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

Giulio Guerrieri
Dipartimento di Informatica – Scienza 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 β_v-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 $\cap$ 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 $A \cap A = A$. This corresponds to an interpretation of a typed term $t : A \cap B$ 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 $A \cap A = A$. In a non-idempotent setting, the meaning of the typed term $t : A \cap A \cap B$ 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 $\cap$: if A and B are non-idempotent intersection types, the multiset $[A, A, B]$ represents the non-idempotent intersection type $A \cap A \cap B$.

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

Submitted to:
ITRS 2018

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, 7] — 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 $\mathbf{Rel}_!$ of the comonad ! is a CCC and then provides a denotational model of the ordinary (*i.e.* call-by-name) λ-calculus. Non-idempotent intersection types can be seen as a syntactic presentation of $\mathbf{Rel}_!$: 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, 27, 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) λ-calculus. In the second case, the construction of the relational model $\mathbf{Rel}_!$ sketched above essentially relies on Girard's call-by-name translation $(\cdot)^n$ of intuitionistic logic into LL, which decomposes the intuitionistic arrow as $(A \Rightarrow B)^n = !A^n \multimap B^n$.

Ehrhard [17] showed that the relational semantics **Rel** of LL induces also a denotational model for the *call-by-value* λ-calculus[1] 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 $(A \Rightarrow B)^v = !(A^v \multimap B^v)$. 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 λ-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 $\to_{\beta_v}$, but it is impossible to get this result in Plotkin's original call-by-value λ-calculus λ_v [35]. Indeed, the terms t and u below are β_v-normal but their semantics in the relational model are empty:

$$t := (\lambda y.\Delta)(zI)\Delta \qquad u := \Delta((\lambda y.\Delta)(zI)) \qquad (\text{where } \Delta := \lambda x.xx \text{ and } I := \lambda x.x) \qquad (1)$$

Actually, t and u should behave like the famous divergent term $\Delta\Delta$, since in λ_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 λ_v.

The reason of this mismatching is that in λ_v there are *stuck β-redexes* such as $(\lambda y.\Delta)(zI)$ in Eq. (1), *i.e.* β-redexes that β_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 β-redexes is that they may prevent the creation of other β_v-redexes, providing *"premature" β_v-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 λ_v.

[1] In call-by-value evaluation $\to_{\beta_v}$, function's arguments are evaluated before being passed to the function, so that β-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 β_v-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]), in the theory of proof assistants such as Coq (in particular, for type checking in a system based on dependent types, see [22]), or to reason about (denotational or operational) equivalences of terms in λ_v that are congruences, or about other theoretical properties of λ_v such as separability or solvability [33, 39, 3, 10].

In order to overcome this issue, we study relational semantics/non-idempotent intersection types in the *shuffling calculus* λ_{sh}, a conservative extension of Plotkin's λ_v proposed in [10] and further studied in [23, 24, 2, 25]. It keeps the same term syntax as λ_v and adds to β_v-reduction two commutation rules, σ_1 and σ_3, which "shuffle" constructors in order to move stuck β-redexes: they unblock β_v-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 $(\cdot)^v$ 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 λ_{sh}:

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. (*quantiative result*) the size of type derivations can be used to measure the execution time, *i.e.* the number of β_v-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 λ_{sh} from a given term have the same number of β_v-steps (Cor. 22); this is not obvious, as we shall explain in Ex. 23;
2. terms whose weak reduction in λ_{sh} ends in a value has an elegant semantic characterization (Prop. 18), and the number of β_v-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* λ_{sh}, namely the call-by-value λ-calculus defined in [10] and further studied in [23, 24, 2, 25]: it adds two commutation rules — the σ_1- and σ_3-reductions — to Plotkin's pure (i.e. without constants) call-by-value λ-calculus λ_v [35]. The syntax for terms of λ_{sh} is the same as Plotkin's λ_v and then the same as the ordinary (i.e. call-by-name) λ-calculus, see Fig. 1.

$$
\begin{array}{rlll}
\textit{terms}: & t,u,s ::= v \mid tu & & (\text{set: } \Lambda)\\
\textit{values}: & v ::= x \mid \lambda x.t & & (\text{set: } \Lambda_v)\\
\textit{contexts}: & C ::= \langle\cdot\rangle \mid \lambda x.C \mid Ct \mid tC & & (\text{set: } \Lambda_C)\\
\textit{Balanced contexts}: & B ::= \langle\cdot\rangle \mid (\lambda x.B)t \mid Bt \mid tB & & (\text{set: } \Lambda_B)\\
\textit{Root-steps}: & (\lambda x.t)v \mapsto_{\beta_v} t\{v/x\} & (\lambda x.t)us \mapsto_{\sigma_1} (\lambda x.ts)u,\ x\notin \mathsf{fv}(s) & v((\lambda x.s)u) \mapsto_{\sigma_3} (\lambda x.vs)u,\ x\notin \mathsf{fv}(v)\\
& \multicolumn{3}{l}{\mapsto_\sigma := \mapsto_{\sigma_1} \cup \mapsto_{\sigma_3} \qquad \mapsto_{\mathsf{sh}} := \mapsto_{\beta_v} \cup \mapsto_\sigma}\\
\mathsf{r}\textit{-reduction}: & \multicolumn{3}{l}{t \to_{\mathsf{r}} u \iff \exists C\in\Lambda_C, \exists t',u'\in\Lambda : t = C\langle t'\rangle, u = C\langle u'\rangle, t' \mapsto_{\mathsf{r}} u'}\\
\mathsf{r}^\flat\textit{-reduction}: & \multicolumn{3}{l}{t \to_{\mathsf{r}^\flat} u \iff \exists B\in\Lambda_B, \exists t',u'\in\Lambda : t = B\langle t'\rangle, u = B\langle u'\rangle, t' \mapsto_{\mathsf{r}} u'}
\end{array}
$$

Figure 1: The shuffling λ-calculus λ_{sh}

Clearly, $\Lambda_v \subsetneq \Lambda$. All terms are considered up to α-conversion (*i.e.* renaming of bound variables). The set of free variables of a term t is denoted by $\mathsf{fv}(t)$: t is *open* if $\mathsf{fv}(t) \neq \emptyset$, *closed* otherwise. Given $v \in \Lambda_v$, $t\{v/x\}$ denotes the term obtained by the *capture-avoiding substitution* of v for each free occurrence of x in the term t. Note that if $v, v' \in \Lambda_v$ then $v\{v'/x\} \in \Lambda_v$ (values are closed under substitution).

One-hole contexts C are defined as usual, see Fig. 1. We use $C\langle t\rangle$ for the term obtained by the capture-allowing substitution of the term t for the hole $\langle\cdot\rangle$ in the context C. In Fig. 1 we define also a special kind of contexts, *balanced contexts B*.

Reductions in the shuffling calculus are defined in Fig. 1 as follows: given a *root-step* rule $\mapsto_{\mathsf{r}} \subseteq \Lambda\times\Lambda$, we define the r*-reduction* $\to_{\mathsf{r}}$ (resp. $\mathsf{r}^\flat$*-reduction* $\to_{\mathsf{r}^\flat}$) as the closure of $\mapsto_{\mathsf{r}}$ under contexts (resp. balanced contexts). The $\mathsf{r}^\flat$-reduction is non-deterministic and — because of balanced contexts — can reduce under abstractions, but it is "morally" *weak*: it reduces under a λ only when the λ is applied to an argument. Clearly, $\to_{\mathsf{sh}^\flat} \subsetneq \to_{\mathsf{sh}}$ since $\to_{\mathsf{sh}}$ can freely reduce under λ's.

The root-steps used in the shuffling calculus are $\mapsto_{\beta_v}$ (the reduction rule in Plotkin's λ_v), the commutation rules $\mapsto_{\sigma_1}$ and $\mapsto_{\sigma_3}$, and $\mapsto_\sigma := \mapsto_{\sigma_1}\cup\mapsto_{\sigma_3}$ and $\mapsto_{\mathsf{sh}} := \mapsto_{\beta_v}\cup\mapsto_\sigma$. The side conditions for $\mapsto_{\sigma_1}$ and $\mapsto_{\sigma_3}$ in Fig. 1 can be always fulfilled by α-renaming. For any $\mathsf{r} \in \{\beta_v, \sigma_1, \sigma_3, \sigma, \mathsf{sh}\}$, if $t \mapsto_{\mathsf{r}} t'$ then t is a r*-redex* and t' is its r*-contractum*. A term of the shape $(\lambda x.t)u$ is a β*-redex*. Clearly, any β_v-redex is a β-redex but the converse does not hold: $(\lambda x.z)(yI)$ is a β-redex but not a β_v-redex. Redexes of different kind may *overlap*: for instance, the term $\Delta I\Delta$ is a σ_1-redex and contains the β_v-redex ΔI; the term $\Delta(I\Delta)(xI)$ is a σ_1-redex and contains the σ_3-redex $\Delta(I\Delta)$, which contains in turn the β_v-redex $I\Delta$.

From definitions in Fig. 1 it follows that $\to_{\mathsf{sh}} = \to_{\beta_v}\cup\to_\sigma$ and $\to_\sigma = \to_{\sigma_1}\cup\to_{\sigma_3}$, as well as $\to_{\mathsf{sh}^\flat} = \to_{\beta_v^\flat}\cup\to_{\sigma^\flat}$ and $\to_{\sigma^\flat} = \to_{\sigma_1^\flat}\cup\to_{\sigma_3^\flat}$. The *shuffling* (resp. *balanced shuffling*) *calculus* λ_{sh} (resp. $\lambda_{\mathsf{sh}}^\flat$) is the set Λ of terms endowed with the reduction $\to_{\mathsf{sh}}$ (resp. $\to_{\mathsf{sh}^\flat}$). The set Λ endowed with the reduction $\to_{\beta_v}$ is Plotkin's pure call-by-value λ-calculus λ_v [35], a sub-calculus of λ_{sh}.

Proposition 1 (Basic properties of reductions, [35, 10])**.** *The σ- and $\sigma^\flat$-reductions are confluent and strongly normalizing. The β_v-, $\beta_v^\flat$-,* sh- *and* $\mathsf{sh}^\flat$*-reductions are confluent.*

Example 2. Recall the terms t and u in Eq. (1): $t = (\lambda y.\Delta)(xI)\Delta \to_{\sigma_1^\flat} (\lambda y.\Delta\Delta)(xI) \to_{\beta_v^\flat} (\lambda y.\Delta\Delta)(xI) \to_{\beta_v^\flat} \dots$ and $u = \Delta((\lambda y.\Delta)(xI)) \to_{\sigma_3^\flat} (\lambda y.\Delta\Delta)(xI) \to_{\beta_v^\flat} (\lambda y.\Delta\Delta)(xI) \to_{\beta_v^\flat} \dots$ are the only possible sh-reduction paths from t and u respectively: t and u are not sh-normalizable and $t \simeq_{\mathsf{sh}} u$. But t and u are β_v-normal ($(\lambda y.\Delta)(xI)$ is a stuck β-redex) and different, hence $t \not\simeq_{\beta_v} u$ by confluence of $\to_{\beta_v}$ (Prop. 1).

Example 2 shows how σ-reduction shuffles constructors and moves stuck β-redex in order to unblock β_v-redexes which are hidden by the "hyper-sequential structure" of terms, avoiding "premature" normal

forms. An alternative approach to circumvent the issue of stuck β-redexes is given by λ_{vsub}, the call-by-value λ-calculus with explicit substitutions introduced in [3], where hidden β_v-redexes are reduced using rules acting at a distance. In [2] it has been shown that λ_{vsub} and λ_{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 λ-terms into linear logic proof-nets [20, 1] according to the linear recursive type $o = !o \multimap !o$, or equivalently $o = !(o \multimap o)$. In this translation, sh-reduction corresponds to cut-elimination, more precisely β_v-steps (resp. σ-steps) correspond to exponential (resp. multiplicative) cut-elimination steps; $\mathsf{sh}^\flat$-reduction corresponds to cut-elimination at depth 0.

Consider the two subsets of terms defined by mutual induction (notice that $\Lambda_a \subsetneq \Lambda_n \supsetneq \Lambda_v$):

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

Any $t \in \Lambda_a$ is neither a value nor a β-redex, but an open applicative term with a free "head variable".

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

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

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

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

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

$$\text{Negative Types:}\quad M,N ::= P \multimap Q \qquad\qquad \text{Positive Types:}\quad P,Q ::= [N_1,\dots,N_n] \ \text{ (with } n \in \mathbb{N}\text{)}$$

where $[N_1,\dots,N_n]$ 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 $[N_1,\dots,N_n]$ has to be intended as a conjunction $N_1 \wedge \cdots \wedge N_n$ of negative types $N_1,\dots,N_n$, for a commutative and associative conjunction connective $\wedge$ 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 Γ 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 Γ and Δ have the same domain: if $x \in \mathsf{dom}(\Gamma) = \mathsf{dom}(\Delta)$, then $(\Gamma \uplus \Delta)(x) = \Gamma(x) \uplus \Delta(x)$. An environment Γ such that $\mathsf{dom}(\Gamma) = \{x_1,\dots,x_n\}$ 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 $\Gamma = x_1 : m_1,\dots,x_n : m_n$. Note that the sum of environments $\uplus$ is commutative, associative and, given an environment Γ with $\mathsf{dom}(\Gamma) = \{x_1,\dots,x_n\}$ (for some $n \in \mathbb{N}$), one has $\Gamma \uplus \Gamma_0 = \Gamma$ iff $\Gamma_0 = x_1 : [],\dots,x_n : []$ (i.e. Γ_0 is the neutral element for $\uplus$). The notation $\pi \rhd \Gamma \vdash t : P$ means that π is a derivation with conclusion the judgment $\Gamma \vdash t : P$. We write $\pi \rhd t$ if π is such that $\pi \rhd \Gamma \vdash t : P$ for some environment Γ and positive type P.

$$\frac{}{\Gamma_0, x:P \vdash x:P}\ \mathsf{ax} \qquad \frac{\Gamma \vdash t:[P \multimap Q] \qquad \Gamma' \vdash u:P}{\Gamma \uplus \Gamma' \vdash tu:Q}\ @$$

$$\frac{\Gamma_1, x:P_1 \vdash t:Q_1 \quad \overset{n\in\mathbb{N}}{\cdots} \quad \Gamma_n, x:P_n \vdash t:Q_n}{\Gamma_0 \uplus \Gamma_1 \uplus \cdots \uplus \Gamma_n \vdash \lambda x.t:[P_1 \multimap Q_1, \ldots, P_n \multimap Q_n]}\ \lambda$$

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

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 π is just the the number of @ rules in π. Note that judgments play no role in 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 $\Gamma = x_1:[], \ldots, x_k:[]$, the derivations (typing II and I with same type and same context)

$$\pi_{II} = \frac{\dfrac{\dfrac{}{\Gamma, x:[] \vdash x:[]}\ \mathsf{ax}}{\Gamma \vdash I:[[] \multimap []]}\ \lambda \qquad \dfrac{}{\Gamma \vdash I:[]}\ \lambda}{\Gamma \vdash II:[]}\ @ \qquad\qquad \pi_I = \frac{}{\Gamma \vdash I:[]}\ \lambda$$

are such that $|\pi_{II}| = 1$ and $|\pi_I| = 0$. Note that $II \to_{\mathsf{sh}^\flat} I$ and $|\pi_I| = |\pi_{II}| + 1$.

