\pi' = \frac{\pi' \triangleright \Gamma_1 \vdash u': [P \rightarrow Q] \quad \Gamma_2 \vdash s: P}{\text{dom} \pi'}
$$

where $|\pi' stan| = |\pi'_1| + |\pi_2| + 1 = |\pi_1| + |\pi_2| + 1 = |\pi|$.

- **Application Right**, i.e. $t = us \rightarrow_{\sigma'} us' = t'$ with $s \rightarrow_{\sigma'} s'$: then,

$$
\pi = \frac{\pi_1 \triangleright \Gamma_1 \vdash u: [P \rightarrow Q] \quad \Gamma_2 \vdash s: P}{\text{dom} \pi_1} \quad \frac{\pi_2 \triangleright \Gamma_2 \vdash t: Q}{\text{dom} \pi_2}
$$

where $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi| = |\pi_1| + |\pi_2| + 1$. By induction hypothesis, there exists a derivation $\pi'_2 \triangleright \Gamma_2 \vdash u': Q$ such that $|\pi'_2| = |\pi_2|$. Therefore, there is a derivation

$$
\pi' = \frac{\pi' \triangleright \Gamma_1 \vdash u: [P \rightarrow Q] \quad \Gamma_2 \vdash s: P}{\text{dom} \pi'}
$$

where $|\pi' stan| = |\pi_1| + |\pi'_2| + 1 = |\pi_1| + |\pi_2| + 1 = |\pi|$.

- **Step inside a $\beta$-redex**, i.e. $t = (\lambda x. u)s \rightarrow_{\sigma'} (\lambda x. u')s = t'$ with $u \rightarrow_{\sigma'} u'$: then,

$$
\pi = \frac{\pi_1 \triangleright \Gamma_1, x: P \vdash u: Q}{{\text{dom} \pi_1}} \quad \frac{\pi_2 \triangleright \Gamma_2 \vdash \lambda x. u': [P \rightarrow Q] \quad \Gamma_2 \vdash s: P}{\text{dom} \pi_2}
$$
where \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) \), \( \Gamma = \Gamma_1 \uplus \Gamma_2 \) and \( |\pi| = |\pi_1| + |\pi_2| + 1 \). By induction hypothesis, there exists a derivation \( \pi'_1 \triangleright \Gamma_1 \vdash u' : [P \rightarrow Q] \) such that \( |\pi'_1| = |\pi_1| \). Therefore, there is a derivation

\[
\pi' = \frac{\pi'_1}{\frac{\Gamma_1 \vdash \lambda x.u' : [P \rightarrow Q]}{\Gamma_1 \vdash \lambda x.u' : [P \rightarrow Q] \vdash s : P}} \triangleq \pi_2
\]

where \( |\pi'| = |\pi'_1| + |\pi_2| + 1 = |\pi_1| + |\pi_2| + 1 = |\pi| \). □

**Lemma 9** (Abstraction commutation).

1. **Abstraction vs. abstraction:** Let \( n \in \mathbb{N} \). If \( \pi \triangleright \Delta \vdash \lambda y.(\lambda x.t)v : \bigcup^n_{i=1} [P'_i \rightarrow P_i] \) and \( y \notin \text{fv}(v) \), then there is \( \pi' \triangleright \Delta \vdash (\lambda x.\lambda y.t)v : \bigcup^n_{i=1} [P'_i \rightarrow P_i] \) such that \( |\pi'| = |\pi| + 1 - n \).

2. **Application vs. abstraction:** If \( \pi \triangleright \Delta \vdash ((\lambda x.t)v)((\lambda x.u)v) : P \) then there exists a derivation \( \pi' \triangleright \Delta \vdash (\lambda x.tu)v : P \) such that \( |\pi'| = |\pi| - 1 \).

**Proof.**

1. We can suppose without loss of generality that \( x, y \notin \text{dom}(\Delta) \). Therefore, there are derivations \( \pi'_1, \pi'_2, \ldots, \pi'_{n}, \pi''_1, \pi''_2 \), environments \( \Gamma'_1, \Gamma'_2, \ldots, \Gamma'_n, \Delta_i \) (with \( \text{dom}(\Delta) = \text{dom}(\Gamma'_1) = \text{dom}(\Delta_i) \) for all \( 1 \leq i \leq n \)) and positive types \( Q_i, \ldots, Q_n \) such that

\[
\pi = \frac{\Gamma'_1, y : P'_i, x : Q_i \vdash t : P_i}{\Gamma'_1, \vdash \Delta_i \vdash v : Q_i} \triangleq \pi''_1 \quad \text{(for all } 1 \leq i \leq n \text{)}
\]

where \( \Delta = \bigcup^n_{i=1} \Gamma_i \uplus \Delta_i \) and \( |\pi| = \sum^n_{i=1} (|\pi'_i| + |\pi''_i| + 1) = n + \sum^n_{i=1} (|\pi'_i| + |\pi''_i|) \). According to Lemma 8, there exists \( \pi'_2 \triangleright \Delta' \vdash v : \bigcup^n_{i=1} Q_i \) with \( \Delta' = \bigcup^n_{i=1} \Delta_i \) (whence \( \Delta = \Delta' \uplus \bigcup^n_{i=1} \Gamma_i \)) and \( |\pi_2| = \sum^n_{i=1} |\pi''_i| \), thus one has

\[
\pi' = \frac{\Delta \vdash (\lambda x.\lambda y.t)v : \bigcup^n_{i=1} [P'_i \rightarrow P_i]}{\Delta' \vdash v : \bigcup^n_{i=1} Q_i} \triangleq \pi_2
\]

where \( |\pi'| = |\pi_2| = |\pi| + 1 + |\sum^n_{i=1} |\pi'_i|| = 1 + |\sum^n_{i=1} (|\pi'_i| + |\pi''_i|)| = |\pi| - n \). □

2. We can suppose without loss of generality that \( x \notin \text{dom}(\Delta) \). Therefore, there are environments \( \Delta_1, \Delta_2, \Delta_3, \Delta_4 \), positive types \( Q, P_1, P_2 \) and derivations \( \pi_1, \pi_2, \pi_3, \pi_4 \) such that

\[
\pi = \frac{\Delta_3, x : P_1 \vdash t : [Q \rightarrow P]}{\Delta_3 \vdash (\lambda x.t)v : [Q \rightarrow P]} \triangleq \pi_1 \quad \text{and} \quad \frac{\Delta_1 \vdash v : P_1}{\Delta_1 \vdash \lambda x.u : [P_1 \rightarrow Q]} \triangleq \pi_2
\]

where \( \Delta = \lambda \lambda \lambda \vdash (\lambda x.t)v : ([Q \rightarrow P] \vdash [Q \rightarrow P]) \).
where $\Delta = \Delta_1 \uplus \Delta_2 \uplus \Delta_3 \uplus \Delta_4$ and $|\pi| = |\pi_1| + |\pi_2| + |\pi_3| + |\pi_4| + 3$. According to Lemma 5 there is $\pi_0 \vdash \Delta_1 \uplus \Delta_2 \vdash \nu: P_1 \uplus P_2$ such that $|\pi_0| = |\pi_1| + |\pi_2|$, thus there exists

$$\pi' = \begin{array}{c} \frac{\Delta_3, x: P_1 \vdash t: [Q \rightarrow P]}{\Delta_3 \uplus \Delta_4, x: P_1 \vdash tu: P} \Delta_4, x: P_2 \vdash u: Q \quad \pi_0 \\
\frac{\Delta_3 \uplus \Delta_4 \vdash \lambda x.tu: [P_1 \vdash P_2 \rightarrow Q]}{\Delta_1 \uplus \Delta_2 \vdash v: P_1 \vdash P_2} \quad \pi_0 \\
\end{array}$$

where $|\pi'| = |\pi_0| + |\pi_3| + |\pi_4| + 2 = |\pi_1| + |\pi_2| + |\pi_3| + |\pi_4| + 2 = |\pi| - 1$. \qed

See p. 22

**Proposition 10** (Quantitative balanced subject expansion). Let $t, t' \in \Lambda$ and $\pi' \vdash \Gamma \vdash t': Q$.

