JFP **30**, e14, 101 pages, 2020. © The Author(s), 2020. Published by Cambridge University Press 1
doi:10.1017/S095679682000012X

Tight typings and split bounds, fully developed

BENIAMINO ACCATTOLI
Inria & LIX, École Polytechnique, UMR 7161, Palaiseau, France
(*e-mail:* beniamino.accattoli@inria.fr)

STÉPHANE GRAHAM-LENGRAND
SRI International, USA
(*e-mail:* stephane.graham-lengrand@csl.sri.com)

DELIA KESNER
IRIF, CNRS, Université de Paris & Institut Universitaire de France, France
(*e-mail:* kesner@irif.fr)

Abstract

Multi types – aka non-idempotent intersection types – have been used. to obtain quantitative bounds on higher-order programs, as pioneered by de Carvalho. Notably, they bound at the same time the number of evaluation steps *and* the size of the result. Recent results show that the number of steps can be taken as a reasonable time complexity measure. At the same time, however, these results suggest that multi types provide quite lax complexity bounds, because the size of the result can be exponentially bigger than the number of steps. Starting from this observation, we refine and generalise a technique introduced by Bernadet and Graham-Lengrand to provide *exact* bounds. Our typing judgements carry counters, one measuring evaluation lengths and the other measuring result sizes. In order to emphasise the modularity of the approach, we provide exact bounds for four evaluation strategies, both in the λ-calculus (head, leftmost-outermost, and maximal evaluation) and in the linear substitution calculus (linear head evaluation). Our work aims at both capturing the results in the literature and extending them with new outcomes. Concerning the literature, it unifies de Carvalho and Bernadet & Graham-Lengrand via a uniform technique and a complexity-based perspective. The two main novelties are exact split bounds for the leftmost strategy – the only known strategy that evaluates terms to full normal forms and provides a reasonable complexity measure – and the observation that the computing device hidden behind multi types is the notion of substitution at a distance, as implemented by the linear substitution calculus.

1 Introduction

Type systems enforce properties of programs, such as termination, deadlock-freedom, or productivity. This paper studies a class of type systems for the λ-calculus that refines termination by providing exact bounds for evaluation lengths and normal forms.

Intersection types and multi types. One of the cornerstones of the theory of λ-calculus is that intersection types *characterise* termination: not only typed programs terminate, but all terminating programs are typable as well (Coppo & Dezani-Ciancaglini, 1978, 1980; Pottinger, 1980; Krivine, 1993). In fact, the λ-calculus comes with different notions of evaluation (e.g. call-by-name, call-by-value, and call-by-need) to different notions of normal forms (head/weak/full, etc.) and, accordingly, with different systems of intersection types.

Intersection types are a flexible tool and, even when one fixes a particular notion of evaluation and normal form, the type system can be formulated in various ways. A flavour that became quite convenient in the last 10 years is that of *non-idempotent* intersection types, where the intersection $A \cap A$ is not equivalent to A. They first appeared in Gardner (1994), Kfoury (2000), and Neergaard & Mairson (2004) but it is the seminal work of de Carvalho (2007, 2018),[1] who found fundamental uses of non-idempotency to characterise *quantitative* properties of λ-calculus, stressing their importance. Roughly, distinguishing $A \cap A$ from A allows counting resource consumption. A survey can be found in Bucciarelli *et al.* (2017).

Non-idempotent intersections can be seen as multi-sets, which is why, to ease the language, we prefer to call them *multi types* rather than *non-idempotent intersection types*. Multi types have two main features:

1. *Bounds on evaluation lengths*. They go beyond simply qualitative characterisations of termination, as typing derivations provide quantitative bounds on the length of evaluation (i.e. on the number of β-steps) and on the size of normal forms. Therefore, they give *intentional* insights on programs, and seem to provide a tool to reason about the complexity of programs.
2. *Linear logic interpretation*. Multi types are deeply linked to linear logic. The relational model (Girard, 1988; Bucciarelli & Ehrhard, 2001) of linear logic (often considered as a sort of canonical model of linear logic) is based on multi-sets, and multi types can be seen as a syntactic presentation of the relational model of the λ-calculus induced by the interpretation into linear logic.

These two facts together have a potential, fascinating consequence: they suggest that denotational semantics may provide abstract tools for complexity analyses, that are theoretically solid, being grounded on linear logic.