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).

Proof p. 15 **Lemma 6** (Judgment decomposition for values)**.** *Let $v \in \Lambda_v$, Δ be an environment, and $P_1, \ldots, P_p$ be positive types (for some $p \in \mathbb{N}$). There is a derivation $\pi \rhd \Delta \vdash v: P_1 \uplus \cdots \uplus P_p$ iff for all $1 \leq i \leq p$ there are an environment Δ_i and a derivation $\pi_i \rhd \Delta_i \vdash v: P_i$ such that $\mathsf{dom}(\Delta_1) = \ldots = \mathsf{dom}(\Delta_p) = \mathsf{dom}(\Delta)$ and $\Delta = \biguplus_{i=1}^{p} \Delta_i$. Moreover, $|\pi| = \sum_{i=1}^{p} |\pi_i|$.*

The left-to-right direction of Lemma 6 means that, given $\pi \rhd \Delta \vdash v: P$, 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 $1 \leq i \leq p$.

Proof p. 16 **Lemma 7** (Substitution)**.** *Let $t \in \Lambda$ and $v \in \Lambda_v$. If $\pi \rhd \Gamma, x: P \vdash t: Q$ and $\pi' \rhd \Delta \vdash v: P$ where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Delta)$, then there exists $\pi'' \rhd \Gamma \uplus \Delta \vdash t\{v/x\}: Q$ 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.

Proof p. 18 **Proposition 8** (Quantitative balanced subject reduction)**.** *Let $t, t' \in \Lambda$ and $\pi \rhd \Gamma \vdash t: Q$.*

1. Shrinkage under $\beta_v^\flat$-step: *If $t \to_{\beta_v^\flat} t'$ then $|\pi| > 0$ and there exists a derivation π' with conclusion $\Gamma \vdash t': Q$ such that $|\pi'| = |\pi| - 1$.*
2. Size invariance under $\sigma^\flat$-step: *If $t \to_{\sigma^\flat} t'$ then $|\pi| > 0$ and there exists a derivation π' with conclusion $\Gamma \vdash t': Q$ such that $|\pi'| = |\pi|$.*

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

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). *Proof p. 21*

1. Abstraction vs. abstraction: *Let $n \in \mathbb{N}$. If $\pi \rhd \Delta \vdash \lambda y.(\lambda x.t)v : \uplus_{i=1}^{n}[P'_i \multimap P_i]$ and $y \notin \mathsf{fv}(v)$, then there is $\pi' \rhd \Delta \vdash (\lambda x.\lambda y.t)v : \uplus_{i=1}^{n}[P'_i \multimap P_i]$ such that $|\pi'| = |\pi| + 1 - n$.*
2. Application vs. abstraction: *If $\pi \rhd \Delta \vdash ((\lambda x.t)v)((\lambda x.u)v) : P$ then there exists a derivation $\pi' \rhd \Delta \vdash (\lambda x.tu)v : P$ such that $|\pi'| = |\pi| - 1$.*

Proposition 10 (Quantitative balanced subject expansion). *Let $t, t' \in \Lambda$ and $\pi' \rhd \Gamma \vdash t' : Q$.* *Proof p. 22*

1. Enlargement under anti-$\beta_v^\flat$-step: *If $t \to_{\beta_v^\flat} t'$ then there is $\pi \rhd \Gamma \vdash t : Q$ with $|\pi| = |\pi'| + 1$.*
2. Size invariance under anti-$\sigma^\flat$-step: *If $t \to_{\sigma^\flat} t'$ then $|\pi'| > 0$ and there exists $\pi \rhd \Gamma \vdash t : Q$ such that $|\pi| = |\pi'|$.*

Actually, subject reduction and expansion hold for the whole sh-reduction $\to_{\mathsf{sh}}$, not only for the balanced sh-reduction $\to_{\mathsf{sh}^\flat}$. The drawback for $\to_{\mathsf{sh}}$ is that the quantitative information about the size of the derivation is lost in the case of a β_v-step.

Lemma 11 (Subject reduction). *Let $t, t' \in \Lambda$ and $\pi \rhd \Gamma \vdash t : Q$.* *Proof p. 26*

1. Shrinkage under β_v-step: *If $t \to_{\beta_v} t'$ then there is $\pi' \rhd \Gamma \vdash t' : Q$ with $|\pi| \geq |\pi'|$.*
2. Size invariance under σ-step: *If $t \to_{\sigma} t'$ then there is $\pi' \rhd \Gamma \vdash t' : Q$ such that $|\pi| = |\pi'|$.*

Lemma 12 (Subject expansion). *Let $t, t' \in \Lambda$ and $\pi' \rhd \Gamma \vdash t' : Q$.* *Proof p. 26*

1. Enlargement under anti-β_v-step: *If $t \to_{\beta_v} t'$ then there is $\pi \rhd \Gamma \vdash t : Q$ with $|\pi| \geq |\pi'|$.*
2. Size invariance under anti-σ-step: *If $t \to_{\sigma} t'$ then there is $\pi \rhd \Gamma \vdash t : Q$ such that $|\pi| = |\pi'|$.*

In Lemmas 11.1 and 12.1 it is impossible to estimate more precisely the relationship between $|\pi|$ and $|\pi'|$. Indeed, Ex. 5 has shown that there are $\pi_I \rhd y : [] \vdash I : []$ and $\pi_{II} \rhd y : [] \vdash II : []$ such that $|\pi_I| = 0$ and $|\pi_{II}| = 1$ (where $I = \lambda x.x$). So, given $n \in \mathbb{N}$, consider the derivations $\pi_n \rhd\vdash \lambda y.II : [[] \multimap [], \overset{n}{\ldots}, [] \multimap []]$ and $\pi'_n \rhd\vdash \lambda y.I : [[] \multimap [], \overset{n}{\ldots}, [] \multimap []]$ below:

$$\pi_n = \dfrac{\begin{array}{ccc}\vdots\,\pi_{II} & & \vdots\,\pi_{II} \\ y : [] \vdash II : [] & \overset{n}{\ldots} & y : [] \vdash II : []\end{array}}{\vdash \lambda y.II : [[] \multimap [], \overset{n}{\ldots}, [] \multimap []]}\,\lambda \qquad \pi'_n = \dfrac{\begin{array}{ccc}\vdots\,\pi_{I} & & \vdots\,\pi_{I} \\ y : [] \vdash I : [] & \overset{n}{\ldots} & y : [] \vdash I : []\end{array}}{\vdash \lambda y.I : [[] \multimap [], \overset{n}{\ldots}, [] \multimap []]}\,\lambda$$

Clearly, $\lambda y.II \to_{\mathsf{sh}} \lambda y.I$ (but $\lambda y.II \not\to_{\mathsf{sh}^\flat} \lambda y.I$) and the π'_n (resp. π_n) is the only derivation typing $\lambda y.I$ (resp. $\lambda y.II$) with the same type and environment as π_n (resp. π'_n). One has $|\pi_n| = n \cdot |\pi_{II}| = n$ and $|\pi'_n| = n \cdot |\pi_I| = 0$, thus the difference of size of the derivations π_n and π'_n can be arbitrarely 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 λ_{sh} (Thm. 14 below).

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

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

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

$$[\![t]\!]_{\vec{x}} = \{((P_1, \ldots, P_n), Q) \mid \exists\, \pi \rhd x_1 : P_1, \ldots, x_n : P_n \vdash t : Q\}\,.$$

Essentially, the semantics of a term t for a suitable list $\vec{x}$ of variables is the set of judgments for $\vec{x}$ and t that can be derived in the non-idempotent intersection type system of Fig. 2.

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

$$\mathscr{U}_0 := \emptyset \qquad \mathscr{U}_{k+1} := \mathscr{M}_{\mathrm{f}}(\mathscr{U}_k) \times \mathscr{M}_{\mathrm{f}}(\mathscr{U}_k) \qquad (\mathscr{M}_{\mathrm{f}}(X) \text{ is the set of finite multisets over the set } X)$$

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

Proof p. 27 **Theorem 14** (Invariance under sh-equivalence)**.** *Let $t,u \in \Lambda$, let $n \in \mathbb{N}$ and let $\vec{x} = (x_1,\dots,x_n)$ be a suitable list of variables for t and u. If $t \simeq_{\mathsf{sh}} u$ then $[\![t]\!]_{\vec{x}} = [\![u]\!]_{\vec{x}}$.*

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

Proof p. 27 **Lemma 15** (Semantics of $\mathsf{sh}^\flat$-normal forms)**.** *Let t be a term, let $k \in \mathbb{N}$ and let $\vec{x} = (x_1,\dots,x_k)$ be a list of variables suitable for t.*

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

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) $\mathsf{sh}^\flat$-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 and 4 can be seen as "intermediate stages" in the proof of the main equivalences, which are informative enough to deserve to be explicitely stated.

Proof p. 28 **Theorem 16** (Semantic and logical characterization of $\mathsf{sh}^\flat$-normalization)**.** *Let $t \in \Lambda$ and let $\vec{x} = (x_1,\dots,x_n)$ be a suitable list of variables for t. The following are equivalent:*

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

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

The implication (5)$\Rightarrow$(3) does not hold in Plotkin's λ_v: indeed, the terms t and u in Eq. (1) (see also Ex. 2) are β_v-normal (because of a stuck β-redex) but $[\![t]\!]_x = \emptyset = [\![u]\!]_x$. Equivalences such as the ones in Thm. 16 hold in a call-by-value setting provided that β_v-reduction is extended, *e.g.* by adding σ-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 $\mathsf{sh}^\flat$-normal and $\vec{x} = (x_1, \dots, x_k)$ be suitable for t.* *Proof p. 29*

1. *For every $\pi \rhd x_1 : [], \dots, x_k : [] \vdash t : []$ and $\pi' \rhd x_1 : [], \dots, x_k : [] \vdash t : []$, one has $t \in \Lambda_v$, $|\pi| = 0$ and $\pi = \pi'$. More precisely, π consists only of the rule ax if t is a variable, otherwise t is an abstraction and π consists only of a 0-ary rule λ.*
2. *The following are equivalent:*
   - *(a) t is a value;*
   - *(b) $(([], \overset{k}{\dots}, []), []) \in [\![t]\!]_{\vec{x}}$;*
   - *(c) there exists $\pi \rhd x_1 : [], \dots, x_k : [] \vdash t : []$;*
   - *(d) there exists $\pi \rhd t$ 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 $\mathsf{sh}^\flat$-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 $\vec{x} = (x_1, \dots, x_k)$ be suitable for t. The following are equivalent:* *Proof p. 29*

1. *t is $\mathsf{sh}^\flat$-normalizable and the $\mathsf{sh}^\flat$-normal form of t is a value;*
2. *$(([], \overset{k}{\dots}, []), []) \in [\![t]\!]_{\vec{x}}$;*
3. *there exists $\pi \rhd x_1 : [], \dots, x_k : [] \vdash t : []$.*

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 ($\mathsf{sh}^\flat$-normalizable) term t is an upper bound on the number of $\beta_v^\flat$-steps in *any* $\mathsf{sh}^\flat$-normalizing reduction sequence from t, since the size of a type derivation decreases by 1 after each $\beta_v^\flat$-step, and does not change after each $\sigma^\flat$-step.

Corollary 19 (Upper bound for the number of $\beta_v^\flat$-steps)**.** *Let t be a $\mathsf{sh}^\flat$-normalizable term and t_0 be its $\mathsf{sh}^\flat$-normal form. For any reduction sequence $d : t \to^*_{\mathsf{sh}^\flat} t_0$ and any $\pi \rhd t$, $\mathrm{leng}_{\beta_v^\flat}(d) \leq |\pi|$.*

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

The *balanced size* of a term t, denoted by $|t|_\flat$, is defined by induction on t as follows ($v \in \Lambda_v$):

$$|v|_\flat = 0 \qquad\qquad |tu|_\flat = \begin{cases} |s|_\flat + |u|_\flat + 1 & \text{if } t = \lambda x.s \\ |t|_\flat + |u|_\flat + 1 & \text{otherwise} \end{cases}$$

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 $t = B\langle us \rangle$ for some balanced context B. For instance, $|(\lambda x.yy)(zz)|_\flat = 3$ and $|(\lambda x.\lambda x'.yy)(zz)|_\flat = 2$. The following lemma can be see as a quantitative version of Lemma 15.

Lemma 20 (Relationship between sizes)**.** *Let $t \in \Lambda$, $k \in \mathbb{N}$ and $\vec{x} = (x_1, \dots, x_k)$ suitable for t.* *Proof p. 30*

1. *If t is $\mathsf{sh}^\flat$-normal then $|t|_\flat = \inf\{|\pi| \mid \pi \rhd x_1 : P_1, \dots, x_k : P_k \vdash t : Q\}$.*
2. *If t is a value then $|t|_\flat = \inf\{|\pi| \mid \pi \rhd x_1 : P_1, \dots, x_k : P_k \vdash t : Q\} = 0$.*

Thus, the balanced size of a $\mathsf{sh}^\flat$-normal form n equals the minimal size of the type derivation of n.

Proof p. 31

Proposition 21 (Exact number of $\beta_v^\flat$-steps)**.** *Let t be a $\mathsf{sh}^\flat$-normalizable term and t_0 be its $\mathsf{sh}^\flat$-normal form. For every reduction sequence $d\colon t \to^*_{\mathsf{sh}^\flat} t_0$ and every $\pi \rhd t$ and $\pi_0 \rhd t_0$ such that $|\pi| = \inf\{|\pi'| \mid \pi' \rhd t\}$ and $|\pi_0| = \inf\{|\pi_0'| \mid \pi_0' \rhd t_0\}$, one has*

$$\mathrm{leng}_{\beta_v^\flat}(d) = |\pi| - |t_0|_\flat = |\pi| - |\pi_0|\,. \tag{2}$$

If moreover t_0 is a value, then $\mathrm{leng}_{\beta_v^\flat}(d) = |\pi|$.

Prop. 21 could seem slightly disappointinig: it allows us to know the exact number of $\beta_v^\flat$-steps of a $\mathsf{sh}^\flat$-normalizing reduction sequence from t only if we already know the $\mathsf{sh}^\flat$-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 $\beta_v^\flat$-steps. However, Prop. 21 says also that this limitation is circumvented in the case t $\mathsf{sh}^\flat$-reduces to a value. Moreover, a notable and immediate consequence of Prop. 21 is:

Corollary 22 (Same number of $\beta_v^\flat$-steps)**.** *Let t be a $\mathsf{sh}^\flat$-normalizable term and t_0 be its $\mathsf{sh}^\flat$-normal form. For all reduction sequences $d\colon t \to^*_{\mathsf{sh}^\flat} t_0$ and $d'\colon t \to^*_{\mathsf{sh}^\flat} t_0$,* $\mathrm{leng}_{\beta_v^\flat}(d) = \mathrm{leng}_{\beta_v^\flat}(d')$.