1. Enlargement under anti-$\beta^\nu$-step: If $t \rightarrow_{\beta^\nu} t'$ then there is $\pi \vdash \Gamma \vdash t: Q$ with $|\pi| = |\pi'| + 1$.

2. Size invariance under anti-$\sigma^\nu$-step: If $t \rightarrow_{\sigma^\nu} t'$ then $|\pi'| > 0$ and there exists $\pi \vdash \Gamma \vdash t: Q$ such that $|\pi| = |\pi'|$.

**Proof.**

1. By induction on $t \in \Lambda$. Cases:

   - **Step at the root**, i.e. $t = (\lambda x.u)v \rightarrow_{\beta^\nu} u[v/x] = t'$. We proceed by induction on $u \in \Lambda$.
     - If $u = x$, then $t' = v$ and $\pi' \vdash \Gamma \vdash v: Q$, while $t = (\lambda x.x)v$. Setting $\Gamma_0 = x_1: [], ..., x_n: []$ where $\text{dom}(\Gamma_0) = \text{dom}(\Gamma)$, we have the derivation
       
       $$\pi = \begin{array}{c} \frac{\Gamma_0, x: [Q \rightarrow Q]}{\Gamma \vdash x: Q} \owe \pi' \\
\frac{\Gamma_0 \vdash \lambda x.x: [Q \rightarrow Q]}{\Gamma \vdash t: Q} \owe \end{array}$$

       with $|\pi| = |\pi'| + 1$.

     - If $u = y \neq x$ (we can suppose without loss of generality that $x \notin \text{fv}(v)$), then $t' = y$ and $\pi' = \begin{array}{c} \frac{\Gamma_0, y: [Q \rightarrow Q]}{\Gamma_0 \vdash y: [Q \rightarrow Q]} \owe \pi' \\
\frac{\Gamma_0, y: [Q \rightarrow Q]}{\Gamma_0 \vdash \lambda x.y: [Q \rightarrow Q]} \owe \end{array}$ with $\Gamma_0 = y_1: [], ..., y_m: []$ (where $\text{dom}(\Gamma_0) = \text{fv}(v) \setminus \{y\}$), while $t = (\lambda x.y)v$. Notice that $|\pi'| = 0$. We have:
       
       $$\pi = \begin{array}{c} \frac{\Gamma_0, x: [Q \rightarrow Q]}{\Gamma \vdash x: Q} \owe \pi' \\
\frac{\Gamma_0 \vdash \lambda x.x: [Q \rightarrow Q]}{\Gamma \vdash t: Q} \owe \end{array}$$

       (notice that the rule $\lambda$ in $\pi$ has 0 premises) with $|\pi| = 1 = |\pi'| + 1$.

     - If $u = \lambda y.s$ (we can suppose without loss of generality that $y \notin \text{fv}(v) \cup \{x\}$), then $t' = \lambda y.s[v/x]$ and $t = (\lambda x.\lambda y.s)v$. As $\pi' \vdash \Gamma \vdash t': Q$, there are $n \in \mathbb{N}$, positive types $P_1, Q_1, ..., P_n, Q_n$, environments $\Gamma_1, ..., \Gamma_n$ with $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \cdots = \text{dom}(\Gamma_n)$, and derivations $\pi'_1, ..., \pi'_n$ such that $Q = [P_1 \rightarrow Q_1, ..., P_n \rightarrow Q_n]$ and
       
       $$\pi' = \begin{array}{c} \frac{\pi'_1}{\Gamma_1, y: P_1 \vdash s[v/x]: Q_1} \owe \pi'_n \\
\frac{\pi'_n}{\Gamma_n, y: P_n \vdash s[v/x]: Q_n} \owe \end{array}$$

   - **Step at the root**, i.e. $t = (\lambda x.u)v \rightarrow_{\beta^\nu} u[v/x] = t'$. We proceed by induction on $u \in \Lambda$.

   - If $u = x$, then $t' = v$ and $\pi' \vdash \Gamma \vdash v: Q$, while $t = (\lambda x.x)v$. Setting $\Gamma_0 = x_1: [], ..., x_n: []$ where $\text{dom}(\Gamma_0) = \text{dom}(\Gamma)$, we have the derivation
       
       $$\pi = \begin{array}{c} \frac{\Gamma_0, x: [Q \rightarrow Q]}{\Gamma \vdash x: Q} \owe \end{array}$$

       with $|\pi| = |\pi'| + 1$.

   - If $u = y \neq x$ (we can suppose without loss of generality that $x \notin \text{fv}(v)$), then $t' = y$ and $\pi' = \begin{array}{c} \frac{\Gamma_0, y: [Q \rightarrow Q]}{\Gamma \vdash y: [Q \rightarrow Q]} \owe \end{array}$ with $\Gamma_0 = y_1: [], ..., y_m: []$ (where $\text{dom}(\Gamma_0) = \text{fv}(v) \setminus \{y\}$), while $t = (\lambda x.y)v$. Notice that $|\pi'| = 0$. We have:
       
       $$\pi = \begin{array}{c} \frac{\Gamma_0, x: [Q \rightarrow Q]}{\Gamma \vdash x: Q} \owe \end{array}$$

       (notice that the rule $\lambda$ in $\pi$ has 0 premises) with $|\pi| = 1 = |\pi'| + 1$.

   - If $u = \lambda y.s$ (we can suppose without loss of generality that $y \notin \text{fv}(v)$), then $t' = \lambda y.s[v/x]$ and $t = (\lambda x.\lambda y.s)v$. As $\pi' \vdash \Gamma \vdash t': Q$, there are $n \in \mathbb{N}$, positive types $P_1, Q_1, ..., P_n, Q_n$, environments $\Gamma_1, ..., \Gamma_n$ with $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \cdots = \text{dom}(\Gamma_n)$, and derivations $\pi'_1, ..., \pi'_n$ such that $Q = [P_1 \rightarrow Q_1, ..., P_n \rightarrow Q_n]$ and
       
       $$\pi' = \begin{array}{c} \frac{\pi'_1}{\Gamma_1, y: P_1 \vdash s[v/x]: Q_1} \owe \pi'_n \\
\frac{\pi'_n}{\Gamma_n, y: P_n \vdash s[v/x]: Q_n} \owe \end{array}$$
where $\Gamma = \bigcup_{i=1}^{n} \Gamma_i$ and $|\pi'| = \sum_{i=1}^{n} |\pi'_i|$. Let $1 \leq i \leq n$: since $(\lambda x.s) v \rightarrow_{\beta_i} s\{v/x\}$, then by \textit{i.h.} there is $\pi_i \triangleright \Gamma, y: P_i \vdash (\lambda x.s)v: Q_i$ with $|\pi_i| = |\pi'_i| + 1$. So, we set

$$
\pi'' = \frac{\vdash \pi_1}{\vdash \pi_n} \frac{\Gamma, y: P_1 \vdash (\{s/x\})v: Q_1}{\Gamma \vdash \lambda y.(\lambda x.s)v: Q}
$$

where $|\pi''| = \sum_{i=1}^{n} |\pi_i| = \sum_{i=1}^{n} |\pi'_i| + n = |\pi'| + n$. According to Lemma $9[1]$ there is a derivation $\pi \triangleright \Gamma \vdash t: Q$ where $|\pi| = |\pi''| - n + 1 = |\pi'| + 1$.