Various works in the literature explore the bounding power of multi types. Often, the bounding power is used *qualitatively*, i.e. without explicitly counting the number of steps to characterise termination and/or the properties of the induced relational model. Indeed, multi types provide combinatorial proofs of termination that are simpler than those developed for (idempotent) intersection types (e.g. reducibility technique). Several papers explore this approach under the call-by-name (Bucciarelli *et al.*, 2012; Kesner & Ventura, 2015; Kesner & Vial, 2017; Ong, 2017; Paolini *et al.*, 2017) or the call-by-value (Ehrhard, 2012; Díaz-Caro *et al.*, 2013; Carraro & Guerrieri, 2014) operational semantics, or both (Ehrhard & Guerrieri, 2016). Sometimes, precise *quantitative* bounds are provided instead, as in de Carvalho (2018) and Bernadet & Graham-Lengrand (2013b). Multi types can also be used to provide characterisation of complexity classes (Benedetti & Ronchi Della Rocca, 2016). Other qualitative (de Carvalho, 2016; Guerrieri *et al.*, 2016) and quantitative (de Carvalho *et al.*, 2011; de Carvalho & Tortora de Falco, 2016) studies are also sometimes done in the more general context of linear logic, rather than in the λ-calculus.

[1] de Carvalho's work published in 2018 is based on the well diffused technical report (de Carvalho, 2009).

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Reasonable cost models. Usually, the quantitative studies define a measure for typing derivations and show that the measure provides a bound on the length of evaluation sequences for typed terms. A criticism that could be raised against these results is, or rather was, that the number of β-steps of the bounded evaluation strategies might not be a reasonable cost model, that is, it might not be a reliable complexity measure. This is because no reasonable cost models for the λ-calculus were known at the time. But the understanding of cost models for the λ-calculus made significant progress in the last few years. Since the nineties, it is known that the number of steps for *weak* strategies (i.e. not reducing under abstraction) is a reasonable cost model (Blelloch & Greiner, 1995), where *reasonable* means polynomially related to the cost model of Turing machines. It is only in 2014, that a solution for the general case has been obtained: the length of leftmost-outermost evaluation to normal form was shown to be a reasonable cost model in Accattoli & Dal Lago (2016). In this work, we essentially update the study of the bounding power of multi types with the insights coming from the study of reasonable cost models. In particular, we provide new answers to the question of whether denotational semantics can really be used as an accurate tool for complexity analyses.

Size explosion and lax bounds. The study of cost models made clear that evaluation lengths are independent from the size of their results. The scepticism about taking the number of β-steps as a reliable complexity measure comes from the *size explosion problem*, that is, the fact that the size of terms can grow exponentially with respect to the number of β-steps. When λ-terms are used to encode decision procedures, the normal forms (encoding true or false) are of constant size, and therefore, there is no size explosion issue.[2] But when λ-terms are used to compute other normal forms than Boolean values, there are families of terms $\{t_n\}_{n\in\mathbb{N}}$, where t_n has size linear in n, it evaluates to normal form in n β-steps, and produces a result p_n of size $\Omega(2^n)$, i.e. exponential in n. Moreover, the size explosion problem is extremely robust, as there are families for which the size explosion is independent of the evaluation strategy. The difficulty in proving that the length of a given strategy provides a reasonable cost model lies precisely in the fact that one needs a compact representation of normal forms, to avoid to fully compute them (because they can be huge and it would be too expensive). A gentle introduction to reasonable cost models and size explosion is Accattoli (2018a).

Now, multi typings do bound the number of β-steps of reasonable strategies, but these bounds are sometimes too generous since they bound at the same time the length of evaluations *and* the size of the normal forms. Therefore, even a notion of *minimal* typing (in the sense of being the smallest derivation) provides a bound that in some cases is exponentially worse than the number of β-steps.

Our observation is that the typings themselves are in fact much bigger than evaluation lengths, and so the widespread point of view for which multi types – and so the relational model of linear logic – faithfully capture evaluation lengths, or even the complexity, is misleading.