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

Example 23. Let $t := (\lambda y.y')(\Delta(xI))I$: one has $u := (\lambda y.y')(\Delta(xI))\ {}_{\sigma_1^\flat}\!\leftarrow t \to_{\sigma_3^\flat} (\lambda z.(\lambda y.y')(zz))(xI)I =: s$ and the only way to join this critical pair is by performing *one* $\sigma_3^\flat$-step from u and *two* $\sigma_1^\flat$-steps from s, so that $u \to_{\sigma_3^\flat} (\lambda z.(\lambda y.y'I)(zz))(xI)\ {}_{\sigma_1^\flat}\!\leftarrow (\lambda z.(\lambda y.y')(zz)I)(xI)\ {}_{\sigma_1^\flat}\!\leftarrow s$. Since each $\sigma^\flat$-step can create a new β_v-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 $\mathrm{leng}_{\beta_v}\colon \Lambda \to \mathbb{N} \cup \{\infty\}$

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

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

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

Proof p. 32

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

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

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

Remind that, according to Cor. 4, any closed term either is not $\mathsf{sh}^\flat$-normalizable, or it $\mathsf{sh}^\flat$-normalizes to a (closed) value. So, this procedure completely determines (qualitatively and quantitatively) the behavior of closed terms with respect to $\mathsf{sh}^\flat$-reduction.

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

6 Conclusions

In order to get a truly semantic measure of the execution time in the shuffling calculus λ_{sh}, we should first be able to give an *upper bound* to the number of $\beta_v^{\flat}$-steps in a $\mathsf{sh}^{\flat}$-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 $P = [P_1 \multimap Q_1, \dots, P_k \multimap Q_k] \in \mathscr{M}_{\mathrm{f}}(\mathscr{U})$ (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 $\multimap$ in P; in particular, $|[]| = 0$. For any $((P_1, \dots, P_n), Q) \in \mathscr{M}_{\mathrm{f}}(\mathscr{U})^k \times \mathscr{M}_{\mathrm{f}}(\mathscr{U})$ (with $k \in \mathbb{N}$), the *size of* $((P_1, \dots, P_k), Q)$ is $|((P_1, \dots, P_k), Q)| = |Q| + \sum_{i=1}^{k} |P_i|$.

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 λ_{sh} this lemma should claim that "For every sh-*normal* form t, if $\pi \rhd x_1 : P_1, \dots, x_k : P_k \vdash t : Q$ then $|\pi| \leq |((P_1, \dots, P_k), Q)|$". Unfortunately, in λ_{sh} this property is false!

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

$$\pi := \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{z:[],y:[],x:[]\vdash x:[]}\,\mathsf{ax}}{z:[],y:[]\vdash \lambda x.x:[[]\multimap[]]}\,\lambda \qquad \vdots\,\pi_y^{[]}}{z:[],y:[[]\multimap[]]\vdash (\lambda x.x)(yy):[]}\,@}{z:[]\vdash \lambda y.(\lambda x.x)(yy):[[[]\multimap[]]\multimap[]]}\,\lambda \qquad \vdots\,\pi_z^{[[]\multimap[]]}}{z:[[]\multimap[]\multimap[]]]\vdash zz:[[]\multimap[]]}\,@}{z:[[]\multimap[]\multimap[]]]\vdash (\lambda y.(\lambda x.x)(yy))(zz):[]}\,@}{\vdash \lambda z.(\lambda y.(\lambda x.x)(yy))(zz):[[[]\multimap[]\multimap[]]]\multimap[]]}\,\lambda$$

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

$$\pi_x^P := \dfrac{\dfrac{}{x:[[]\multimap P]\vdash x:[[]\multimap P]}\,\mathsf{ax} \qquad \dfrac{}{x:[]\vdash x:[]}\,\mathsf{ax}}{x:[[]\multimap P]\vdash xx:P}\,@\,.$$

Then, $|\pi| = 2 + |\pi_y^{[]}| + |\pi_z^{[[]\multimap[]]}| = 4 > 3 = |[[[]\multimap[[]\multimap[]]]\multimap[]]|$.

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 λ_{sh} and the other ones studied in [2]. Intuitively, in Ex. 25 t contains two applications — $(\lambda x.x)(yy)$ and $(\lambda y.(\lambda x.x)(yy))(zz)$ — that are stuck β-redexes and are the source of two "useless" instances of the rule @ in π. The idea for the new calculus is to "fire" a stuck β-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 β-redex.

Since λ_{sh} is compatible with Girard's call-by-value translation of λ-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 λ_{sh}. Actually, there is no contradiction because LL proof-nets in [12] always require an explicit constructor for dereliction, whereas λ_{sh} is outside of this fragment since variables correspond to exponential axioms (which keep implicit the dereliction).

References

[1] Beniamino Accattoli (2015): *Proof nets and the call-by-value λ-calculus*. *Theor. Comput. Sci.* 606, pp. 2–24.

[2] Beniamino Accattoli & Giulio Guerrieri (2016): *Open Call-by-Value*. In: *APLAS 2016*, pp. 206–226.

[3] Beniamino Accattoli & Luca Paolini (2012): *Call-by-Value Solvability, revisited*. In: *FLOPS*, pp. 4–16.

[4] Hendrik Pieter Barendregt (1984): *The Lambda Calculus – Its Syntax and Semantics*. 103, North-Holland.

[5] Erika De Benedetti & Simona Ronchi Della Rocca (2016): *A type assignment for λ-calculus complete both for FPTIME and strong normalization*. *Inf. Comput.* 248, pp. 195–214.

[6] Alexis Bernadet & Stéphane Lengrand (2013): *Non-idempotent intersection types and strong normalisation*. *Logical Methods in Computer Science* 9(4).

[7] Antonio Bucciarelli & Thomas Ehrhard (2001): *On phase semantics and denotational semantics: the exponentials*. *Ann. Pure Appl. Logic* 109(3), pp. 205–241.

[8] Antonio Bucciarelli, Thomas Ehrhard & Giulio Manzonetto (2012): *A relational semantics for parallelism and non-determinism in a functional setting*. *Ann. Pure Appl. Logic* 163(7), pp. 918–934.

[9] Antonio Bucciarelli, Delia Kesner & Daniel Ventura (2017): *Non-idempotent intersection types for the Lambda-Calculus*. *Logic Journal of the IGPL* 25(4), pp. 431–464.

[10] Alberto Carraro & Giulio Guerrieri (2014): *A Semantical and Operational Account of Call-by-Value Solvability*. In: *FOSSACS 2014*, pp. 103–118.

[11] Daniel de Carvalho (2009): *Execution Time of lambda-Terms via Denotational Semantics and Intersection Types*. *CoRR* abs/0905.4251.

[12] Daniel de Carvalho, Michele Pagani & Lorenzo Tortora de Falco (2011): *A semantic measure of the execution time in linear logic*. *Theor. Comput. Sci.* 412(20), pp. 1884–1902.

[13] Daniel de Carvalho & Lorenzo Tortora de Falco (2016): *A semantic account of strong normalization in linear logic*. *Inf. Comput.* 248, pp. 104–129.

[14] Mario Coppo & Mariangiola Dezani-Ciancaglini (1978): *A new type assignment for λ-terms*. *Arch. Math. Log.* 19(1), pp. 139–156.

[15] Mario Coppo & Mariangiola Dezani-Ciancaglini (1980): *An extension of the basic functionality theory for the λ-calculus*. *Notre Dame Journal of Formal Logic* 21(4), pp. 685–693.

[16] Alejandro Díaz-Caro, Giulio Manzonetto & Michele Pagani (2013): *Call-by-Value Non-determinism in a Linear Logic Type Discipline*. In: *LFCS 2013*, *Lecture Notes in Computer Science* 7734, Springer, pp. 164–178.

[17] Thomas Ehrhard (2012): *Collapsing non-idempotent intersection types*. In: *CSL*, pp. 259–273.

[18] Thomas Ehrhard & Giulio Guerrieri (2016): *The Bang Calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value*. In: *PPDP 2016*, ACM, pp. 174–187.

[19] Philippa Gardner (1994): *Discovering Needed Reductions Using Type Theory*. In: *TACS '94*, *Lecture Notes in Computer Science* 789, Springer, pp. 555–574.

[20] Jean-Yves Girard (1987): *Linear Logic*. *Theoretical Computer Science* 50, pp. 1–102.

[21] Jean-Yves Girard (1988): *Normal functors, power series and λ-calculus*. *Ann. Pure Appl. Logic* 37(2), pp. 129–177.

[22] Benjamin Grégoire & Xavier Leroy (2002): *A compiled implementation of strong reduction*. In: *ICFP '02*, pp. 235–246.

[23] Giulio Guerrieri (2015): *Head reduction and normalization in a call-by-value lambda-calculus*. In: *WPTE 2015*, pp. 3–17.

[24] Giulio Guerrieri, Luca Paolini & Simona Ronchi Della Rocca (2015): *Standardization of a Call-By-Value Lambda-Calculus*. In: *TLCA 2015*, pp. 211–225.

[25] Giulio Guerrieri, Luca Paolini & Simona Ronchi Della Rocca (2017): *Standardization and Conservativity of a Refined Call-by-Value lambda-Calculus*. *Logical Methods in Computer Science* 13(4).

[26] N. D. Jones, C. K. Gomard & P. Sestoft (1993): *Partial Evaluation and Automatic Program Generation*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[27] Delia Kesner & Daniel Ventura (2015): *A Resource Aware Computational Interpretation for Herbelin's Syntax*. In: *ICTAC 2015*, *Lecture Notes in Computer Science* 9399, Springer, pp. 388–403.

[28] Delia Kesner & Pierre Vial (2017): *Types as Resources for Classical Natural Deduction*. In: *FSCD 2017*, *LIPIcs* 84, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 24:1–24:17.

[29] A. J. Kfoury (2000): *A linearization of the Lambda-calculus and consequences*. *J. Log. Comput.* 10(3), pp. 411–436.

[30] Jean-Louis Krivine (1993): *Lambda-calculus, types and models*. Ellis Horwood series, Masson.

[31] Damiano Mazza, Luc Pellissier & Pierre Vial (2018): *Polyadic Approximations, Fibrations and Intersection Types*. *Proceedings of the ACM on Programming Languages* 2(POPL:6).

[32] Peter Møller Neergaard & Harry G. Mairson (2004): *Types, potency, and idempotency: why nonlinearity and amnesia make a type system work*. In: *ICFP 2004*, ACM, pp. 138–149.

[33] Luca Paolini (2002): *Call-by-Value Separability and Computability*. In: *ICTCS*, pp. 74–89.

[34] Luca Paolini, Mauro Piccolo & Simona Ronchi Della Rocca (2017): *Essential and relational models*. *Mathematical Structures in Computer Science* 27(5), pp. 626–650.

[35] Gordon D. Plotkin (1975): *Call-by-Name, Call-by-Value and the lambda-Calculus*. *Theor. Comput. Sci.* 1(2), pp. 125–159.

[36] Garrel Pottinger (1980): *A type assignment for the strongly normalizable λ-terms*. In: *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, Academic Press, pp. 561–577.

[37] Laurent Regnier (1992): *Lambda-calcul et réseaux*. PhD thesis, Univ. Paris VII.

[38] Laurent Regnier (1994): *Une équivalence sur les lambda-termes*. *TCS* 2(126), pp. 281–292.

[39] Simona Ronchi Della Rocca & Luca Paolini (2004): *The Parametric λ-Calculus – A Metamodel for Computation*. Springer.

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 λ-terms is denoted by Λ. We set $I \coloneqq \lambda x.x$ and $\Delta \coloneqq \lambda x.xx$. Let $\to_{\mathsf{r}} \subseteq \Lambda \times \Lambda$.

- The reflexive-transitive closure of $\to_{\mathsf{r}}$ is denoted by $\to_{\mathsf{r}}^*$. The r-*equivalence* $\simeq_{\mathsf{r}}$ is the reflexive-transitive and symmetric closure of $\to_{\mathsf{r}}$.
- Let t be a term: t is r-*normal* if there is no term u such that $t \to_{\mathsf{r}} u$; t is r-*normalizable* if there is a r-normal term u such that $t \to_{\mathsf{r}}^* u$, 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, $\to_{\mathsf{r}}$ is *strongly normalizing* if every $u \in \Lambda$ is strongly r-normalizable.
- $\to_{\mathsf{r}}$ is *confluent* if ${}^*_{\mathsf{r}}\!\leftarrow \cdot \to_{\mathsf{r}}^* \subseteq \to_{\mathsf{r}}^* \cdot {}^*_{\mathsf{r}}\!\leftarrow$. From confluence it follows that: $t \simeq_{\mathsf{r}} u$ iff $t \to_{\mathsf{r}}^* s \, {}^*_{\mathsf{r}}\!\leftarrow u$ 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 $\mathsf{sh}^\flat$-normal forms)**.** *Let t be a term:*

- *t is $\mathsf{sh}^\flat$-normal iff $t \in \Lambda_n$;*
- *t is $\mathsf{sh}^\flat$-normal and is neither a value nor a β-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 $\to_{\mathsf{sh}^\flat}$, one can deduce that u and s $\mathsf{sh}^\flat$-normal, u is not a β-redex (otherwise t would be a σ_1-redex) and if u is of the shape $\lambda x.u'$ then u' is $\mathsf{sh}^\flat$-normal; furthermore t is neither a β_v- nor a σ_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 β-redex: either u is a variable or $u \in \Lambda_a$ by induction hypothesis (since u is neither a value nor a β-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 β-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 $\mathsf{sh}^\flat$-normal, so $u' \in \Lambda_n$ by induction hypothesis and thus $t \in \Lambda_n$ 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 β-redex. We prove the first statement by induction on $t \in \Lambda_n$.

If t is a value then t is $\mathsf{sh}^\flat$-normal (no rule of $\to_{\mathsf{sh}^\flat}$ can be applied to t).