Finally, if $u = sr$, then $t' = s\{v/x\}r\{v/x\}$ and $t = (\lambda x.s)r v$. Since $\pi' \triangleright \Gamma \vdash t': Q$, there are derivations $\pi'_1$ and $\pi'_2$, a positive type $P$, environments $\Gamma_1$ and $\Gamma_2$ (where $\text{dom}(\Gamma) = \text{dom}(\Gamma_1)$) and $\Gamma = \Gamma_1 \uplus \Gamma_2$ such that

$$
\pi' = \frac{\vdash \pi'_1}{\vdash \pi'_2} \frac{\Gamma_1 \vdash s\{v/x\}: [P \rightarrow Q]}{\Gamma_2 \vdash r\{v/x\}: P} \frac{\Gamma \vdash t': Q}{@}
$$

where $|\pi'| = |\pi'_1| + |\pi'_2| + 1$. Since $(\lambda x.s)v \rightarrow_{\beta_i} s\{v/x\}$ and $(\lambda x.r)v \rightarrow_{\beta_i} r\{v/x\}$, then by \textit{i.h.} there are $\pi_1 \triangleright \Gamma_1 \vdash (\lambda x.s)v: [P \rightarrow Q]$ and $\pi_2 \triangleright \Gamma_2 \vdash (\lambda x.r)v: P$ with $|\pi_1| = |\pi'_1| + 1$ and $|\pi_2| = |\pi'_2| + 1$. So, we set

$$
\pi'' = \frac{\vdash \pi_1}{\vdash \pi_2} \frac{\Gamma \vdash ((\lambda x.s)v)((\lambda x.r)v): Q}{@}
$$

where $|\pi''| = |\pi_1| + |\pi_2| + 1 = |\pi'_1| + |\pi'_2| + 3 = |\pi'| + 2$. According to Lemma $9[2]$ there is a derivation $\pi \triangleright \Gamma \vdash t: Q$ with $|\pi| = |\pi''| - 1 = |\pi'| + 1$.

- \textbf{Application Left, i.e. $t = us \rightarrow_{\beta_i} u's = t'$ with $u \rightarrow_{\beta_i} u'$}: then,

$$
\pi' = \frac{\vdash \pi'_1}{\vdash \pi'_2} \frac{\Gamma_1 \vdash u': [P \rightarrow Q]}{\Gamma_2 \vdash s: P} \frac{\Gamma \vdash t': Q}{@}
$$

where $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi'| = |\pi'_1| + |\pi'_2| + 1$. By induction hypothesis, there is $\pi_1 \triangleright \Gamma \vdash u: [P \rightarrow Q]$ with $|\pi_1| = |\pi'_1| + 1$. So, there is

$$
\pi = \frac{\vdash \pi_1}{\vdash \pi_2} \frac{\Gamma \vdash t: Q}{@}
$$

where $|\pi| = |\pi_1| + |\pi_2| + 1 = |\pi'_1| + 1 + |\pi'_2| + 1 = |\pi'| + 1$.

- \textbf{Application Right, i.e. $t = us \rightarrow_{\beta_i} us' = t'$ with $s \rightarrow_{\beta_i} s'$}: then,

$$
\pi' = \frac{\vdash \pi'_1}{\vdash \pi'_2} \frac{\Gamma_1 \vdash u: [P \rightarrow Q]}{\Gamma_2 \vdash s': P} \frac{\Gamma \vdash t': Q}{@}
$$

where $|\pi| = |\pi'_1| + |\pi'_2| + 1 = |\pi'_1| + 1 + |\pi'_2| + 1 = |\pi'| + 1$. 


Towards a Semantic Measure of the Execution Time in Call-by-Value lambda-Calculus

where \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) \), \( \Gamma = \Gamma_1 \uplus \Gamma_2 \) and \( \vert \pi \vert = \vert \pi_1 \vert + \vert \pi_2 \vert + 1 \). By induction hypothesis, there exists \( \pi_2 \vdash \Gamma_2 \vdash s: P \) with \( \vert \pi_2 \vert = \vert \pi'_2 \vert + 1 \). So, there is

\[
\pi = \frac{\vdash \pi_1}{\Gamma_1 \vdash u: [P \rightarrow Q]} \quad \frac{\vdash \pi_2}{\Gamma_2 \vdash s: P} \quad @
\]

where \( \vert \pi \vert = \vert \pi_1 \vert + \vert \pi_2 \vert + 1 = \vert \pi_1 \vert + \vert \pi'_2 \vert + 1 + 1 = \vert \pi' \vert + 1 \).

- Step inside a \( \beta \)-redex, i.e., \( t = (\lambda x.u)s \rightarrow_{\beta} (\lambda x.u')s = t' \) with \( u \rightarrow_{\beta} u' \); then,

\[
\pi' = \frac{\vdash \pi_1}{\Gamma_1, x: P \vdash u': Q} \quad \frac{\vdash \lambda x.u': [P \rightarrow Q]}{\Gamma_1 \vdash \lambda x.u': \lfloor P \rightarrow Q \rfloor \lambda} \quad \frac{\vdash \pi_2}{\Gamma_2 \vdash s: P} \quad @
\]

where \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) \), \( \Gamma = \Gamma_1 \uplus \Gamma_2 \) and \( \vert \pi' \vert = \vert \pi_1 \vert + \vert \pi_2 \vert + 1 \). By induction hypothesis, there exists \( \pi_1 \vdash \Gamma_1, x: P \vdash Q \) with \( \vert \pi_1 \vert = \vert \pi'_1 \vert + 1 \). So, there is

\[
\pi = \frac{\vdash \pi_1}{\Gamma_1, x: P \vdash u: [Q \rightarrow O]} \quad \frac{\vdash \lambda x.u: [P \rightarrow Q]}{\Gamma_1 \vdash \lambda x.u: \lfloor P \rightarrow Q \rfloor \lambda} \quad \frac{\vdash \pi_2}{\Gamma_2 \vdash s: P} \quad @
\]

where \( \vert \pi \vert = \vert \pi_1 \vert + \vert \pi_2 \vert + 1 = \vert \pi'_1 \vert + 1 + \vert \pi_2 \vert + 1 = \vert \pi' \vert + 1 \).