[2] To be precise, size explosion may happen also when the normal form is a Boolean: a term may make arguments grow exponentially in size and then erase them. Such a form of size explosion is an issue for implementations, but not for the topic of this paper.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

More precisely, multi typings do measure *part of* the size of the normal form, namely, the part concerned by the notion of evaluation that the typings are meant to measure. In the case of head evaluation, for instance, they measure the size of the *spine*, that is, the left branch of the term syntax tree, because head evaluation never enters arguments and so their size is not taken into account. Notably, the size of the spine never explodes, even on families of terms whose size explodes via head evaluation. In the case of leftmost evaluation, however, multi typings measure the whole size of the term, which does explode. Therefore, the inaccuracy of the measurement depends on the notion of evaluation under study.

1.1 Contributions

The tightening technique. Our starting point is a technique introduced in a technical report by Bernadet & Graham-Lengrand (2013a). They study the case of maximal evaluation and present a multi-type system where typing derivations of terms provide an *upper bound* on the number of β-steps to normal form. More interestingly, they show that every strongly normalising term admits a typing derivation that is sufficiently tight, where the obtained bound is *exactly* the length of the longest β-reduction path. This improved on previous results, e.g. Bernadet & Lengrand (2011) and Bernadet & Graham-Lengrand (2013b), where multi types provided the exact measure of longest evaluation paths *plus* the size of the normal forms which, as discussed above, can be exponentially bigger. Finally, they enrich the structure of base types so that, for those typing derivations providing the exact lengths, the type of a term gives the structure (and hence the size) of its normal form. This paper embraces this tightening technique, simplifying it with the use of *tight constants* for base types, and generalising it to a range of other evaluation strategies, described next.

Modular approach. We develop all our results by using a unique schema that modularly applies to different evaluation strategies. Our approach isolates the key concepts for the correctness and completeness of multi types, providing a powerful and modular technique, having at least two by-products. First, it reveals the relevance of *neutral terms* and of their properties with respect to types. Second, the concrete instantiations of the schema on four different cases always require subtle definitions, stressing the key conceptual properties of each case study.

Head and leftmost-outermost evaluation. The first application of the tightening technique is to the head and leftmost evaluation strategies (we often say simply *leftmost* instead of *leftmost-outermost*). The head case is the simplest possible one. The leftmost case is the natural iteration of the head one, and the only known strong strategy whose number of steps provides a reasonable cost model (Accattoli & Dal Lago, 2016). Multi types bounding the lengths of leftmost normalising terms have been also studied in Bucciarelli *et al*. (2017) and Kesner & Ventura (2014), but the exact number of steps taken by the leftmost strategy has not been measured via multi types before – therefore, this is a new result, as we now explain.

The study of the head and the leftmost strategies, at first sight, seems to be a minor reformulation of de Carvalho's results about measuring via multi types the length of executions of the Krivine abstract machine (shortened KAM) – implementing weak head evaluation –

and of the iterated KAM – that implements leftmost evaluation (de Carvalho, 2018). The study of cost models is here enlightening: de Carvalho's iterated KAM does implement leftmost evaluation, but the overhead of the machine (that is counted by de Carvalho's measure) is exponential in the number of β-steps, while here we only measure the number of β-steps, thus providing a much more parsimonious – and yet reasonable – measure. The machine overhead is actually reflected by the size of the normal form, that can be exponential, but it is measured separately.

Another work that is closely related to ours is de Carvalho *et al*. (2011), where the relational model of linear logic is used to measure evaluation lengths in proof nets. They do not however split the bounds, that is, they do not have a way to measure separately the number of steps and the size of the normal form. Moreover, their notion of cut-elimination by levels does not correspond to leftmost evaluation.

Shrinking. The study of leftmost evaluation via tight typings is then compared with traditional multi types for leftmost evaluation without tight constants. Traditional type systems characterise leftmost termination using a *shrinking* constraint: no negative occurrences of the empty multi-set in the final judgement of the type derivation (Krivine, 1993; Kesner & Ventura, 2014; Bucciarelli *et al*., 2017; de Carvalho, 2018). The comparison is instructive. First, it shows that tightness and being shrinking are predicates formulated following similar principles. Second, it allows us to provide a new proof technique for the shrinking case, by adapting the one for the tight case. Third, we provide a detailed study producing exact bounds starting from traditional derivations, revisiting the study in de Carvalho (2018).