If $t = xv$ for some variable x and value v then x and v are $\mathsf{sh}^\flat$-normal and xv is not a sh-redex; therefore t is $\mathsf{sh}^\flat$-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 $\mathsf{sh}^\flat$-normal, moreover a is not a β-redex (so t is not a σ_3-redex), thus t is $\mathsf{sh}^\flat$-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 $\mathsf{sh}^\flat$-normal by induction hypothesis; moreover a is neither a value nor a β-redex, thus t is not a sh-redex; therefore t is $\mathsf{sh}^\flat$-normal. □

Corollary 4 (Syntactic characterization of closed $\mathsf{sh}^\flat$-normal forms)**.** *Let t be a closed term: t is $\mathsf{sh}^\flat$-normal iff t is a value iff $t = \lambda x.u$ for some term u with $\mathsf{fv}(u) \subseteq \{x\}$.* *See p. 5*

Proof. By Prop. 3 and since t is closed, t is $\mathsf{sh}^\flat$-normal iff t is a value (all terms in Λ_a are open). Since t is closed and variables are open, t is a value iff $t = \lambda x.u$ for some u with $\mathsf{fv}(u) \subseteq \{x\}$. □

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 $\mathsf{fv}(t) \subseteq \mathsf{dom}(\Gamma)$ and if moreover $x \in \mathsf{dom}(\Gamma) \setminus \mathsf{fv}(t)$ then $\Gamma(x) = []$.*

Proof. By straightforward induction on $t \in \Lambda$. □

Remark 27. If t is an application and $\pi \rhd t$, then $|\pi| > 0$.

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

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

⇒: If $v = y$, then the last rule of π is ax and thus $\Delta = x_1 : [], \dots, x_m : [], y : P_1 \uplus \dots \uplus P_p$. So, for all $1 \le i \le p$, there are an environment $\Delta_i = x_1 : [], \dots, x_m : [], y : P_i$ and a derivation

$$\pi_i = \dfrac{}{\Delta_i \vdash v : P_i}\ \mathsf{ax}$$

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 π is λ, so there are $n \in \mathbb{N}$, positive types $Q_1, Q'_1, \dots, Q_n, Q'_n$, environments $\Gamma_1, \dots, \Gamma_n$ such that $\Delta = \biguplus_{j=1}^n \Gamma_j$, $\biguplus_{i=1}^p P_i = [Q_1 \multimap Q'_1, \dots, Q_n \multimap Q'_n]$ and

$$\pi = \dfrac{\begin{array}{ccc} \vdots \pi'_1 & & \vdots \pi'_n \\ \Gamma_1, x : Q_1 \vdash t : Q'_1 & \overset{n}{\dots} & \Gamma_n, x : Q_n \vdash t : Q'_n \end{array}}{\Delta \vdash v : [Q_1 \multimap Q'_1, \dots, Q_n \multimap Q'_n]}\ \lambda$$

Thus, up to renumbering the Q_i's and Q'_i's, there are environments $\Delta_1, \dots, \Delta_p$, derivations $\pi_1, \dots, \pi_p$ and integers $k_1 = 1 \le k_2 \le \dots \le k_p \le k_{p+1} = p$, such that, for all $1 \le i \le p$, $P_i = \biguplus_{j=k_i}^{k_{i+1}} [Q_j \multimap Q'_j]$ and

$$\pi_i = \dfrac{\begin{array}{ccc} \vdots \pi'_{k_i} & & \vdots \pi'_{k_{i+1}} \\ \Gamma_{k_i}, x : Q_{k_i} \vdash t : Q'_{k_i} & \overset{k_{i+1}-k_i}{\dots} & \Gamma_{k_{i+1}}, x : Q_{k_{i+1}} \vdash t : Q'_{k_{i+1}} \end{array}}{\Delta_i \vdash v : P_i}\ \lambda \qquad \text{with } \Delta_i = \bigoplus_{j=k_i}^{k_{i+1}} \Gamma_j$$

where $|\pi_i| = \sum_{j=k_i}^{k_{i+1}} |\pi'_j|$, hence $|\pi| = \sum_{j=1}^{n} |\pi'_j| = \sum_{i=1}^{p} \sum_{j=k_i}^{k_{i+1}} |\pi'_j| = \sum_{i=1}^{n} |\pi_i|$.

$\Leftarrow$**:** If $v = y$ then, for all $1 \leq i \leq p$, the last rule of π_i is ax, so $\Delta_i = x_1 \colon [], \ldots, x_n \colon [], y \colon P_i$ and $|\pi_i| = 0$. Since $\Delta = \uplus_{i=1}^{p} \Delta_i = x_1 \colon [], \ldots, x_n \colon [], y \colon \uplus_{i=1}^{p} P_i$, there is a derivation

$$\pi = \frac{}{\Delta \vdash v \colon \uplus_{i=1}^{p} P_i}\ \mathsf{ax}$$

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

If $v = \lambda x.t$ then, for all $1 \leq i \leq p$, the last rule of π_i is λ, so there are $k_i \in \mathbb{N}$, environments $\Delta_{i1}, \ldots, \Delta_{ik_i}$, positive types $Q_{i1}, Q'_{i1}, \ldots, Q_{ik_i}, Q'_{ik_i}$ and derivations $\pi_{i1}, \ldots, \pi_{ik_i}$ such that $P_i = \uplus_{j=1}^{k_i} [Q_{ij} \multimap Q'_{ij}]$, $\Delta_i = \uplus_{j=1}^{k_i} \Delta_{ij}$ and

$$\pi_i = \frac{\begin{array}{ccc} \vdots \pi_{i1} & & \vdots \pi_{ik_i} \\ \Delta_{i1}, x \colon Q_{i1} \vdash t \colon Q'_{i1} & \overset{k_i}{\ldots} & \Delta_{ik_i}, x \colon Q_{ik_i} \vdash t \colon Q'_{ik_i} \end{array}}{\Delta_i \vdash v \colon P_i}\ \lambda \qquad \text{where } |\pi_i| = \sum_{j=1}^{k_i} |\pi_{ij}|.$$

Since $\Delta = \uplus_{i=1}^{p} \Delta_i$ and $\uplus_{i=1}^{p} P_i = \uplus_{i=1}^{p} \uplus_{j=1}^{k_i} [Q_{ij} \multimap Q'_{ij}]$, there is a derivation $\pi =$

$$\frac{\begin{array}{cccc} \vdots \pi_{11} & \vdots \pi_{1k_1} & \vdots \pi_{p1} & \vdots \pi_{pk_p} \\ \Delta_{11}, x \colon Q_{11} \vdash t \colon Q'_{11} \overset{k_1}{\ldots} & \Delta_{1k_1}, x \colon Q_{1k_1} \vdash t \colon Q'_{1k_1} \overset{p}{\ldots} & \Delta_{p1}, x \colon Q_{p1} \vdash t \colon Q'_{p1} \overset{k_p}{\ldots} & \Delta_{pk_p}, x \colon Q_{pk_p} \vdash t \colon Q'_{pk_p} \end{array}}{\Delta \vdash v \colon \uplus_{i=1}^{p} P_i}\ \lambda$$

where $|\pi| = \sum_{i=1}^{p} \sum_{j=1}^{k_i} |\pi_{ij}| = \sum_{i=1}^{p} |\pi_i|$ because $|\pi_i| = \sum_{j=1}^{k_i} |\pi_{ij}|$. □

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

Proof. Apply the right-to-left direction of Lemma 6 taking $p = 0$. □

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

1. Adding weakening: *If $\pi \rhd \Gamma \vdash t \colon P$ then there is $\pi' \rhd \Gamma, x \colon [] \vdash t \colon P$ such that $|\pi| = |\pi'|$.*
2. Removing weakening: *If $\pi' \rhd \Gamma, x \colon [] \vdash t \colon P$ and $x \notin \mathsf{fv}(t)$, then there exists $\pi \rhd \Gamma \vdash t \colon P$ such that $|\pi| = |\pi'|$.*

Proof. Both points are proved by straightforward induction on $t \in \Lambda$. □

Lemma 29 means that, given two judgments $\Gamma \vdash t \colon P$ and $\Delta \vdash u \colon Q$, we can suppose without loss of generality that $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Delta)$. In Lemma 29.1 the hypothesis $x \notin \mathsf{dom}(\Gamma)$ implies $x \notin \mathsf{fv}(t)$, by Lemma 26. Lemma 29.2 — a weakened converse of Lemma 29.1 — fails when $x \in \mathsf{fv}(t)$, indeed the judgment $x \colon [] \vdash x \colon []$ is derivable but the judgment $\vdash x \colon []$ is not.

See p. 6 **Lemma 7** (Substitution)**.** *Let $t \in \Lambda$ and $v \in \Lambda_v$. If $\pi \rhd \Gamma, x \colon P \vdash t \colon Q$ and $\pi' \rhd \Delta \vdash v \colon P$ where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Delta)$, then there exists $\pi'' \rhd \Gamma \uplus \Delta \vdash t\{v/x\} \colon Q$ 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 π is ax with $P = Q$ and $\Gamma = y_1 : [], \dots, y_n : []$ ($y_i \neq x$ for all $1 \leq i \leq n$), whence $\Gamma \uplus \Delta = \Delta$ and $|\pi| = 0$. We conclude by setting $\pi'' = \pi'$.

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

$$\pi'' = \frac{}{\Gamma \vdash y : Q}\ \mathsf{ax}$$

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

If $t = us$, then $t\{v/x\} = u\{v/x\}s\{v/x\}$ and

$$\pi = \frac{\begin{array}{cc}\vdots \pi_1 & \vdots \pi_2 \\ \Gamma_1, x : P_1 \vdash u : [Q_2 \multimap Q] & \Gamma_2, x : P_2 \vdash s : Q_2\end{array}}{\Gamma, x : P \vdash t : Q}\ @$$

with $|\pi| = |\pi_1| + |\pi_2| + 1$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $P = P_1 \uplus P_2$. According to Lemma 6, there are environments Δ_1, Δ_2 and derivations $\pi'_1 \rhd \Delta_1 \vdash v : P_1$ and $\pi'_2 \rhd \Delta_2 \vdash v : P_2$ such that $\Delta = \Delta_1 \uplus \Delta_2$ and $|\pi'| = |\pi'_1| + |\pi'_2|$. By induction hypothesis, there are derivations π''_1 and π''_2 with conclusion $\Gamma_1 \uplus \Delta_1 \vdash u\{v/x\} : [Q_2 \multimap Q]$ and $\Gamma_2 \uplus \Delta_2 \vdash s\{v/x\} : Q_2$, respectively, such that $|\pi''_1| = |\pi_1| + |\pi'_1|$ and $|\pi''_2| = |\pi_2| + |\pi'_2|$. As $\Gamma \uplus \Delta = \Gamma_1 \uplus \Delta_1 \uplus \Gamma_2 \uplus \Delta_2$, there is a derivation

$$\pi'' = \frac{\begin{array}{cc}\vdots \pi''_1 & \vdots \pi''_2 \\ \Gamma_1 \uplus \Delta_1 \vdash u\{v/x\} : [Q_2 \multimap Q] & \Gamma_2 \uplus \Delta_2 \vdash s\{v/x\} : Q_2\end{array}}{\Gamma \uplus \Delta \vdash t\{v/x\} : Q}\ @$$

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

If $t = \lambda y.u$, then we can suppose without loss of generality that $y \notin \mathsf{fv}(v) \cup \{x\} \cup \mathsf{dom}(\Delta)$, therefore $t\{v/x\} = \lambda y.u\{v/x\}$ and there are $n \in \mathbb{N}$, environments $\Gamma_1, \dots, \Gamma_n$, positive types $P_1, Q_1, Q'_1, \dots, P_n, Q_n, Q'_n$ such that $\Gamma = \biguplus_{i=1}^n \Gamma_i$, $P = \biguplus_{i=1}^n P_i$, $Q = \biguplus_{i=1}^n [Q_i \multimap Q'_i]$ and

$$\pi = \frac{\begin{array}{ccc}\vdots \pi_1 & & \vdots \pi_n \\ \Gamma_1, y : Q_1, x : P_1 \vdash u : Q'_1 & \overset{n}{\cdots} & \Gamma_n, y : Q_n, x : P_n \vdash u : Q'_n\end{array}}{\Gamma, x : P \vdash t : Q}\ \lambda$$

with $|\pi| = \sum_{i=1}^n |\pi_i|$. According to Lemma 29.1, since $y \notin \mathsf{dom}(\Delta)$, there is a derivation π'_0 with conclusion $\Delta, y : [] \vdash v : P$ such that $|\pi'_0| = |\pi'|$. By applying Lemma 6 to π'_0 (as $P = \biguplus_{i=1}^n P_i$), for all $1 \leq i \leq n$ there are an environment Δ_i and a derivation $\pi'_i \rhd \Delta_i, y : [] \vdash v : P_i$ such that $\Delta = \biguplus_{i=1}^n \Delta_i$ and $|\pi'_0| = \sum_{i=1}^n |\pi'_i|$. By induction hypothesis, for all $1 \leq i \leq n$, there is a derivation $\pi''_i \rhd \Gamma_i \uplus \Delta_i, y : Q_i \vdash u\{v/x\} : Q'_i$ such that $|\pi''_i| = |\pi_i| + |\pi'_i|$. Since $\Gamma \uplus \Delta = \biguplus_{i=1}^n \Gamma_i \uplus \Delta_i$, there is a derivation

$$\pi'' = \frac{\begin{array}{ccc}\vdots \pi''_1 & & \vdots \pi''_n \\ \Gamma_1 \uplus \Delta_1, y : Q_1 \vdash u\{v/x\} : Q'_1 & \overset{n}{\cdots} & \Gamma_n \uplus \Delta_n, y : Q_n \vdash u\{v/x\} : Q'_n\end{array}}{\Gamma \uplus \Delta \vdash t\{v/x\} : Q}\ \lambda$$

where $|\pi''| = \sum_{i=1}^n |\pi''_i| = \sum_{i=1}^n |\pi_i| + \sum_{i=1}^n |\pi'_i| = |\pi| + |\pi'_0| = |\pi| + |\pi'|$. □

See p. 6 **Proposition 8** (Quantitative balanced subject reduction). *Let $t, t' \in \Lambda$ and $\pi \rhd \Gamma \vdash t : Q$.*

1. Shrinkage under $\beta_v^\flat$-step: *If $t \to_{\beta_v^\flat} t'$ then $|\pi| > 0$ and there exists a derivation π' with conclusion $\Gamma \vdash t' : Q$ such that $|\pi'| = |\pi| - 1$.*
2. Size invariance under $\sigma^\flat$-step: *If $t \to_{\sigma^\flat} t'$ then $|\pi| > 0$ and there exists a derivation π' with conclusion $\Gamma \vdash t' : Q$ such that $|\pi'| = |\pi|$.*

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

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

$$\pi = \dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1, x : P \vdash u : Q}\,\lambda \qquad \vdots\,\pi_2}{\Gamma \vdash t : Q}\,@ \qquad \Gamma_1 \vdash \lambda x.u : [P \multimap Q] \qquad \Gamma_2 \vdash v : P$$

  where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi| = |\pi_1| + |\pi_2| + 1$. By the substitution lemma (Lemma 7), there exists a derivation $\pi' \rhd \Gamma \vdash t' : Q$ such that $|\pi'| = |\pi_1| + |\pi_2| = |\pi| - 1$.
- *Application Left, i.e. $t = us \to_{\beta_v^\flat} u's = t'$ with $u \to_{\beta_v^\flat} u'$*: then,

$$\pi = \dfrac{\vdots\,\pi_1 \qquad \vdots\,\pi_2}{\Gamma \vdash t : Q}\,@ \qquad \Gamma_1 \vdash u : [P \multimap Q] \qquad \Gamma_2 \vdash s : P$$