2. Since \( \rightarrow_{\sigma} \) cannot reduce to a value, \( t' \) is not a value and thus \( \vert \pi' \vert > 0 \) according to Remark 27. The proof that there exists \( \pi \vdash \Gamma \vdash t: Q \) with \( \vert \pi \vert = \vert \pi' \vert \) is by induction on \( t \in \Lambda \). Cases:

- Step at the root: there are two sub-cases:
  - \( t = (\lambda x.u)s \rightarrow_{\sigma} (\lambda x.ur)s = t' \) with \( x \notin \text{fv}(r) \). We can suppose without loss of generality that \( x \notin \text{dom}(\Gamma) \). So,

\[
\pi' = \frac{\vdash \pi_1}{\Gamma_1, x: P \vdash u: [Q' \rightarrow O]} \quad \frac{\vdash \lambda x.u: [P \rightarrow Q']}{\Gamma_1 \vdash \lambda x.u: \lfloor P \rightarrow Q' \rfloor \lambda} \quad \frac{\vdash \pi_2}{\Gamma_2 \vdash s: P} \quad @
\]

with \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) = \text{dom}(\Gamma_3) \), \( \Gamma = \Gamma_1 \uplus \Gamma_2 \uplus \Gamma_3 \) and \( \vert \pi' \vert = \vert \pi_1 \vert + \vert \pi_2 \vert + \vert \pi'_3 \vert + 2 \). According to Lemma 29, since \( x \notin \text{fv}(r) \), there is a derivation \( \pi_3 \vdash \Gamma_3 \vdash r: Q' \) such that \( \vert \pi_3 \vert = \vert \pi'_3 \vert \). Therefore, there is a derivation

\[
\pi = \frac{\vdash \pi_1}{\Gamma_1, x: P \vdash u: [Q' \rightarrow O]} \quad \frac{\vdash \lambda x.u: [P \rightarrow [Q' \rightarrow O]]}{\Gamma_1 \vdash \lambda x.u: \lfloor P \rightarrow [Q' \rightarrow O] \rfloor \lambda} \quad \frac{\vdash \pi_2}{\Gamma_2 \vdash s: P} \quad \frac{\vdash \pi_3}{\Gamma_3 \vdash r: Q'} \quad @
\]

where \( \vert \pi \vert = \vert \pi_1 \vert + \vert \pi_3 \vert + \vert \pi_2 \vert + 2 = \vert \pi_1 \vert + \vert \pi'_3 \vert + \vert \pi_2 \vert + 2 = \vert \pi' \vert \).
\( t = v((\lambda x.u)s) \rightarrow_{\sigma_1} (\lambda x.vu)s = t' \) with \( x \notin \text{fv}(v) \). We can suppose without loss of generality that \( x \notin \text{dom}(\Gamma) \). Therefore,

\[
\pi' = \frac{\vdash_1, x: [] \vdash v: [Q' \rightarrow Q]}{\Gamma_1 \cup \Gamma_3, x: P \vdash u: Q'} \quad \frac{\vdash_3, x: P \vdash u: Q'}{\Gamma_1 \cup \Gamma_3, \lambda x.u: [P \rightarrow Q] \vdash s: P} \quad \vdash_2
\]

with \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) = \text{dom}(\Gamma_3) \), \( \Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \) and \( |\pi'| = |\pi_1| + |\pi_2| + |\pi_3| + 2 \). According to Lemma \([29, 2]\) since \( x \notin \text{fv}(v) \), there is a derivation \( \pi_1 \vdash \Gamma_1 \vdash v: [Q' \rightarrow Q] \) such that \( |\pi_1| = |\pi'_1| \). Thus, there is a derivation

\[
\pi = \frac{\vdash_1, x: P \vdash u: Q'}{\Gamma_2, x: P \vdash u: Q'} \frac{\vdash_3, \lambda x.u: [P \rightarrow Q'] \vdash s: P}{\Gamma_2 \cup \Gamma_3 \vdash (\lambda x.u)s: Q'} \quad \vdash_2
\]

where \( |\pi| = |\pi_1| + |\pi_3| + |\pi_2| + 2 = |\pi_1| + |\pi'_1| + |\pi_2| + |\pi_3| + 2 = |\pi'| \).

- Application Left, i.e. \( t = us \rightarrow_{\sigma'} us' = t' \) with \( u \rightarrow_{\sigma'} u' \): then,

\[
\pi' = \frac{\vdash_1, u': [P \rightarrow Q]}{\Gamma_1 \vdash u': [P \rightarrow Q]} \quad \frac{\vdash_2, s: P}{\Gamma_2 \vdash s: P} \quad \vdash_2
\]

where \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) \), \( \Gamma = \Gamma_1 \cup \Gamma_2 \) and \( |\pi'| = |\pi_1| + |\pi_2| + 1 \). By induction hypothesis, there exists \( \pi_1 \vdash \Gamma_1 \vdash u': [P \rightarrow Q] \) with \( |\pi_1| = |\pi'_1| \). So, there is

\[
\pi = \frac{\vdash_1, u: [P \rightarrow Q]}{\Gamma_1 \vdash u: [P \rightarrow Q]} \quad \frac{\vdash_2, s: P}{\Gamma_2 \vdash s: P} \quad \vdash_2
\]

where \( |\pi| = |\pi_1| + |\pi_2| + 1 = |\pi'_1| + |\pi_2| + 1 = |\pi'| \).

- Application Right, i.e. \( t = us \rightarrow_{\sigma'} us' = t' \) with \( s \rightarrow_{\sigma'} s' \): then,

\[
\pi' = \frac{\vdash_1, u: [P \rightarrow Q]}{\Gamma_1 \vdash u: [P \rightarrow Q]} \quad \frac{\vdash_2, s': P}{\Gamma_2 \vdash s': P} \quad \vdash_2
\]

where \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2) \), \( \Gamma = \Gamma_1 \cup \Gamma_2 \) and \( |\pi'| = |\pi_1| + |\pi_2| + 1 \). By induction hypothesis, there exists \( \pi_2 \vdash \Gamma_2 \vdash s: P \) such that \( |\pi_2| = |\pi'_2| \). So, there is

\[
\pi = \frac{\vdash_1, u: [P \rightarrow Q]}{\Gamma_1 \vdash u: [P \rightarrow Q]} \quad \frac{\vdash_2, s: P}{\Gamma_2 \vdash s: P} \quad \vdash_2
\]

where \( |\pi| = |\pi_1| + |\pi_2| + 1 = |\pi_1| + |\pi'_2| + 1 = |\pi'| \).
• Step inside a $\beta$-redex, i.e. $t = (\lambda x.u)s \rightarrow (\lambda x.u')s = t'$ with $u \rightarrow (\lambda x.u')$: then,
\[
\begin{align*}
\pi' = & \begin{array}{c}
\frac{\pi_1'}{\Gamma_1, x : P \vdash u' : Q} \frac{\pi_2}{\lambda \Gamma_2 \vdash s : P}
\end{array} \\
\Gamma_1 \vdash \lambda x.u' : [P \rightarrow Q] \\
\Gamma \vdash t' : Q
\end{align*}
\]
where $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \text{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi'| = |\pi_1| + |\pi_2| + 1$. By induction hypothesis, there exists $\pi_1 \triangleright \Gamma_1, x : P \vdash u : Q$ with $|\pi_1| = |\pi_1'|$. So, there is
\[
\begin{align*}
\pi = & \begin{array}{c}
\frac{\pi_1}{\Gamma_1, x : P \vdash u : Q} \frac{\pi_2}{\lambda \Gamma_2 \vdash s : P}
\end{array} \\
\Gamma_1 \vdash \lambda x.u : [P \rightarrow Q] \\
\Gamma \vdash t : Q
\end{align*}
\]
where $|\pi| = |\pi_1| + |\pi_2| + 1 = |\pi_1'| + |\pi_2| + 1 = |\pi'|$. □

See p.7 Lemma 11 (Subject reduction). Let $t, t' \in \Lambda$ and $\pi \triangleright \Gamma \vdash t : Q$.