Maximal evaluation. We also apply the technique to the *maximal strategy*, which takes the maximum number of steps to normal form, if any, and diverges otherwise. The maximal strategy has been bounded in Bernadet & Lengrand (2011), and exactly measured in Bernadet & Graham-Lengrand (2013a) via the idea of tightening, as described above. The differences with respect to Bernadet & Graham-Lengrand (2013a) are the following:

1. *Uniformity with other strategies*. The typing system in Bernadet & Graham-Lengrand (2013a) uses a form of sub-typing to deal with erasing λ-abstractions. Here, we align the type grammar with the one used for all the other evaluation strategies, which in turn allows the typing rules for λ-abstractions to be the same as for head and leftmost evaluation. This makes the whole approach more uniform across the different strategies that we treat in the paper. Moreover, our completeness theorem for the maximal strategy bears quantitative information (about evaluation lengths and size of normal forms), in contrast with Bernadet & Graham-Lengrand (2013a).
2. *Quantitative aspects of normal forms*. While Bernadet and Graham-Lengrand encode the shape of normal forms into base types, we only use two (tight) constant base types. We measure typing derivations with two indices: the first one matches the maximal evaluation *length* of the typed term, and the second one matches the *size* of its normal form together with the size of all terms that are erased by the evaluation process.

3. *Neutral terms*: we emphasise the key role of neutral terms in the technical development by describing their specificities with respect to typing. This is not explicitly broached in Bernadet & Graham-Lengrand (2013a).

Linear head evaluation. Last, we apply the tightening technique to *linear* head evaluation (Mascari & Pedicini, 1994; Danos & Regnier, 2004) (*lhd* for short), formulated in the linear substitution calculus (LSC), a λ-calculus with explicit substitutions introduced by Accattoli & Kesner (Accattoli, 2012; Accattoli *et al*., 2014) that is strongly related to linear logic proof nets (Accattoli, 2018b), and can also be seen as a minor variation over a calculus by Milner (2007). The literature contains a characterisation of *lhd*-normalisable terms (Kesner & Ventura, 2014). Moreover, de Carvalho (2018) measures the executions of the KAM, a result that can also be interpreted as a measure of *lhd*-evaluation. What we show however is stronger, and somewhat unexpected.

To bound *lhd*-evaluation, in fact, we can strongly stand on the bounds obtained for head evaluation. More precisely, the result for the exact bounds for *head* evaluation takes only into account the number of abstraction and application typing rules. For *linear* head evaluation, instead, we simply need to count also the axioms, i.e. the rules typing variable occurrences, nothing else. It turns out that the length of a linear head evaluation plus the size of the linear head normal form is *exactly* the size of the tight typing.

Said differently, multi typings simply encode evaluations in the LSC. In particular, we do not have to adapt multi types to the LSC, as for instance de Carvalho does to deal with the KAM. It actually is the other way around. As they are, multi typings naturally measure evaluations in the LSC. To measure evaluations in the λ-calculus, instead, one has to forget the role of the axioms. The best way to stress it, probably, is that the LSC is the computing device behind multi types.

Journal versus conference version. This paper is the journal version of Accattoli *et al*. (2018). In the conference paper, the head and leftmost cases were presented at the same time, while here we present them sequentially: first the simple head case, to introduce the main concepts in an easy setting, and then the leftmost case, stressing its subtleties. Moreover, the study of shrinking derivations has been considerably extended, adding in particular exact bounds via the study of unitary shrinking derivations.

In this paper, we include the key cases of the proofs of the important properties, which were not included in the conference version proofs. We also revisited the whole technical development, correcting a number of minor bugs in some statements and proofs. Some key notions have slightly changed, without any technical impact but only for a presentation purposes: the index counting the number of β-steps (in the conference version the index was twice the number, while we now make them equal, as suggested by Pierre Vial), for instance, or the definition of size of a derivation (before it was parametric, now it is unique for all the systems).

Proofs. For the sake of readability, the details of many proofs are in the Appendices. □

1.2 Other related works

Apart from the papers already cited, let us mention some other related works. A recent, general categorical framework to define intersection- and multi-type systems appears in Mazza *et al*. (2018).