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

$$\pi' = \dfrac{\vdots\,\pi_1' \qquad \vdots\,\pi_2}{\Gamma \vdash t' : Q}\,@ \qquad \Gamma_1 \vdash u' : [P \multimap Q] \qquad \Gamma_2 \vdash s : P$$

  where $|\pi'| = |\pi_1'| + |\pi_2| + 1 = |\pi_1| + |\pi_2| + 1 - 1 = |\pi| - 1$.
- *Application Right, i.e. $t = us \to_{\beta_v^\flat} us' = t'$ with $s \to_{\beta_v^\flat} s'$*: then,

$$\pi = \dfrac{\vdots\,\pi_1 \qquad \vdots\,\pi_2}{\Gamma \vdash t : Q}\,@ \qquad \Gamma_1 \vdash u : [P \multimap Q] \qquad \Gamma_2 \vdash s : P$$

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

$$\pi' = \dfrac{\vdots\,\pi_1 \qquad \vdots\,\pi_2'}{\Gamma \vdash t' : Q}\,@ \qquad \Gamma_1 \vdash u : [P \multimap Q] \qquad \Gamma_2 \vdash s' : P$$

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

- *Step inside a β-redex, i.e.* $t = (\lambda x.u)s \to_{\beta_v^\flat} (\lambda x.u')s = t'$ with $u \to_{\beta_v^\flat} u'$: then,

$$\pi = \dfrac{\dfrac{\begin{array}{c}\vdots\pi_1\\ \Gamma_1, x: P \vdash u: Q\end{array}}{\Gamma_1 \vdash \lambda x.u: [P \multimap Q]}\,\lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s: P\end{array}}{\Gamma \vdash t: Q}\,@$$

where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi| = |\pi_1| + |\pi_2| + 1 > 0$. By induction hypothesis, there exists a derivation $\pi_1' \rhd \Gamma_1 \vdash u': [P \multimap Q]$ such that $|\pi_1'| = |\pi_1| - 1$. Therefore, there is a derivation

$$\pi' = \dfrac{\dfrac{\begin{array}{c}\vdots\pi_1'\\ \Gamma_1, x: P \vdash u': Q\end{array}}{\Gamma_1 \vdash \lambda x.u': [P \multimap Q]}\,\lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s: P\end{array}}{\Gamma \vdash t': Q}\,@$$

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

2. Since t is not σ-normal, t is not a value and thus $|\pi| > 0$ according to Remark 27. The proof that there exists a derivation π' with conclusion $\Gamma \vdash t': Q$ such that $|\pi'| = |\pi|$ is by induction con $t \in \Lambda$. Cases:
   - *Step at the root*: there are two sub-cases:
     - $t = (\lambda x.u)sr \mapsto_{\sigma_1} (\lambda x.ur)s = t'$ with $x \notin \mathsf{fv}(r)$: then,

$$\pi = \dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\pi_1\\ \Gamma_1, x: P \vdash u: [Q' \multimap Q]\end{array}}{\Gamma_1 \vdash \lambda x.u: [P \multimap [Q' \multimap Q]]}\,\lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s: P\end{array}}{\Gamma_1 \uplus \Gamma_2 \vdash t: [Q' \multimap Q]}\,@ \qquad \begin{array}{c}\vdots\pi_3\\ \Gamma_3 \vdash r: Q'\end{array}}{\Gamma \vdash t: Q}\,@$$

     with $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2) = \mathsf{dom}(\Gamma_3)$, $\Gamma = \Gamma_1 \uplus \Gamma_2 \uplus \Gamma_3$ and $|\pi| = |\pi_1| + |\pi_2| + |\pi_3| + 2$. According to Lemma 26 and Lemma 29.2, we can suppose without loss of generality that $x \notin \mathsf{dom}(\Gamma_3)$, since $x \notin \mathsf{fv}(r)$. By Lemma 29.1, there exists $\pi_3' \rhd \Gamma_3, x: [\,] \vdash r: Q'$ such that $|\pi_3'| = |\pi_3|$. Therefore, there is a derivation

$$\pi' = \dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\pi_1\\ \Gamma_1, x: P \vdash u: [Q' \multimap Q]\end{array} \qquad \begin{array}{c}\vdots\pi_3'\\ \Gamma_3, x: [\,] \vdash r: Q'\end{array}}{\Gamma_1 \uplus \Gamma_3, x: P \vdash ur: Q}\,@}{\Gamma_1 \uplus \Gamma_3 \vdash \lambda x.ur: [P \multimap Q]}\,\lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s: P\end{array}}{\Gamma \vdash t': Q}\,@$$

     where $|\pi'| = |\pi_1| + |\pi_3'| + 1 + |\pi_2| + 1 = |\pi_1| + |\pi_3| + 1 + |\pi_2| + 1 = |\pi|$.
     - $t = v((\lambda x.u)s) \mapsto_{\sigma_3} (\lambda x.vu)s = t'$ with $x \notin \mathsf{fv}(v)$: then,

$$\pi = \dfrac{\begin{array}{c}\vdots\pi_1\\ \Gamma_1 \vdash v: [Q' \multimap Q]\end{array} \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\pi_2\\ \Gamma_2, x: P \vdash u: Q'\end{array}}{\Gamma_2 \vdash \lambda x.u: [P \multimap Q']}\,\lambda \qquad \begin{array}{c}\vdots\pi_3\\ \Gamma_3 \vdash s: P\end{array}}{\Gamma_2 \uplus \Gamma_3 \vdash (\lambda x.u)s: Q'}\,@}{\Gamma \vdash t: Q}\,@$$

with $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2) = \mathsf{dom}(\Gamma_3)$, $\Gamma = \Gamma_1 \uplus \Gamma_2 \uplus \Gamma_3$ and $|\pi| = |\pi_1| + |\pi_2| + |\pi_3| + 2$. According to Lemma 26 and Lemma 29.2, we can suppose without loss of generality that $x \notin \mathsf{dom}(\Gamma_1)$, since $x \notin \mathsf{fv}(v)$. By Lemma 29.1, there exists $\pi_1' \rhd \Gamma_1, x\colon [\,] \vdash v\colon [Q' \multimap Q]$ such that $|\pi_1'| = |\pi_1|$. So, there is a derivation

$$\pi' = \dfrac{\dfrac{\dfrac{\vdots\,\pi_1' \qquad \vdots\,\pi_3}{\Gamma_1, x\colon [\,] \vdash v\colon [Q' \multimap Q] \qquad \Gamma_3, x\colon P \vdash u\colon Q'}\,@}{\Gamma_1 \uplus \Gamma_3, x\colon P \vdash vu\colon Q}}{\Gamma_1 \uplus \Gamma_3 \vdash \lambda x.ur\colon [P \multimap Q]}\,\lambda \qquad \dfrac{\vdots\,\pi_2}{\Gamma_2 \vdash s\colon P}}{\Gamma \vdash t'\colon Q}\,@$$

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

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

$$\pi = \dfrac{\dfrac{\vdots\,\pi_1 \qquad \vdots\,\pi_2}{\Gamma_1 \vdash u\colon [P \multimap Q] \qquad \Gamma_2 \vdash s\colon P}\,@}{\Gamma \vdash t\colon Q}$$

where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi| = |\pi_1| + |\pi_2| + 1$. By induction hypothesis, there exists a derivation $\pi_1' \rhd \Gamma_1 \vdash u'\colon [P \multimap Q]$ such that $|\pi_1'| = |\pi_1|$. Therefore, there exists a derivation

$$\pi' = \dfrac{\dfrac{\vdots\,\pi_1' \qquad \vdots\,\pi_2}{\Gamma_1 \vdash u'\colon [P \multimap Q] \qquad \Gamma_2 \vdash s\colon P}\,@}{\Gamma \vdash t'\colon Q}$$

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

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

$$\pi = \dfrac{\dfrac{\vdots\,\pi_1 \qquad \vdots\,\pi_2}{\Gamma_1 \vdash u\colon [P \multimap Q] \qquad \Gamma_2 \vdash s\colon P}\,@}{\Gamma \vdash t\colon Q}$$

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

$$\pi' = \dfrac{\dfrac{\vdots\,\pi_1 \qquad \vdots\,\pi_2'}{\Gamma_1 \vdash u\colon [P \multimap Q] \qquad \Gamma_2 \vdash s'\colon P}\,@}{\Gamma \vdash t'\colon Q}$$

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

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

$$\pi = \dfrac{\dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1, x\colon P \vdash u\colon Q}}{\Gamma_1 \vdash \lambda x.u\colon [P \multimap Q]}\,\lambda \qquad \dfrac{\vdots\,\pi_2}{\Gamma_2 \vdash s\colon P}}{\Gamma \vdash t\colon Q}\,@$$

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

$$\pi' = \dfrac{\dfrac{\dfrac{\vdots\,\pi_1'}{\Gamma_1, x : P \vdash u' : Q}}{\Gamma_1 \vdash \lambda x.u' : [P \multimap Q]}\,\lambda \qquad \dfrac{\vdots\,\pi_2}{\Gamma_2 \vdash s : P}}{\Gamma \vdash t' : Q}\,@$$

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

Lemma 9 (Abstraction commutation)**.** *See p. 7*

1. Abstraction vs. abstraction: *Let $n \in \mathbb{N}$. If $\pi \rhd \Delta \vdash \lambda y.(\lambda x.t)v : \biguplus_{i=1}^n [P_i' \multimap P_i]$ and $y \notin \mathsf{fv}(v)$, then there is $\pi' \rhd \Delta \vdash (\lambda x.\lambda y.t)v : \biguplus_{i=1}^n [P_i' \multimap P_i]$ such that $|\pi'| = |\pi| + 1 - n$.*
2. Application vs. abstraction: *If $\pi \rhd \Delta \vdash ((\lambda x.t)v)((\lambda x.u)v) : P$ then there exists a derivation $\pi' \rhd \Delta \vdash (\lambda x.tu)v : P$ such that $|\pi'| = |\pi| - 1$.*

Proof. 1. We can suppose without loss of generality that $x, y \notin \mathsf{dom}(\Delta)$. Therefore, there are derivations $\pi_1^1, \pi_1^2, \dots, \pi_n^1, \pi_n^2$, environments $\Gamma_1, \Delta_1, \dots, \Gamma_n, \Delta_n$ (with $\mathsf{dom}(\Delta) = \mathsf{dom}(\Gamma_i) = \mathsf{dom}(\Delta_i)$ for all $1 \le i \le n$) and positive types $Q_i, \dots, Q_n$ such that

$$\pi = \dfrac{\dfrac{\dfrac{\dfrac{\vdots\,\pi_i^1}{\Gamma_i, y : P_i', x : Q_i \vdash t : P_i}}{\Gamma_i, y : P_i' \vdash \lambda x.t : [Q_i \multimap P_i]}\,\lambda \qquad \dfrac{\vdots\,\pi_i^2}{\Delta_i \vdash v : Q_i}}{\Gamma_i \uplus \Delta_i, y : P_i' \vdash (\lambda x.t)v : P_i}\,@ \qquad (\text{for all } 1 \le i \le n)}{\Delta \vdash \lambda y.(\lambda x.t)v : \biguplus_{i=1}^n [P_i' \multimap P_i]}\,\lambda$$

where $\Delta = \biguplus_{i=1}^n \Gamma_i \uplus \Delta_i$ and $|\pi| = \sum_{i=1}^n (|\pi_i^1| + |\pi_i^2| + 1) = n + \sum_{i=1}^n (|\pi_i^1| + |\pi_i^2|)$. According to Lemma 6, there exists $\pi_2 \rhd \Delta' \vdash v : \biguplus_{i=1}^n Q_i$ with $\Delta' = \biguplus_{i=1}^n \Delta_i$ (whence $\Delta = \Delta' \uplus \biguplus_{i=1}^n \Gamma_i$) and $|\pi_2| = \sum_{i=1}^n |\pi_i^2|$, thus one has

$$\pi' = \dfrac{\dfrac{\dfrac{\dfrac{\vdots\,\pi_i^1}{\Gamma_i, y : P_i', x : Q_i \vdash t : P_i} \qquad (\text{for all } 1 \le i \le n)}{\biguplus_{i=1}^n \Gamma_i, x : \biguplus_{i=1}^n Q_i \vdash \lambda y.t : \biguplus_{i=1}^n [P_i' \multimap P_i]}\,\lambda}{\biguplus_{i=1}^n \Gamma_i \vdash \lambda x.\lambda y.t : [\biguplus_{i=1}^n Q_i \multimap \biguplus_{i=n}^n [P_i' \multimap P_i]]}\,\lambda \qquad \dfrac{\vdots\,\pi_2}{\Delta' \vdash v : \biguplus_{i=1}^n Q_i}}{\Delta \vdash (\lambda x.\lambda y.t)v : \biguplus_{i=n}^n [P_i' \multimap P_i]}\,@$$

where $|\pi'| = |\pi_2| + 1 + \sum_{i=1}^n |\pi_i^1| = 1 + \sum_{i=1}^n (|\pi_i^1| + |\pi_i^2|) = |\pi| + 1 - n$.