1. Shrinkage under $\beta_r$-step: If $t \rightarrow_{\beta_r} t'$ then there is $\pi' \triangleright \Gamma \vdash t' : Q$ with $|\pi| \geq |\pi'|$.

2. Size invariance under $\sigma$-step: If $t \rightarrow_{\sigma} t'$ then there is $\pi' \triangleright \Gamma \vdash t' : Q$ such that $|\pi| = |\pi'|$.

Proof. Analogous to the proofs of Prop. 8.1-2, paying attention that now the induction hypothesis is weaker. The only novelty is the presence of the following case, since $\rightarrow_{\text{sh}}$ reduces under $\lambda$'s:

• Abstraction, i.e. $t = \lambda x.u \rightarrow_r \lambda x.u' = t'$ with $u \rightarrow_r u'$ and $r \in \{\beta_v, \sigma\}$: then,
\[
\begin{align*}
\pi = & \begin{array}{c}
\frac{\pi_1}{\Gamma_1, x : P_1 \vdash u : Q_1} \frac{\pi_n}{\Gamma_n, x : P_n \vdash u : Q_n}
\end{array} \\
\Gamma \vdash t : [P_1 \rightarrow Q_1, \ldots, P_n \rightarrow Q_n]
\end{align*}
\]
where $\text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \cdots = \text{dom}(\Gamma_n)$, $\Gamma = \biguplus_{i=1}^n \Gamma_i$ and $|\pi| = \sum_{i=1}^n |\pi_i|$. By induction hypothesis, for all $1 \leq i \leq n$ there is $\pi_i' \triangleright \Gamma_i, x : P_i \vdash u' : Q_i$ with $|\pi_i| \geq |\pi_i'|$ if $r = \beta_v$, and $|\pi_i| = |\pi_i'|$ if $r = \sigma$. So, there is
\[
\begin{align*}
\pi' = & \begin{array}{c}
\frac{\pi_1'}{\Gamma_1, x : P_1 \vdash u' : Q_1} \frac{\pi_n'}{\Gamma_n, x : P_n \vdash u' : Q_n}
\end{array} \\
\Gamma \vdash t' : [P_1 \rightarrow Q_1, \ldots, P_n \rightarrow Q_n]
\end{align*}
\]
where $|\pi'| = \sum_{i=1}^n |\pi_i'|$. Therefore, $|\pi| \geq |\pi'|$ if $r = \beta_v$, and $|\pi| = |\pi'|$ if $r = \sigma$. □

See p.2 Lemma 12 (Subject expansion). Let $t, t' \in \Lambda$ and $\pi' \triangleright \Gamma \vdash t' : Q$.

1. Enlargement under anti-$\beta_v$-step: If $t \rightarrow_{\beta_v} t'$ then there is $\pi \triangleright \Gamma \vdash t : Q$ with $|\pi| \geq |\pi'|$.

2. Size invariance under anti-$\sigma$-step: If $t \rightarrow_{\sigma} t'$ then there is $\pi \triangleright \Gamma \vdash t : Q$ such that $|\pi| = |\pi'|$.

Proof. Analogous to the proofs of Prop. 10.1-2, paying attention that now the induction hypothesis is weaker. The only novelty is the presence of the following case, since $\rightarrow_{\text{sh}}$ reduces under $\lambda$'s:
• Abstraction, i.e. \( t = \lambda x. u \rightarrow_t \lambda x. u' = t' \) with \( u \rightarrow_t u' \) and \( r \in \{ \beta, \sigma \} \): then,

\[
\pi' = \frac{\pi'_1}{\Gamma, x: P_1 \vdash u': Q_1} : \frac{\pi'_n}{\Gamma, x: P_1, \ldots, P_n \vdash u': Q_n} \quad \frac{n \in \mathbb{N}}{\Gamma \vdash t': [P_1 \rightarrow Q_1, \ldots, P_n \rightarrow Q_n] \lambda}
\]

where \( \text{dom}(\Gamma) = \text{dom}(\Gamma_1) = \cdots = \text{dom}(\Gamma_n) \), \( \Gamma = \bigcup_{i=1}^n \Gamma_i \), and \( |\pi'| = \sum_{i=1}^n |\pi'_i| \). By induction hypothesis, for all \( 1 \leq i \leq n \) there is \( \pi_i : \Gamma_i, x: P_1 \vdash u: Q_i \) with \( |\pi_i| \geq |\pi'_i| \) if \( r = \beta \), and \( |\pi_i| = |\pi'_i| \) if \( r = \sigma \). So, there is

\[
\pi = \frac{\pi_1}{\Gamma_1, x: P_1 \vdash u: Q_1} : \frac{\pi_n}{\Gamma_n, x: P_1, \ldots, P_n \vdash u: Q_n} \quad \frac{n \in \mathbb{N}}{\Gamma \vdash t: [P_1 \rightarrow Q_1, \ldots, P_n \rightarrow Q_n] \lambda}
\]

where \( |\pi| = \sum_{i=1}^n |\pi_i| \). Therefore, \( |\pi| \geq |\pi'| \) if \( r = \beta \), and \( |\pi| = |\pi'| \) if \( r = \sigma \).

A.4 Omitted proofs and remarks of Section 4

Theorem 14 (Invariance under sh-equivalence). Let \( t, u \in \Lambda \), let \( n \in \mathbb{N} \) and let \( \bar{x} = (x_1, \ldots, x_n) \) be a suitable list of variables for \( t \) and \( u \). If \( t \equiv_{sh} u \) then \([t]_{\bar{x}} = [u]_{\bar{x}}\).

Proof. Since \( t \equiv_{sh} u \), there exist \( k \in \mathbb{N} \) and \( t_0, \ldots, t_k \) such that \( t = t_0, u = t_k \) and \( t_i \rightarrow_{sh} t_{i+1} \) or \( t_{i+1} \rightarrow_{sh} t_i \), for all \( 0 \leq i < k \). Using subject reduction (Lemma 11) and subject expansion (Lemma 12), it is immediate to prove by induction on \( k \in \mathbb{N} \) that \([t]_{\bar{x}} = [u]_{\bar{x}}\).

Lemma 15 (Semantics of \( sh^\beta \)-normal forms). Let \( t \) be a term, let \( n \in \mathbb{N} \) and let \( \bar{x} = (x_1, \ldots, x_k) \) be a list of variables suitable for \( t \).

1. If \( t \in \Lambda_a \) then for every positive type \( Q \) there exist positive types \( P_1, \ldots, P_k \) and a derivation \( \pi \triangleright x_1: P_1, \ldots, x_k: P_k \vdash t: Q \).
2. If \( t \in \Lambda_n \) then there are positive types \( Q, P_1, \ldots, P_k \) and a derivation \( \pi \triangleright x_1: P_1, \ldots, x_k: P_k \vdash t: Q \).
3. If \( t \) is \( sh^\beta \)-normal then \([t]_{\bar{x}} \neq \emptyset \).

Proof. Point 3 is an immediate consequence of Point 2 via the syntactic characterization of \( sh^\beta \)-normal forms (Prop. 3).

We prove simultaneously Points 1, 2 by mutual induction on \( t \in \Lambda_a \cup \Lambda_n \).

Cases for \( t \in \Lambda_a \):

• \( t = xv \) for some variable \( x \) and value \( v \): since \( \bar{x} \) is suitable for \( t \), one has \( x = x_i \) for some \( 1 \leq i \leq k \). According to Cor. 28 there exists a derivation \( \pi' \triangleright x_1: [], \ldots, x_k: [] \vdash v: [] \), so for any positive type \( Q \) there exists the derivation

\[
\pi = \frac{\pi'}{x_1: [], \ldots, x_i: [] \vdash v: []} : \frac{\pi'}{x_1: [], \ldots, x_i: [] \vdash v: []} \quad \frac{x_1: [], \ldots, x_i: [] \vdash v: []}{x_1: [], \ldots, x_i: [] \vdash v: []}
\]
• $t = xa$ for some variable $x$ and $a \in \Lambda_n$: since $\bar{x}$ is suitable for $t$, one has $x = x_i$ for some $1 \leq i \leq k$. By i.h., there exists a derivation $\pi' \triangleright x_1: P_1, \ldots, x_k: P_k \vdash a: []$ for some positive types $P_1, \ldots, P_k$. So, for any positive type $Q$ there exists a derivation

$$
\pi = \frac{x_1: [], \ldots, x_i: [[\to Q]], \ldots, x_k: [] \vdash x_j: [[\to Q]] \quad \text{ax}}{x_1: P_1, \ldots, x_i: P_i \uplus \ldots \uplus x_k: P_k \vdash t: Q} \quad \triangleright \pi'
$$

• $t = an$ for some $a \in \Lambda_a$ and $n \in \Lambda_n$: by i.h. applied to $n$, there is a derivation $\pi'' \triangleright x_1: P_1, \ldots, x_k: P_k \vdash n: P$ for some positive types $P, P_1, \ldots, P_k$. Given a positive type $Q$, by i.h. applied to $a$, there exists a derivation $\pi' \triangleright x_1: P'_1, \ldots, x_k: P'_k \vdash a: [P \to Q]$ for some positive types $P'_1, \ldots, P'_k$. So, there is a derivation

$$
\pi = \frac{x_1: P'_1, \ldots, x_i: P_i \uplus \ldots \uplus x_k: P_k \vdash a: [P \to Q] \quad x_1: P_1, \ldots, x_i: P_i \uplus \ldots \uplus x_k: P_k \vdash n: P}{x_1: P_1 \uplus P'_1, \ldots, x_i: P_i \uplus P'_i, \ldots, x_k: P_k \uplus P'_k \vdash t: Q} \quad \check{\pi'} \check{\pi''}
$$

Cases for $t \in \Lambda_n$:

• $t \in \Lambda_n$: see above.

• $t$ is a value: the statement follows from Cor. 28.

• $t = (\lambda x. a)\, n$ for some $n \in \Lambda_n$ and $a \in \Lambda_a$: by i.h. applied to $n$, there exists a derivation $\pi' \triangleright x_1: P'_1, \ldots, x_k: P'_k, x: P \vdash n: Q$ for some positive types $P'_1, \ldots, P'_k, P, Q$. By i.h. applied to $a$, there exists a derivation $\pi'' \triangleright x_1: P'_1, \ldots, x_k: P'_k \vdash a: P$ for some positive types $P'_1, \ldots, P'_k$. Therefore, there is a derivation

$$
\pi = \frac{x_1: P'_1, \ldots, x_i: P'_i, x: P \vdash n: Q \quad \lambda x. a \vdash P \to Q}{x_1: P'_1, \ldots, x_i: P'_i, x: P \vdash \lambda x. a: P} \quad \triangleright \pi' \lambda \pi''
$$

$\square$

**Theorem 16** (Semantic and logic characterization of $sh^\triangleright$-normalization). Let $t \in \Lambda$ and let $\bar{x} = (x_1, \ldots, x_n)$ be a suitable list of variables for $t$. The following are equivalent:

1. $t$ is $sh^\triangleright$-normalizable;
2. $t \simeq_{sh^\triangleright} u$ for some $sh^\triangleright$-normal $u \in \Lambda$;
3. $\llbracket t \rrbracket_{\bar{x}} \neq \emptyset$;
4. there exists a derivation $\pi \triangleright x_1: P_1, \ldots, x_n: P_n \vdash t: Q$ for some positive types $P_1, \ldots, P_n, Q$;
5. $t$ is strongly $sh^\triangleright$-normalizable.

**Proof.**

1)$\Rightarrow$ 2): Trivial, since $\to_{sh^\triangleright} \subseteq \simeq_{sh^\triangleright}$.

2)$\Rightarrow$ 3): First, note that we can suppose without loss of generality that $\bar{x}$ is suitable also for $u$. By Lemma [15], $\llbracket u \rrbracket_{\bar{x}} \neq \emptyset$. By invariance of relational semantics (Thm. [14], $\llbracket t \rrbracket_{\bar{x}} = \llbracket u \rrbracket_{\bar{x}}$.  

See p. 28
Proposition 18

Let \( t \) be a term and \( \bar{x} = (x_1, \ldots, x_k) \) be suitable for \( t \). The following are equivalent:

1. \( t \) is a value;
2. \( \pi \vdash x_1 : \ldots, x_k : [] \vdash t : [] \) and \( \pi' \vdash x_1 : \ldots, x_k : [] \vdash t : [] \), one has \( t \in \Lambda_\omega \) and \( \pi = \pi' \). More precisely, \( \pi \) consists only of the rule \( \forall x \) if \( t \) is a variable, otherwise \( t \) is an abstraction and \( \pi \) consists only of a 0-ary rule \( \lambda \).

3. There is \( \pi \vdash t \) such that \( |\pi| = 0 \).

Proof. 1. According to Prop. 18, \( t \in \Lambda_\omega \) since \( t \) is sh\( \beta \)-normal, so there are only three cases.

If \( t \) is a value, then \( |\pi| = 0 \) by the left-to-right direction of Lemma 6 (take \( p = 0 \)). Moreover, if \( t \) is a variable, then \( t = x_i \) for some \( 1 \leq i \leq k \), and the only derivation with conclusion \( x_1 : \ldots, x_k : [] \vdash t : [] \) is

\[
\pi = \frac{}{x_1 : \ldots, x_k : [] \vdash x_i : []} \frac{\forall x}{}.
\]

otherwise, \( t = \lambda x. u \) and the only derivation with conclusion \( x_1 : \ldots, x_k : [] \vdash t : [] \) is

\[
\pi = \frac{}{x_1 : \ldots, x_k : [] \vdash \lambda x. u : []} \frac{\lambda}{}.
\]

If \( t \in \Lambda_\omega \) then \( t \not\in \Lambda_\omega \) and it is impossible that \( \pi \vdash x_1 : \ldots, x_k : [] \vdash t : [] \) by Lemma 30.

Finally, if \( t \in \Lambda_\omega \cap (\Lambda_\forall \cup \Lambda_\omega) \), then \( t = (\lambda x. n) a \) for some \( n \in \Lambda_\omega \) and \( a \in \Lambda_\omega \). By necessity,

\[
\pi = \frac{\vdash \pi'}{x_1 : \ldots, x_k : [] \vdash \lambda x. n : P \to P} \frac{\vdash \pi''}{x_1 : \ldots, x_k : [] \vdash a : P} \frac{\vdash \pi''}{x_1 : \ldots, x_k : [] \vdash t : []}
\]

but it is impossible that \( \pi \vdash x_1 : \ldots, x_k : [] \vdash a : P \), according to Lemma 30. Therefore, it is impossible that \( \pi \vdash x_1 : \ldots, x_k : [] \vdash t : [] \).