2. We can suppose without loss of generality that $x \notin \mathsf{dom}(\Delta)$. 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

$$\pi = \dfrac{\dfrac{\dfrac{\dfrac{\vdots\,\pi_3}{\Delta_3, x : P_1 \vdash t : [Q \multimap P]}}{\Delta_3 \vdash \lambda x.u : [P_1 \multimap [Q \multimap P]]}\,\lambda \qquad \dfrac{\vdots\,\pi_1}{\Delta_1 \vdash v : P_1}}{\Delta_3 \uplus \Delta_1 \vdash (\lambda x.t)v : [Q \multimap P]}\,@ \qquad \dfrac{\dfrac{\dfrac{\vdots\,\pi_4}{\Delta_4, x : P_2 \vdash u : Q}}{\Delta_4 \vdash \lambda x.u : [P_2 \multimap Q]}\,\lambda \qquad \dfrac{\vdots\,\pi_2}{\Delta_2 \vdash v : P_2}}{\Delta_4 \uplus \Delta_2 \vdash (\lambda x.u)v : Q}\,@}{\Delta \vdash ((\lambda x.t)v)((\lambda x.u)v) : P}\,@$$

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 6, there is $\pi_0 \rhd \Delta_1 \uplus \Delta_2 \vdash v : P_1 \uplus P_2$ such that $|\pi_0| = |\pi_1| + |\pi_2|$, thus there exists

$$\pi' = \dfrac{\dfrac{\dfrac{\vdots \pi_3 \quad \Delta_3, x : P_1 \vdash t : [Q \multimap P] \qquad \vdots \pi_4 \quad \Delta_4, x : P_2 \vdash u : Q}{\Delta_3 \uplus \Delta_4, x : P_1 \uplus P_2 \vdash tu : P}\ @}{\Delta_3 \uplus \Delta_4 \vdash \lambda x.tu : [P_1 \uplus P_2 \multimap P]}\ \lambda \qquad \vdots \pi_0 \quad \Delta_1 \uplus \Delta_2 \vdash v : P_1 \uplus P_2}{\Delta \vdash (\lambda x.tu)v : P}\ @$$

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

See p. 7 **Proposition 10** (Quantitative balanced subject expansion)**.** *Let $t, t' \in \Lambda$ and $\pi' \rhd \Gamma \vdash t' : Q$.*

1. Enlargement under anti-$\beta_v^\flat$-step: *If $t \to_{\beta_v^\flat} t'$ then there is $\pi \rhd \Gamma \vdash t : Q$ with $|\pi| = |\pi'| + 1$.*
2. Size invariance under anti-$\sigma^\flat$-step: *If $t \to_{\sigma^\flat} t'$ then $|\pi'| > 0$ and there exists $\pi \rhd \Gamma \vdash t : Q$ such that $|\pi| = |\pi'|$.*

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

- *Step at the root, i.e.* $t = (\lambda x.u)v \mapsto_{\beta_v} u\{v/x\} = t'$. We proceed by induction on $u \in \Lambda$.
  - If $u = x$, then $t' = v$ and $\pi' \rhd \Gamma \vdash v : Q$, while $t = (\lambda x.x)v$. Setting $\Gamma_0 = x_1 : [], \ldots, x_n : []$ where $\mathsf{dom}(\Gamma_0) = \mathsf{dom}(\Gamma)$, we have the derivation
  $$\pi = \dfrac{\dfrac{}{\Gamma_0, x : Q \vdash x : Q}\ \mathsf{ax}}{\Gamma_0 \vdash \lambda x.x : [Q \multimap Q]}\ \lambda \qquad \dfrac{\vdots \pi' \quad \Gamma \vdash v : Q}{\Gamma \vdash t : Q}\ @$$
  with $|\pi| = |\pi'| + 1$.
  - If $u = y \neq x$ (we can suppose without loss of generality that $x \notin \mathsf{fv}(v)$), then $t' = y$ and $\pi' = \dfrac{}{\Gamma_0, y : Q \vdash y : Q}\ \mathsf{ax}$ with $\Gamma = \Gamma_0, y : Q$ and $\Gamma_0 = x_1 : [], \ldots, x_n : []$ (where $\mathsf{dom}(\Gamma_0) = \mathsf{fv}(v) \setminus \{y\}$), while $t = (\lambda x.y)v$. Notice that $|\pi'| = 0$. We have:
  $$\pi = \dfrac{\dfrac{\dfrac{}{\Gamma_0, x : [], y : Q \vdash y : Q}\ \mathsf{ax}}{\Gamma_0, y : Q \vdash \lambda x.y : [[] \multimap Q]}\ \lambda \qquad \dfrac{}{\Gamma_0, y : [] \vdash v : []}\ \lambda}{\Gamma \vdash t : Q}\ @$$
  (notice that the rule λ in π has 0 premises) with $|\pi| = 1 = |\pi'| + 1$.
  - If $u = \lambda y.s$ (we can suppose without loss of generality that $y \notin \mathsf{fv}(v) \cup \{x\}$), then $t' = \lambda y.s\{v/x\}$ and $t = (\lambda x.\lambda y.s)v$. As $\pi' \rhd \Gamma \vdash t' : Q$, there are $n \in \mathbb{N}$, positive types $P_1, Q_1, \ldots, P_n, Q_n$, environments $\Gamma_1, \ldots, \Gamma_n$ with $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \cdots = \mathsf{dom}(\Gamma_n)$, and derivations $\pi'_1, \ldots, \pi'_n$ such that $Q = [P_1 \multimap Q_1, \ldots, P_n \multimap Q_n]$ and
  $$\pi' = \dfrac{\vdots \pi'_1 \quad \Gamma_1, y : P_1 \vdash s\{v/x\} : Q_1 \qquad \overset{n \in \mathbb{N}}{\cdots} \qquad \vdots \pi'_n \quad \Gamma_n, y : P_n \vdash s\{v/x\} : Q_n}{\Gamma \vdash t' : Q}\ \lambda$$

where $\Gamma = \biguplus_{i=1}^{n} \Gamma_i$ and $|\pi'| = \sum_{i=1}^{n} |\pi'_i|$. Let $1 \leq i \leq n$: since $(\lambda x.s)v \mapsto_{\beta_v} s\{v/x\}$, then by *i.h.* there is $\pi_i \rhd \Gamma_i, y : P_i \vdash (\lambda x.s)v : Q_i$ with $|\pi_i| = |\pi'_i| + 1$. So, we set

$$\pi'' = \dfrac{\begin{array}{cc}\vdots\pi_1 & \vdots\pi_n \\ \Gamma_1, y : P_1 \vdash (\{s/x\})v : Q_1 & \overset{n\in\mathbb{N}}{\ldots} \quad \Gamma_n, y : P_n \vdash (\lambda x.s)v : Q_n\end{array}}{\Gamma \vdash \lambda y.(\lambda x.s)v : Q}\ \lambda$$

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 $\pi \rhd \Gamma \vdash t : Q$ where $|\pi| = |\pi''| - n + 1 = |\pi'| + 1$.

– Finally, if $u = sr$, then $t' = s\{v/x\}r\{v/x\}$ and $t = (\lambda x.sr)v$. Since $\pi' \rhd \Gamma \vdash t' : Q$, there are derivations π'_1 and π'_2, a positive type P, environments Γ_1 and Γ_2 (where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2)$ and $\Gamma = \Gamma_1 \uplus \Gamma_2$) such that

$$\pi' = \dfrac{\begin{array}{cc}\vdots\pi'_1 & \vdots\pi'_2 \\ \Gamma_1 \vdash s\{v/x\} : [P \multimap Q] & \Gamma_2 \vdash r\{v/x\} : P\end{array}}{\Gamma \vdash t' : Q}\ @$$

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

$$\pi'' = \dfrac{\begin{array}{cc}\vdots\pi_1 & \vdots\pi_2 \\ \Gamma_1 \vdash (\lambda x.s)v : [P \multimap Q] & \Gamma_2 \vdash (\lambda x.r)v : P\end{array}}{\Gamma \vdash ((\lambda x.s)v)((\lambda x.r)v) : Q}\ @$$

where $|\pi''| = |\pi_1| + |\pi_2| + 1 = |\pi'_1| + |\pi'_2| + 3 = |\pi'| + 2$. According to Lemma 9.2, there is a derivation $\pi \rhd \Gamma \vdash t : Q$ with $|\pi| = |\pi''| - 1 = |\pi'| + 1$.

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

$$\pi' = \dfrac{\begin{array}{cc}\vdots\pi'_1 & \vdots\pi_2 \\ \Gamma_1 \vdash u' : [P \multimap Q] & \Gamma_2 \vdash s : P\end{array}}{\Gamma \vdash t' : Q}\ @$$

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

$$\pi = \dfrac{\begin{array}{cc}\vdots\pi_1 & \vdots\pi_2 \\ \Gamma_1 \vdash u : [P \multimap Q] & \Gamma_2 \vdash s : P\end{array}}{\Gamma \vdash t : Q}\ @$$

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

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

$$\pi' = \dfrac{\begin{array}{cc}\vdots\pi_1 & \vdots\pi'_2 \\ \Gamma_1 \vdash u : [P \multimap Q] & \Gamma_2 \vdash s' : P\end{array}}{\Gamma \vdash t' : Q}\ @$$

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

$$\pi = \dfrac{\begin{array}{cc}\vdots\pi_1 & \vdots\pi_2\\ \Gamma_1 \vdash u : [P \multimap Q] & \Gamma_2 \vdash s : P\end{array}}{\Gamma \vdash t : Q}\ @$$

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

- *Step inside a β-redex, i.e. $t = (\lambda x.u)s \to_{\beta_v^\flat} (\lambda x.u')s = t'$ with $u \to_{\beta_v^\flat} u'$*: then,

$$\pi' = \dfrac{\dfrac{\begin{array}{c}\vdots\pi'_1\\ \Gamma_1, x : P \vdash u' : Q\end{array}}{\Gamma_1 \vdash \lambda x.u' : [P \multimap Q]}\ \lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s : P\end{array}}{\Gamma \vdash t' : Q}\ @$$

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

$$\pi = \dfrac{\dfrac{\begin{array}{c}\vdots\pi_1\\ \Gamma_1, x : P \vdash u : Q\end{array}}{\Gamma_1 \vdash \lambda x.u : [P \multimap Q]}\ \lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Delta \vdash s : P\end{array}}{\Gamma \vdash t : Q}\ @$$

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

2. Since $\to_{\sigma^\flat}$ cannot reduce to a value, t' is not a value and thus $|\pi'| > 0$ according to Remark 27. The proof that there exists $\pi \rhd \Gamma \vdash t : Q$ with $|\pi| = |\pi'|$ is by induction con $t \in \Lambda$. Cases:
   - *Step at the root*: there are two sub-cases:
     - $t = (\lambda x.u)sr \mapsto_{\sigma_1} (\lambda x.ur)s = t'$ with $x \notin \mathsf{fv}(r)$. We can suppose without loss of generality that $x \notin \mathsf{dom}(\Gamma)$. So,

$$\pi' = \dfrac{\dfrac{\dfrac{\begin{array}{cc}\vdots\pi_1 & \vdots\pi'_3\\ \Gamma_1, x : P \vdash u : [Q' \multimap Q] & \Gamma_3, x : [] \vdash r : Q'\end{array}}{\Gamma_1 \uplus \Gamma_3, x : P \vdash ur : Q}\ @}{\Gamma_1 \uplus \Gamma_3 \vdash \lambda x.ur : [P \multimap Q]}\ \lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s : P\end{array}}{\Gamma \vdash t' : Q}\ @$$

     with $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2) = \mathsf{dom}(\Gamma_3)$, $\Gamma = \Gamma_1 \uplus \Gamma_2 \uplus \Gamma_3$ and $|\pi'| = |\pi_1| + |\pi_2| + |\pi'_3| + 2$. According to Lemma 29.2, since $x \notin \mathsf{fv}(r)$, there is a derivation $\pi_3 \rhd \Gamma_3 \vdash r : Q'$ such that $|\pi_3| = |\pi'_3|$. Therefore, there is a derivation

$$\pi = \dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\pi_1\\ \Gamma_1, x : P \vdash u : [Q' \multimap Q]\end{array}}{\Gamma_1 \vdash \lambda x.u : [P \multimap [Q' \multimap Q]]}\ \lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s : P\end{array}}{\Gamma_1 \uplus \Gamma_2 \vdash t : [Q' \multimap Q]}\ @ \qquad \begin{array}{c}\vdots\pi_3\\ \Gamma_3 \vdash r : Q'\end{array}}{\Gamma \vdash t : Q}\ @$$

     where $|\pi| = |\pi_1| + |\pi_3| + |\pi_2| + 2 = |\pi_1| + |\pi'_3| + |\pi_2| + 2 = |\pi'|$.

– $t = v((\lambda x.u)s) \mapsto_{\sigma_3} (\lambda x.vu)s = t'$ with $x \notin \mathsf{fv}(v)$. We can suppose without loss of generality that $x \notin \mathsf{dom}(\Gamma)$. Therefore,

$$\pi' = \dfrac{\dfrac{\dfrac{\vdots\pi'_1 \qquad \vdots\pi_3}{\Gamma_1, x\colon [\,] \vdash v\colon [Q' \multimap Q] \qquad \Gamma_3, x\colon P \vdash u\colon Q'}\,@}{\Gamma_1 \uplus \Gamma_3, x\colon P \vdash vu\colon Q}\,\lambda \qquad \vdots\pi_2}{\Gamma_1 \uplus \Gamma_3 \vdash \lambda x.ur\colon [P \multimap Q] \qquad \Gamma_2 \vdash s\colon P}\,@}{\Gamma \vdash t'\colon Q}$$

with $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2) = \mathsf{dom}(\Gamma_3)$, $\Gamma = \Gamma_1 \uplus \Gamma_2 \uplus \Gamma_3$ and $|\pi'| = |\pi_1| + |\pi_2| + |\pi_3| + 2$. According to Lemma 29.2, since $x \notin \mathsf{fv}(v)$, there is a derivation $\pi_1 \rhd \Gamma_1 \vdash v\colon [Q' \multimap Q]$ such that $|\pi_1| = |\pi'_1|$. Thus, there is a derivation

$$\pi = \dfrac{\dfrac{\vdots\pi_1 \qquad \dfrac{\dfrac{\vdots\pi_2}{\Gamma_2, x\colon P \vdash u\colon Q'}\,\lambda \qquad \vdots\pi_3}{\Gamma_2 \vdash \lambda x.u\colon [P \multimap Q'] \qquad \Gamma_3 \vdash s\colon P}\,@}{\Gamma_1 \vdash v\colon [Q' \multimap Q] \qquad \Gamma_2 \uplus \Gamma_3 \vdash (\lambda x.u)s\colon Q'}\,@}{\Gamma \vdash t\colon Q}\,@$$

where $|\pi| = |\pi_1| + |\pi_3| + |\pi_2| + 2 = |\pi_1| + |\pi'_3| + |\pi_2| + 2 = |\pi'|$.