2. The equivalence \( (2b) \iff (2c) \) follows immediately from Definition 13. From Lemma 17 it follows that \( (2c) \Rightarrow (2d) \). By Cor. 28, the implication \( (2a) \Rightarrow (2d) \) holds. In order to prove that \( (2d) \Rightarrow (2a) \), it is enough to notice that there is no instance of the rule \( @ \) in \( \pi \vdash t \) since \( |\pi| = 0 \), so \( t \) is either a variable or an abstraction, i.e. a value. \( \square \)

Lemma 17 (Uniqueness of the derivation with empty types; Logic and semantic characterization of values). Let \( t \in \Lambda \) be sh\( \beta \)-normal and \( \bar{x} = (x_1, \ldots, x_k) \) be suitable for \( t \). See p. 9

1. For every \( \pi \vdash x_1 : \ldots, x_k : [] \vdash t : [] \) and \( \pi' \vdash x_1 : \ldots, x_k : [] \vdash t : [] \), one has \( t \in \Lambda_\omega \) and \( \pi = \pi' \). More precisely, \( \pi \) consists only of the rule \( \forall x \) if \( t \) is a variable, otherwise \( t \) is an abstraction and \( \pi \) consists only of a 0-ary rule \( \lambda \).

2. The following are equivalent:

(a) \( t \) is a value;
(b) \( (x_1 \ldots, x_k, \lambda) \in [t]_\beta \);
(c) there is \( \pi \vdash x_1 : \ldots, x_k : [] \vdash t : [] \);
(d) there is \( \pi \vdash t \) such that \( |\pi| = 0 \).

Proof. 1. According to Prop. 17, \( t \in \Lambda_\omega \) since \( t \) is sh\( \beta \)-normal, so there are only three cases.

If \( t \) is a value, then \( |\pi| = 0 \) by the left-to-right direction of Lemma 6 (take \( p = 0 \)). Moreover, if \( t \) is a variable, then \( t = x_i \) for some \( 1 \leq i \leq k \), and the only derivation with conclusion \( x_1 : \ldots, x_k : [] \vdash t : [] \) is

\[
\pi = \frac{}{x_1 : \ldots, x_k : [] \vdash x_i : []} \frac{\forall x}{}.
\]

otherwise, \( t = \lambda x. u \) and the only derivation with conclusion \( x_1 : \ldots, x_k : [] \vdash t : [] \) is

\[
\pi = \frac{}{x_1 : \ldots, x_k : [] \vdash \lambda x. u : []} \frac{\lambda}{}.
\]

If \( t \in \Lambda_\omega \) then \( t \not\in \Lambda_\omega \) and it is impossible that \( \pi \vdash x_1 : \ldots, x_k : [] \vdash t : [] \) by Lemma 30.

Finally, if \( t \in \Lambda_\omega \cap (\Lambda_\forall \cup \Lambda_\omega) \), then \( t = (\lambda x. n) a \) for some \( n \in \Lambda_\omega \) and \( a \in \Lambda_\omega \). By necessity,
1. \( t \) is \( \text{sh}^b \)-normalizable and the \( \text{sh}^b \)-normal form of \( t \) is a value;
2. \( (([]), k, [], []) \in [t]_x \);
3. there exists \( \pi \triangleright x_1 : [], \ldots, x_k : [] \vdash t : [] \).

**Proof.** The equivalence \( (2) \Leftrightarrow (3) \) follows immediately from Definition [13]. In order to prove the equivalence \( (1) \Leftrightarrow (2) \), let us consider the two possible cases:

- either \( t \) is not \( \text{sh}^b \)-normalizable, and then \( [t]_x = \emptyset \) according to the semantic characterization of \( \text{sh}^b \)-normalization (Thm. [16]), in particular \( (([]), k, [], []) \notin [t]_x \);
- or \( t \) is \( \text{sh}^b \)-normalizable; let \( t_0 \) be its \( \text{sh}^b \)-normal form; according to the semantic characterization of values (Lemma [17]), \( (([]), k, [], []) \in [t_0]_x \) if and only if it is a value; by invariance of the semantics (Thm. [14]), \( [t]_x = [t_0]_x \); therefore, \( (([]), k, [], []) \in [t]_x \) if and only if \( t_0 \) is a value. \( \square \)

### A.5 Omitted proofs and remarks of Section 5

See p. 9

**Lemma 20** (Relationship between sizes). Let \( t \in \Lambda, k \in \mathbb{N} \) and \( \vec{x} = (x_1, \ldots, x_k) \) suitable for \( t \).

1. If \( t \) is \( \text{sh}^b \)-normal then \( |t|_b = \inf \{ |\pi| \mid \pi \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \} \).
2. If \( t \) is a value then \( |t|_b = \inf \{ |\pi| \mid \pi \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \} = 0 \).

**Proof.**

1. By Prop. [3] since \( t \) is \( \text{sh}^b \)-normal, we can proceed by induction on \( t \in \Lambda_n \). Moreover, for \( t \in \Lambda_n \) we prove also that for any positive type \( Q \) there exist positive types \( P_1, \ldots, P_k \) and a derivation \( \pi \triangleright x : P_1, \ldots, x_k : P_k \vdash t : Q \) such that \( |t|_b = |\pi| : \) this stronger statement is required to handle the case where \( t \) is a \( \text{sh}^b \)-normal \( \beta \)-redex.

If \( t \) is a value, then \( |t|_b = 0 \) by definition, and there is a derivation \( \pi \triangleright x_1 : [], \ldots, x_k : [] \vdash v : [] \) such that \( |\pi| = 0 \), according to Cor. [28]. Thus, \( |t|_b = \inf \{ |\pi| \mid \pi \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \} \).

If \( t \in \Lambda_n \), then there are three cases:

- \( t = xv \) for some variable \( x \) and value \( v \): \( |t|_b = 1, k > 0 \) and \( x = x_i \) for some \( 1 \leq i \leq k \). According to Cor. [28] there exists a derivation \( \pi' \triangleright x_1 : [], \ldots, x_k : [] \vdash v : [] \) such that \( |\pi'| = 0 \), so for any positive type \( Q \) there exists the derivation

\[
\pi = \begin{array}{c}
\vdash x_1 : [], \ldots, x_k : []
\end{array} \rightarrow Q, x_1 : [], \ldots, x_k : [] \vdash t : Q
\]

where \( |\pi| = |\pi'| + 1 = 1 = |t|_b \). The last rule of any derivation \( \pi'' \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \) is by necessity \( @ \), thus \( |\pi''| \geq 1 \) and hence \( \inf \{ |\pi''| \mid \pi'' \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \} = 1 = |t|_b \).

- \( t = xa \) for some variable \( x \) and \( a \in \Lambda_n : |a|_b = |a| + 1, k > 0 \) and \( x = x_i \) for some \( 1 \leq i \leq k \). By i.h., \( |a|_b = \inf \{ |\pi'| \mid \pi' \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash a : [] \} \) and there exists a derivation \( \pi' \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash a : [] \) for some positive types \( P_1, \ldots, P_k \) such that \( |a|_b = |\pi'| \). Therefore, for any positive type \( Q \) there exists a derivation

\[
\pi = \begin{array}{c}
\vdash x_1 : [], \ldots, x_k : []
\end{array} \rightarrow Q, x_1 : P_1, \ldots, x_k : P_k \vdash a : []
\]

where \( |\pi| = |\pi'| + 1 = |a|_b + 1 = |t|_b \). The last rule of any derivation \( \pi'' \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \) is by necessity \( @ \) having a derivation typing \( a \) as a premise, thus \( |\pi''| \geq |a|_b + 1 \) and hence \( \inf \{ |\pi''| \mid \pi'' \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash t : Q \} = |a|_b + 1 = |t|_b \).
For every reduction sequenced by induction on Proposition 21

2. By definition, if \( n \) is a value, then \( |n|_\beta = |n| \). Given a derivation \( \pi'' \vdash x_1 : P_1, \ldots, x_k : P_k \vdash \top : Q \), its last rule is by necessity \( \pi'' \vdash x_1 : P_1, \ldots, x_k : P_k \vdash \top : Q \), having derivations typing \( a \) and \( \lambda n.x \) as premises (the last rule of the latter is necessarily \( \lambda \) having a derivation typing \( n \) as unique premise), thus \( |\pi''| \geq |a| + |n| + 1 \) and hence \( \inf \{ |\pi''| \mid \pi'' \vdash x_1 : P_1, \ldots, x_k : P_k \vdash \top : Q \} = |a| + |n| + 1 = |\pi| \).

2. By definition, \( |\pi| = 0 \). According to Cor. 28, there is a derivation \( \pi \vdash x_1 : [], \ldots, x_k : [] \vdash : [] \) such that \( |\pi| = 0 \), hence \( \inf |\pi| \mathcal{E} = 0 \).

**Proposition 21** (Exact number of \( \beta^k \)-steps). Let \( t \) be a \( sh^k \)-normalizable term and \( t_0 \) be its \( sh^k \)-normal form. See p.10

For every reduction \( d : t \rightarrow^*_{sh^k} t_0 \), and every \( \pi \vdash t \) and \( \pi_0 \vdash t_0 \), one has

\[
\text{length}_{\beta^k}(d) = |\pi| \quad |t_0| = |\pi| = |\pi_0|.
\]

(3)

If moreover \( t_0 \) is a value, then \( \text{length}_{\beta^k}(d) = |\pi| \).

**Proof.** The statement concerning the case where \( t_0 \) is a value is an immediate consequence of Eq. (3) and Lemma 20. The second identity in Eq. (3) follows immediately from Lemma 20. We prove the first identity in Eq. (3) by induction on \( n = \text{length}(d) \in \mathbb{N} \).

If \( n = 0 \) then \( \text{length}_{\beta^k}(d) = 0 \) and \( t = t_0 \) (thus \( \pi = \pi_0 \)), hence \( |\pi| = |t| = |t_0| = |\pi_0| \).

If \( n > 0 \) then \( d \) has the form \( t \rightarrow_{sh^k} t' \rightarrow_{sh^k} t_0 \) for some term \( t' \); let \( d' \) be the sub-reduction sequence \( t' \rightarrow_{sh^k} t_0 \). There are two cases:
• A $\beta_v$-step at the beginning of $d$, i.e. $t \rightarrow_{\beta_v} t'$: according to the quantitative subject reduction for $\rightarrow_{\beta_v}$ (Prop. [8][1]), there is a derivation $\pi' \triangleright t'$ such that $|\pi'| = |\pi| - 2$. According to the quantitative subject expansion for $\rightarrow_{\beta_v}$ (Prop. [10][2]), for any derivation $\pi'' \triangleright t'$ there exists a derivation $\pi''' \triangleright t$ such that $|\pi'''| = |\pi''| - 2$. Therefore, from the minimality of $|\pi|$ follows the minimality of $|\pi'|$ (among the derivations $\pi'' \triangleright t$). We can then apply the i.h. to $d'$, so that $\text{leng}_{\beta_v}(d') = |\pi'| - |t_0| = |\pi| - 1 - |t_0|$, and hence $\text{leng}_{\beta_v}(d) = \text{leng}_{\beta_v}(d') + 1 = |\pi| - |t_0|$. 

• A $\sigma$-step at the beginning of $d$, i.e. $t \rightarrow_{\sigma} t'$: according to the quantitative subject reduction for $\rightarrow_{\sigma}$ (Prop. [8][2]), there is a derivation $\pi' \triangleright t'$ such that $|\pi'| = |\pi|$. According to the quantitative subject expansion for $\rightarrow_{\sigma}$ (Prop. [10][2]), for any derivation $\pi'' \triangleright t'$ there exists a derivation $\pi''' \triangleright t$ such that $|\pi'''| = |\pi''|$. Therefore, from the minimality of $|\pi|$ follows the minimality of $|\pi'|$ (among the derivations $\pi'' \triangleright t$). We can then apply the i.h. to $d'$, so that $\text{leng}_{\beta_v}(d') = \text{leng}_{\beta_v}(d'') = |\pi'| - |t_0| = |\pi| - |t_0|$. \hfill $\square$

**Lemma 30.** Let $t \in \Lambda_0$. For all $\pi \triangleright x_1 : P_1, \ldots, x_k : P_k \vdash t : Q$ there is $1 \leq i \leq k$ with $P_i \neq \emptyset$.

**Proof.** By induction on $t \in \Lambda_0$.

If $t = xu$ where $u \in \Lambda_v \cup \Lambda_0$, then $x = x_i$ for some $1 \leq i \leq k$, and hence

$$
\pi = \frac{\pi'}{\pi''} \frac{\vdash \pi''}{\vdash x : [P \rightarrow Q], \ldots, x_k : P_k' \vdash x : [P \rightarrow Q]} \frac{\vdash \pi'}{\vdash x : [P \rightarrow Q], \ldots, x_k : P_k' \vdash x : [P \rightarrow Q]} \frac{x_1 : P_1', \ldots, x_k : P_k' \vdash t : Q}{x_1 : P_1, \ldots, x_k : P_k \vdash t : Q}
$$

where $P_j = P_j' \uplus P_j''$ for all $1 \leq j \leq k$ such that $j \neq i$, and $P_i = [P \rightarrow Q] \uplus P_i'' \neq \emptyset$.

If $t = an$ for some $a \in \Lambda_d$ and $n \in \Lambda_n$, then

$$
\pi = \frac{\vdash \pi'}{\vdash \pi''} \frac{\vdash \pi''}{\vdash x_1 : P_1', \ldots, x_k : P_k' \vdash a : [P \rightarrow Q]} \frac{x_1 : P_1', \ldots, x_k : P_k' \vdash n : P}{x_1 : P_1, \ldots, x_k : P_k \vdash t : Q}
$$

where $P_j = P_j' \uplus P_j''$ for all $1 \leq j \leq k$. By i.h., there is $1 \leq i \leq k$ such that $P_i' \neq \emptyset$, thus $P_i = P_i' \uplus P_i'' \neq \emptyset$. \hfill $\square$

**Theorem 24** (Exact number of $\beta_v$-steps for valuables). Let $t \rightarrow^*_{\text{sh}} v \in \Lambda_v$. For any $x = (x_1, \ldots, x_k)$ suitable for $t$, and any $\pi \triangleright x_1 : [], \ldots, x_k : [] \vdash t : []$, one has $\text{leng}_{\beta_v}(t) = |\pi|$. 

**Proof.** Given $\pi \triangleright x_1 : [], \ldots, x_k : [] \vdash t : []$, there is $\pi_0 \triangleright x_1 : [], \ldots, x_k : [] \vdash v : []$ with $|\pi| = |\pi_0| + \text{leng}_{\beta_v}(t)$ by the quantitative subject reduction (Prop. [8][1][2]). According to Lemma [17][1], $|\pi_0| = 0$. \hfill $\square$