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

$$\pi' = \dfrac{\vdots\pi'_1 \qquad \vdots\pi_2}{\Gamma_1 \vdash u'\colon [P \multimap Q] \qquad \Gamma_2 \vdash s\colon P}\,@$$

where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi'| = |\pi'_1| + |\pi_2| + 1$. By induction hypothesis, there exists $\pi_1 \rhd \Gamma_1 \vdash u'\colon [P \multimap Q]$ with $|\pi_1| = |\pi'_1|$. So, there is

$$\pi = \dfrac{\vdots\pi_1 \qquad \vdots\pi_2}{\Gamma_1 \vdash u\colon [P \multimap Q] \qquad \Gamma_2 \vdash s\colon P}\,@$$

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

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

$$\pi' = \dfrac{\vdots\pi_1 \qquad \vdots\pi'_2}{\Gamma_1 \vdash u\colon [P \multimap Q] \qquad \Gamma_2 \vdash s'\colon P}\,@$$

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

$$\pi = \dfrac{\vdots\pi_1 \qquad \vdots\pi_2}{\Gamma_1 \vdash u\colon [P \multimap Q] \qquad \Gamma_2 \vdash s\colon P}\,@$$

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

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

$$\pi' = \dfrac{\dfrac{\begin{array}{c}\vdots\pi_1'\\ \Gamma_1, x\colon P \vdash u'\colon Q\end{array}}{\Gamma_1 \vdash \lambda x.u'\colon [P \multimap Q]}\,\lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s\colon P\end{array}}{\Gamma \vdash t'\colon Q}\,@$$

where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \mathsf{dom}(\Gamma_2)$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $|\pi'| = |\pi_1| + |\pi_2| + 1$. By induction hypothesis, there exists $\pi_1 \rhd \Gamma_1, x\colon P \vdash u\colon Q$ with $|\pi_1| = |\pi_1'|$. So, there is

$$\pi = \dfrac{\dfrac{\begin{array}{c}\vdots\pi_1\\ \Gamma_1, x\colon P \vdash u\colon Q\end{array}}{\Gamma_1 \vdash \lambda x.u\colon [P \multimap Q]}\,\lambda \qquad \begin{array}{c}\vdots\pi_2\\ \Gamma_2 \vdash s\colon P\end{array}}{\Gamma \vdash t\colon Q}\,@$$

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 $\pi \rhd \Gamma \vdash t\colon Q$.*

1. Shrinkage under β_v-step: *If $t \to_{\beta_v} t'$ then there is $\pi' \rhd \Gamma \vdash t'\colon Q$ with $|\pi| \geq |\pi'|$.*
2. Size invariance under σ-step: *If $t \to_\sigma t'$ then there is $\pi' \rhd \Gamma \vdash t'\colon Q$ 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 $\to_{\mathsf{sh}}$ reduces under λ's:

- *Abstraction, i.e. $t = \lambda x.u \to_{\mathsf{r}} \lambda x.u' = t'$ with $u \to_{\mathsf{r}} u'$ and $\mathsf{r} \in \{\beta_v, \sigma\}$*: then,

$$\pi = \dfrac{\begin{array}{c}\vdots\pi_1\\ \Gamma_1, x\colon P_1 \vdash u\colon Q_1\end{array} \quad {}^{n\in\mathbb{N}}\cdots \quad \begin{array}{c}\vdots\pi_n\\ \Gamma_n, x\colon P_n \vdash u\colon Q_n\end{array}}{\Gamma \vdash t\colon [P_1 \multimap Q_1, \ldots, P_n \multimap Q_n]}\,\lambda$$

where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \cdots = \mathsf{dom}(\Gamma_n)$, $\Gamma = \biguplus_{i=1}^n \Gamma_i$ and $|\pi| = \sum_{i=1}^n |\pi_i|$. By induction hypothesis, for all $1 \leq i \leq n$ there is $\pi_i' \rhd \Gamma_i, x\colon P_i \vdash u'\colon Q_i$ with $|\pi_i| \geq |\pi_i'|$ if $\mathsf{r} = \beta_v$, and $|\pi_i| = |\pi_i'|$ if $\mathsf{r} = \sigma$. So, there is

$$\pi' = \dfrac{\begin{array}{c}\vdots\pi_1'\\ \Gamma_1, x\colon P_1 \vdash u'\colon Q_1\end{array} \quad \ldots \quad \begin{array}{c}\vdots\pi_n'\\ \Gamma_n, x\colon P_n \vdash u'\colon Q_n\end{array}}{\Gamma \vdash t'\colon [P_1 \multimap Q_1, \ldots, P_n \multimap Q_n]}\,\lambda$$

where $|\pi'| = \sum_{i=1}^n |\pi_i'|$. Therefore, $|\pi| \geq |\pi'|$ if $\mathsf{r} = \beta_v$, and $|\pi| = |\pi'|$ if $\mathsf{r} = \sigma$. □

See p. 7 **Lemma 12** (Subject expansion)**.** *Let $t, t' \in \Lambda$ and $\pi' \rhd \Gamma \vdash t'\colon Q$.*

1. Enlargement under anti-β_v-step: *If $t \to_{\beta_v} t'$ then there is $\pi \rhd \Gamma \vdash t\colon Q$ with $|\pi| \geq |\pi'|$.*
2. Size invariance under anti-σ-step: *If $t \to_\sigma t'$ then there is $\pi \rhd \Gamma \vdash t\colon Q$ 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 $\to_{\mathsf{sh}}$ reduces under λ's:

- *Abstraction*, *i.e.* $t = \lambda x.u \to_{\mathsf{r}} \lambda x.u' = t'$ with $u \to_{\mathsf{r}} u'$ and $\mathsf{r} \in \{\beta_v, \sigma\}$: then,

$$\pi' = \dfrac{\begin{array}{ccc}\vdots\pi'_1 & & \vdots\pi'_n\\ \Gamma_1, x: P_1 \vdash u': Q_1 & \overset{n\in\mathbb{N}}{\cdots} & \Gamma_n, x: P_n \vdash u': Q_n\end{array}}{\Gamma \vdash t': [P_1 \multimap Q_1, \dots, P_n \multimap Q_n]}\ \lambda$$

where $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma_1) = \cdots = \mathsf{dom}(\Gamma_n)$, $\Gamma = \biguplus_{i=1}^{n} \Gamma_i$ and $|\pi'| = \sum_{i=1}^{n} |\pi'_i|$. By induction hypothesis, for all $1 \le i \le n$ there is $\pi_i \rhd \Gamma_i, x: P_i \vdash u: Q_i$ with $|\pi_i| \ge |\pi'_i|$ if $\mathsf{r} = \beta_v$, and $|\pi_i| = |\pi'_i|$ if $\mathsf{r} = \sigma$. So, there is

$$\pi = \dfrac{\begin{array}{ccc}\vdots\pi_1 & & \vdots\pi_n\\ \Gamma_1, x: P_1 \vdash u: Q_1 & \dots & \Gamma_n, x: P_n \vdash u: Q_n\end{array}}{\Gamma \vdash t: [P_1 \multimap Q_1, \dots, P_n \multimap Q_n]}\ \lambda$$

where $|\pi| = \sum_{i=1}^{n} |\pi_i|$. Therefore, $|\pi| \ge |\pi'|$ if $\mathsf{r} = \beta_v$, and $|\pi| = |\pi'|$ if $\mathsf{r} = \sigma$. □

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 $\vec{x} = (x_1, \dots, x_n)$ be a suitable list of variables for t and u. If $t \simeq_{\mathsf{sh}} u$ then $[\![t]\!]_{\vec{x}} = [\![u]\!]_{\vec{x}}$.* *See p. 8*

Proof. Since $t \simeq_{\mathsf{sh}} u$, there exist $k \in \mathbb{N}$ and $t_0, \dots, t_k$ such that $t = t_0$, $u = t_k$ and $t_i \to_{\mathsf{sh}} t_{i+1}$ or $t_{i+1} \to_{\mathsf{sh}} t_i$, for all $0 \le i < k$. Using subject reduction (Lemma 11) and subject expansion (Lemma 12), it is immediate to prove by induction on $k \in \mathbb{N}$ that $[\![t]\!]_{\vec{x}} = [\![u]\!]_{\vec{x}}$. □

Lemma 15 (Semantics of $\mathsf{sh}^\flat$-normal forms)**.** *Let t be a term, let $k \in \mathbb{N}$ and let $\vec{x} = (x_1, \dots, x_k)$ be a list of variables suitable for t.* *See p. 8*

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

Proof. Point 3 is an immediate consequence of Point 2 via the syntactic characterization of $\mathsf{sh}^\flat$-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 $\vec{x}$ is suitable for t, one has $x = x_i$ for some $1 \le i \le k$. According to Cor. 28 there exists a derivation $\pi' \rhd x_1: [], \dots, x_k: [] \vdash v: []$, so for any positive type Q there exists the derivation

$$\pi = \dfrac{\dfrac{}{x_1: [], \dots, x_i: [[] \multimap Q], \dots, x_k: [] \vdash x_i: [[] \multimap Q]}\ \mathsf{ax} \qquad \begin{array}{c}\vdots\pi'\\ x_1: [], \dots, x_k: [] \vdash v: []\end{array}}{x_1: [], \dots, x_i: [[] \multimap Q], \dots, x_k: [] \vdash t: Q}\ @$$

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

$$\pi = \dfrac{\dfrac{}{x_1 : [\,], \ldots, x_i : [[\,] \multimap Q], \ldots, x_k : [\,] \vdash x_i : [[\,] \multimap Q]}\,\mathsf{ax} \qquad \begin{array}{c}\vdots \pi' \\ x_1 : P_1, \ldots, x_k : P_k \vdash a : [\,]\end{array}}{x_1 : P_1, \ldots, x_i : [[\,] \multimap Q] \uplus P_i, \ldots, x_k : P_k \vdash t : Q}\,@$$

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

$$\pi = \dfrac{\begin{array}{c}\vdots \pi' \\ x_1 : P'_1, \ldots, x_k : P'_k \vdash a : [P \multimap Q]\end{array} \qquad \begin{array}{c}\vdots \pi'' \\ x_1 : P_1, \ldots, x_k : P_k \vdash n : P\end{array}}{x_1 : P_1 \uplus P'_1, \ldots, x_k : P_k \uplus P'_k \vdash t : Q}\,@ .$$

Cases for $t \in \Lambda_n$:

- $t \in \Lambda_a$: see above.
- t is a value: the statement follows from Cor. 28.
- $t = (\lambda x.n)a$ for some $n \in \Lambda_n$ and $a \in \Lambda_a$: by *i.h.* applied to n, there exists a derivation $\pi' \rhd x_1 : P'_1, \ldots, x_k : P'_k, x : P \vdash n : Q$ for some positive types $P'_1, \ldots, P'_k, P, Q$. By *i.h.* applied to a, there exists a derivation $\pi'' \rhd x_1 : P_1, \ldots, x_k : P_k \vdash a : P$ for some positive types $P_1, \ldots, P_k$. Therefore, there is a derivation

$$\pi = \dfrac{\dfrac{\begin{array}{c}\vdots \pi' \\ x_1 : P'_1, \ldots, x_k : P'_k, x : P \vdash n : Q\end{array}}{x_1 : P'_1, \ldots, x_k : P'_k \vdash \lambda x.n : [P \multimap Q]}\,\lambda \qquad \begin{array}{c}\vdots \pi'' \\ x_1 : P_1, \ldots, x_k : P_k \vdash a : P\end{array}}{x_1 : P_1 \uplus P'_1, \ldots, x_k : P_k \uplus P'_k \vdash t : Q}\,@ .$$

□

See p. 8 **Theorem 16** (Semantic and logic characterization of $\mathsf{sh}^\flat$-normalization)**.** *Let $t \in \Lambda$ and let $\vec{x} = (x_1, \ldots, x_n)$ be a suitable list of variables for t. The following are equivalent:*

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

Proof. (1)⇒(2)**:** Trivial, since $\to_{\mathsf{sh}^\flat} \subseteq \simeq_{\mathsf{sh}}$.

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

(3)⇒(4) Trivial, according to Definition 13.

(4)⇒(5)**:** If there is a derivation $\pi \rhd x_1 : P_1, \dots, x_n : P_n \vdash t : Q$ for some positive types $P_1, \dots, P_n, Q$, then every $\mathsf{sh}^\flat$-reduction sequence from t has at most $|\pi| \in \mathbb{N}$ $\beta_v^\flat$-reduction steps by the quantitative subject reduction (Prop. 8.1-2). As there is no $\mathsf{sh}^\flat$-reduction sequence from t with infinitely many $\beta_v^\flat$-reduction steps, then every infinite $\mathsf{sh}^\flat$-reduction sequence from t would have infinitely many $\sigma^\flat$-reduction steps, but this is impossible since $\to_{\sigma^\flat}$ is strongly normalizing. Therefore, there is no infinite $\mathsf{sh}^\flat$-reduction sequence from t, which means that t is strongly $\mathsf{sh}^\flat$-normalizable.

(5)⇒(1)**:** Trivial. □

Lemma 17 (Uniqueness of the derivation with empty types; Logic and semantic characterization of values)**.** *Let $t \in \Lambda$ be $\mathsf{sh}^\flat$-normal and $\vec{x} = (x_1, \dots, x_k)$ be suitable for t.* *See p. 9*

1. *For every $\pi \rhd x_1 : [], \dots, x_k : [] \vdash t : []$ and $\pi' \rhd x_1 : [], \dots, x_k : [] \vdash t : []$, one has $t \in \Lambda_v$, $|\pi| = 0$ and $\pi = \pi'$. More precisely, π consists only of the rule* ax *if t is a variable, otherwise t is an abstraction and π consists only of a 0-ary rule λ.*
2. *The following are equivalent:*

   (a) t is a value;
   (b) $((\overset{k}{[]}, \dots, []), []) \in [\![t]\!]_{\vec{x}}$;
   (c) there is $\pi \rhd x_1 : [], \dots, x_k : [] \vdash t : []$;
   (d) there is $\pi \rhd t$ such that $|\pi| = 0$.

Proof. 1. According to Prop. 3, $t \in \Lambda_n$ since t is $\mathsf{sh}^\flat$-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 $1 \le i \le k$, and the only derivation with conclusion $x_1 : [], \dots, x_k : [] \vdash t : []$ is

$$\pi = \frac{}{x_1 : [], \dots, x_k : [] \vdash x_i : []}\ \mathsf{ax};$$

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

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

If $t \in \Lambda_a$ then $t \notin \Lambda_v$ and it is impossible that $\pi \rhd x_1 : [], \dots, x_k : [] \vdash t : []$ by Lemma 30.
Finally, if $t \in \Lambda_n \setminus (\Lambda_v \cup \Lambda_a)$, then $t = (\lambda x.n)a$ for some $n \in \Lambda_n$ and $a \in \Lambda_a$. By necessity,

$$\pi = \frac{\begin{array}{cc} \vdots \pi' & \vdots \pi'' \\ x_1 : [], \dots, x_k : [] \vdash \lambda x.n : P \multimap [] & x_1 : [], \dots, x_k : [] \vdash a : P \end{array}}{x_1 : [], \dots, x_k : [] \vdash t : []}\ @$$

but it is impossible that $\pi \rhd x_1 : [], \dots, x_k : [] \vdash a : P$, according to Lemma 30. Therefore, it is impossible that $\pi \rhd x_1 : [], \dots, x_k : [] \vdash t : []$.

2. The equivalence (2b)⇔(2c) follows immediately from Definition 13. From Lemma 17.1 it follows that (2c)⇒(2d). By Cor. 28, the implication (2a)⇒(2c) holds. In order to prove that (2d)⇒(2a), it is enough to notice that there is no instance of the rule @ in $\pi \rhd t$ since $|\pi| = 0$, so t is either a variable or an abstraction, *i.e.* a value. □

Proposition 18 (Logic and semantic characterization of valuability)**.** *Let t be a term and $\vec{x} = (x_1, \dots, x_k)$ be suitable for t. The following are equivalent:* *See p. 9*

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

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

- either t is not $\mathsf{sh}^\flat$-normalizable, and then $[\![t]\!]_{\vec{x}} = \emptyset$ according to the semantic characterization of $\mathsf{sh}^\flat$-normalization (Thm. 16), in particular $(([],\overset{k}{\ldots},[]),[]) \notin [\![t]\!]_{\vec{x}}$;
- or t is $\mathsf{sh}^\flat$-normalizable; let t_0 be its $\mathsf{sh}^\flat$-normal form; according to the semantic characterization of values (Lemma 17.2), $(([],\overset{k}{\ldots},[]),[]) \in [\![t_0]\!]_{\vec{x}}$ iff t_0 is a value; by invariance of the semantics (Thm. 14), $[\![t]\!]_{\vec{x}} = [\![t_0]\!]_{\vec{x}}$; therefore, $(([],\overset{k}{\ldots},[]),[]) \in [\![t]\!]_{\vec{x}}$ iff t_0 is a value. □

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 $\mathsf{sh}^\flat$-normal then $|t|_\flat = \inf\{|\pi| \mid \pi \rhd x_1 : P_1, \ldots, x_k : P_k \vdash t : Q\}$.*
2. *If t is a value then $|t|_\flat = \inf\{|\pi| \mid \pi \rhd x_1 : P_1, \ldots, x_k : P_k \vdash t : Q\} = 0$.*

Proof. 1. By Prop. 3, since t is $\mathsf{sh}^\flat$-normal, we can proceed by induction on $t \in \Lambda_n$. Moreover, for $t \in \Lambda_a$ we prove also that for *any* positive type Q there exist positive types $P_1, \ldots, P_k$ and a derivation $\pi \rhd x : P_1, \ldots, x_k : P_k \vdash t : Q$ such that $|t|_\flat = |\pi|$: this stronger statement is required to handle the case where t is a $\mathsf{sh}^\flat$-normal β-redex.

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

If $t \in \Lambda_a$, then there are three cases:

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

$$\pi = \dfrac{\dfrac{}{x_1 : [], \ldots, x_i : [[] \multimap Q], \ldots, x_k : [] \vdash x_i : [[] \multimap Q]}\,\mathsf{ax} \qquad \begin{matrix}\vdots \pi' \\ x_1 : [], \ldots, x_k : [] \vdash v : []\end{matrix}}{x_1 : [], \ldots, x_i : [[] \multimap Q], \ldots, x_k : [] \vdash t : Q}\,@$$

  where $|\pi| = |\pi'| + 1 = 1 = |t|_\flat$. The last rule of any derivation $\pi'' \rhd x_1 : P_1, \ldots, x_k : P_k \vdash t : Q$ is by necessity @, thus $|\pi''| \ge 1$ and hence $\inf\{|\pi''| \mid \pi'' \rhd x_1 : P_1, \ldots, x_k : P_k \vdash t : Q\} = 1 = |t|_\flat$.
- $t = xa$ for some variable x and $a \in \Lambda_a$: $|t|_\flat = |a|_\flat + 1$, $k > 0$ and $x = x_i$ for some $1 \le i \le k$. By *i.h.*, $|a|_\flat = \inf\{|\pi'| \mid \pi' \rhd x_1 : P_1, \ldots, x_k : P_k \vdash a : []\}$ and there exists a derivation $\pi' \rhd x_1 : P_1, \ldots, x_k : P_k \vdash a : []$ for some positive types $P_1, \ldots, P_k$ such that $|a|_\flat = |\pi'|$. Therefore, for any positive type Q there exists a derivation

$$\pi = \dfrac{\dfrac{}{x_1 : [], \ldots, x_i : [[] \multimap Q], \ldots, x_k : [] \vdash x_i : [[] \multimap Q]}\,\mathsf{ax} \qquad \begin{matrix}\vdots \pi' \\ x_1 : P_1, \ldots, x_k : P_k \vdash a : []\end{matrix}}{x_1 : P_1, \ldots, x_i : [[] \multimap Q] \uplus P_i, \ldots, x_k : P_k \vdash t : Q}\,@$$

  where $|\pi| = |\pi'| + 1 = |a|_\flat + 1 = |t|_\flat$. The last rule of any derivation $\pi'' \rhd x_1 : P_1, \ldots, x_k : P_k \vdash t : Q$ is by necessity @ having a derivation typing a as a premise, thus $|\pi''| \ge |a|_\flat + 1$ and hence $\inf\{|\pi''| \mid \pi'' \rhd x_1 : P_1, \ldots, x_k : P_k \vdash t : Q\} = |a|_\flat + 1 = |t|_\flat$.

- $t = an$ with $a \in \Lambda_a$ and $n \in \Lambda_n$: by *i.h.* applied to n, $\inf\{|\pi''| \mid \pi'' \rhd x_1 : P_1, \dots, x_k : P_k \vdash n : Q\} = |n|_\flat$, in particular there is a derivation $\pi'' \rhd x_1 : P_1, \dots, x_k : P_k \vdash n : P$ for some positive types $P, P_1, \dots, P_k$ such that $|\pi| = |n|_\flat$. For any positive type Q, by *i.h.* applied to a, there are positive types $P'_1, \dots, P'_k$ and a derivation $\pi' \rhd x_1 : P'_1, \dots, x_k : P'_k \vdash a : [P \multimap Q]$ such that $|a|_\flat = |\pi'| = \inf\{|\pi'| \mid \pi' \rhd x_1 : P'_1, \dots, x_k : P'_k \vdash a : [P \multimap Q]\}$. So, there is a derivation

$$\pi = \dfrac{\begin{array}{cc} \vdots \pi' & \vdots \pi'' \\ x_1 : P'_1, \dots, x_k : P'_k \vdash a : [P \multimap Q] & x_1 : P_1, \dots, x_k : P_k \vdash n : P \end{array}}{x_1 : P_1 \uplus P'_1, \dots, x_k : P_k \uplus P'_k \vdash t : Q}\ @$$

  where $|\pi| = |\pi'| + |\pi''| + 1 = |a|_\flat + |n|_\flat + 1 = |t|_\flat$ (the last equation holds since a is not an abstraction, see Prop. 3). The last rule of any derivation $\pi''' \rhd x_1 : P_1, \dots, x_k : P_k \vdash t : Q$ is by necessity @ having derivations typing a and n as premises, thus $|\pi'''| \geq |a|_\flat + |n|_\flat + 1$ and hence $\inf\{|\pi'''| \mid \pi''' \rhd x_1 : P_1, \dots, x_k : P_k \vdash t : Q\} = |a|_\flat + |n|_\flat + 1 = |t|_\flat$.

Finally, if $t = (\lambda x.n)a$ for some $a \in \Lambda_a$ and $n \in \Lambda_n$, then $|t|_\flat = |n|_\flat + |a|_\flat + 1$ by definition. By *i.h.*, applied to n, there is a derivation $\pi' \rhd x_1 : P'_1, \dots, x_k : P'_k, x : P \vdash n : Q$ for some positive types $P'_1, \dots, P'_k, P, Q$ such that $|n|_\flat = |\pi'| = \inf\{|\pi'| \mid \pi' \rhd x_1 : P'_1, \dots, x_k : P'_k \vdash n : Q\}$. By *i.h.* applied to a, there exists a derivation $\pi'' \rhd x_1 : P_1, \dots, x_k : P_k \vdash a : P$ for some positive types $P_1, \dots, P_k$ such that $|a|_\flat = |\pi''| = \inf\{|\pi''| \mid \pi'' \rhd x_1 : P_1, \dots, x_k : P_k \vdash a : P\}$. Therefore, there is a derivation

$$\pi = \dfrac{\dfrac{\vdots \pi'}{x_1 : P'_1, \dots, x_k : P'_k, x : P \vdash n : Q}}{x_1 : P'_1, \dots, x_k : P'_k \vdash \lambda x.n : [P \multimap Q]}\ \lambda \qquad \dfrac{\vdots \pi''}{x_1 : P_1, \dots, x_k : P_k \vdash a : P}\ @$$

  where $|\pi| = |\pi'| + |\pi''| + 1 = |n|_\flat + |a|_\flat + 1 = |t|_\flat$. Given a derivation $\pi''' \rhd x_1 : P_1, \dots, x_k : P_k \vdash t : Q$, its last rule is by necessity @ having derivations typing a and $\lambda x.n$ as premises (the last rule of the latter is necessarily λ having a derivation typing n as unique premise), thus $|\pi'''| \geq |a|_\flat + |n|_\flat + 1$ and hence $\inf\{|\pi'''| \mid \pi''' \rhd x_1 : P_1, \dots, x_k : P_k \vdash t : Q\} = |a|_\flat + |n|_\flat + 1 = |t|_\flat$.

2. By definition, $|t|_\flat = 0$. According to Cor. 28, there is a derivation $\pi \rhd x_1 : [], \dots, x_k : [] \vdash t : []$ such that $|\pi| = 0$, hence $\inf [\![t]\!]_{\vec{x}} = |(([], \dots, []), [])| = 0$. □

Proposition 21 (Exact number of $\beta_v^\flat$-steps)**.** *Let t be a $\mathsf{sh}^\flat$-normalizable term and t_0 be its $\mathsf{sh}^\flat$-normal form.* *See p. 10* *For every reduction sequence $d : t \to_{\mathsf{sh}^\flat}^* t_0$ and every $\pi \rhd t$ and $\pi_0 \rhd t_0$ such that $|\pi| = \inf\{|\pi'| \mid \pi' \rhd t\}$ and $|\pi_0| = \inf\{|\pi'_0| \mid \pi'_0 \rhd t_0\}$, one has*

$$\mathrm{leng}_{\beta_v^\flat}(d) = |\pi| - |t_0|_\flat = |\pi| - |\pi_0|\,. \tag{3}$$

If moreover t_0 is a value, then $\mathrm{leng}_{\beta_v^\flat}(d) = |\pi|$.

Proof. The statement concerning the case where t_0 is a value is an immediate consequence of Eq. (3) and Lemma 20.2. The second identity in Eq. (3) follows immediately from Lemma 20.1. We prove the first identity in Eq. (3) by induction on $n = \mathrm{leng}(d) \in \mathbb{N}$.

If $n = 0$ then $\mathrm{leng}_{\beta_v^\flat}(d) = 0$ and $t = t_0$ (thus $\pi = \pi_0$), hence $|\pi| = |t|_\flat = |t_0|_\flat$ according to Lemma 20.1, and therefore Eq. (2) holds.

If $n > 0$ then d has the form $t \to_{\mathsf{sh}^\flat} t' \to_{\mathsf{sh}^\flat}^{n-1} t_0$ for some term t'; let d' be the sub-reduction sequence $t' \to_{\mathsf{sh}^\flat}^{n-1} t_0$ in d. There are two cases:

- *A β_v-step* at the beginning of d, *i.e.* $t \to_{\beta_v^\flat} t'$: according to the quantitative subject reduction for $\to_{\beta_v^\flat}$ (Prop. 8.1), there is a derivation $\pi' \rhd t'$ such that $|\pi'| = |\pi| - 2$. According to the quantitative subject expansion for $\to_{\beta_v^\flat}$ (Prop. 10.1), for any derivation $\pi'' \rhd t'$ there exists a derivation $\pi''' \rhd t$ such that $|\pi''| = |\pi'''| - 2$. Therefore, from the minimality of $|\pi|$ follows the minimality of $|\pi'|$ (among the derivations $\pi'' \rhd t'$). We can then apply the *i.h.* to d', so that $\mathrm{leng}_{\beta_v^\flat}(d') = |\pi'| - |t_0|_\flat = |\pi| - 1 - |t_0|_\flat$ and hence $\mathrm{leng}_{\beta_v^\flat}(d) = \mathrm{leng}_{\beta_v^\flat}(d') + 1 = |\pi| - |t_0|_\flat$.
- *A σ-step* at the beginning of d, *i.e.* $t \to_{\sigma^\flat} t'$: according to the quantitative subject reduction for $\to_{\sigma^\flat}$ (Prop. 8.2), there is a derivation $\pi' \rhd t'$ such that $|\pi'| = |\pi|$. According to the quantitative subject expansion for $\to_{\sigma^\flat}$ (Prop. 10.2), for any derivation $\pi'' \rhd t'$ there exists a derivation $\pi''' \rhd t$ such that $|\pi''| = |\pi'''|$. Therefore, from the minimality of $|\pi|$ follows the minimality of $|\pi'|$ (among the derivations $\pi'' \rhd t'$). We can then apply the *i.h.* to d', so that $\mathrm{leng}_{\beta_v^\flat}(d) = \mathrm{leng}_{\beta_v^\flat}(d') = |\pi'| - |t_0|_\flat = |\pi| - |t_0|_\flat$. □

Lemma 30. *Let $t \in \Lambda_a$. For all $\pi \rhd x_1 : P_1, \dots, x_k : P_k \vdash t : Q$ there is $1 \le i \le k$ with $P_i \neq [\,]$.*

Proof. By induction on $t \in \Lambda_a$.

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

$$\pi = \dfrac{\dfrac{}{x_1 : P'_1, \dots, x_i : [P \multimap Q], \dots, x_k : P'_k \vdash x : [P \multimap Q]}\,\mathsf{ax} \qquad \begin{matrix}\vdots \pi'' \\ x_1 : P''_1, \dots, x_k : P''_k \vdash u : P\end{matrix}}{x_1 : P_1, \dots, x_k : P_k \vdash t : Q}\,@$$

where $P_j = P'_j \uplus P''_j$ for all $1 \le j \le k$ such that $j \neq i$, and $P_i = [P \multimap Q] \uplus P''_i \neq [\,]$.

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

$$\pi = \dfrac{\begin{matrix}\vdots \pi' \\ x_1 : P'_1, \dots, x_k : P'_k \vdash a : [P \multimap Q]\end{matrix} \qquad \begin{matrix}\vdots \pi'' \\ x_1 : P''_1, \dots, x_k : P''_k \vdash n : P\end{matrix}}{x_1 : P_1, \dots, x_k : P_k \vdash t : Q}\,@$$

where $P_j = P'_j \uplus P''_j$ for all $1 \le j \le k$. By *i.h.*, there is $1 \le i \le k$ such that $P'_i \neq [\,]$, thus $P_i = P'_i \uplus P''_i \neq [\,]$. □

See p. 10 **Theorem 24** (Exact number of $\beta_v^\flat$-steps for valuables)**.** *Let $t \to_{\mathsf{sh}^\flat}^* v \in \Lambda_v$. For any $\vec{x} = (x_1, \dots, x_k)$ suitable for t, and any $\pi \rhd x_1 : [\,], \dots, x_k : [\,] \vdash t : [\,]$, one has $\mathrm{leng}_{\beta_v^\flat}(t) = |\pi|$.*

Proof. Given $\pi \rhd x_1 : [\,], \dots, x_k : [\,] \vdash t : [\,]$, there is $\pi_0 \rhd x_1 : [\,], \dots, x_k : [\,] \vdash v : [\,]$ with $|\pi| = |\pi_0| + \mathrm{leng}_{\beta_v^\flat}(t)$ by the quantitative subject reduction (Prop. 8.1-2). According to Lemma 17.1, $|\pi_0| = 0$. □