While the inhabitation problem is undecidable for idempotent intersection types (Urzyczyn, 1999), the quantitative aspects provided by multi types make it decidable (Bucciarelli *et al*., 2014). Intersection types are also used in Dudenhefner & Rehof (2017) to give a bounded dimensional description of λ-terms via a notion of *norm*, which is resource-aware and orthogonal to that of *rank*. It is proved that inhabitation in bounded dimension is decidable (EXPSPACE-complete) and subsumes decidability in rank 2 (Urzyczyn, 2009).

The quantitative approach yielding upper bounds for evaluation lengths has also been extended to classical logic (Kesner & Vial, 2019), which does not only capture pure functional programming, but also control operators.

Bounds for evaluation lengths are also studied by Dal Lago & Gaboardi (2011) and Dal Lago & Petit (2013, 2014) using linear dependent types rather than intersection types.

Other works propose a more practical perspective on resource-aware analyses for functional programs. In particular, type-based techniques for automatically inferring bounds on higher-order functions have been developed, based on sized types (Hughes *et al*., 1996; Portillo *et al*., 2002; Vasconcelos & Hammond, 2004; Avanzini & Dal Lago, 2017) or amortised analysis (Hofmann & Jost, 2003; Hoffmann & Hofmann, 2010; Jost *et al*., 2017). This led to practical cost analysis tools like *Resource-Aware ML* (Hoffmann *et al*., 2012) (see raml.co). Intersection types have been used (Simões *et al*., 2007) to address the *size aliasing* problem of sized types, whereby cost analysis sometimes over-approximates cost to the point of losing all cost information (Portillo *et al*., 2002). How our multi types could further refine the integration of intersection types with sized types is a direction for future work.

Finally, in between the publication of the conference paper and the submission of this journal extension, the main ideas of our work were adapted to obtain exact bounds for open call-by-value (Accattoli & Guerrieri, 2018), call-by-need (Accattoli *et al*., 2019), and pattern-matching calculi (Alves *et al*., 2019).

2 A bird's-eye view

Our study is based on a schema that is repeated for different evaluation strategies, making most notions parametric in the strategy $\to_S$ under study. The following concepts constitute the main ingredients of our technique:

1. *Strategy, together with the normal, neutral, and abs predicates*. There is a deterministic evaluation strategy $\to_S$ whose normal forms are characterised via two related predicates, $\mathtt{normal}_S(t)$ and $\mathtt{neutral}_S(t)$, the intended meaning of the second one is that t is S-normal and can never behave as an abstraction (that is, it does not create a redex when applied to an argument). We further parametrise also this last notion by using a predicate $\mathtt{abs}_S(t)$ identifying abstractions, because

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

the definition of deterministic strategies requires some subterms to not be abstractions.[3]

2. *Typing derivations*. Derivations, denoted by Φ, are trees constructed by means of different typing rules. The following features deserve to be highlighted:

   - *Tight constants*. There are two new type constants, `neutral` and `abs`, and new rules introducing them. As their name suggests, the constants `neutral` and `abs` are used to type terms whose normal form is a neutral term or an abstraction, respectively.
   - *Tight derivations*. There is a notion of tight derivation that requires a special use of the constants.
   - *Indices*. Typing judgements have the shape $\Gamma \vdash^{(b,r)} t : A$, where b and r are indices meant to count, when the derivation is tight, the number of steps to normal form and the size of the normal form, respectively.

3. *Sizes of normal forms*. The notion of normal forms depends on the strategy, and so do their notions of size, noted $|t|_S$. A fact that may seem counter-intuitive is that these sizes do not count all the constructors in a term, but only some of them – often variables are ignored – and only those appearing in some specific positions, typically the head sizes do not count the size of arguments but only their presence. The reasons are explained precisely where the sizes are defined. Different type system inspect different aspects of (different notions of) normal forms and thus account for different quantitative aspects. The basic idea is that our notions of size measure the cost of checking that a term is normal, with respect to the given strategy.

   Additionally, there is a notion of size of typing derivations $|\Phi|$ that gives an upper bound to the sum of the indices associated to the last judgement of Φ.
4. *Characterisation*. We prove that $\Gamma \vdash^{(b,r)} t : A$ is a tight typing relatively to $\rightarrow_S$ if and only if there exists an S normal term p such that $t \rightarrow_S^b p$ and $|p|_S = r$.
5. *Proof technique*. The characterisation is obtained always through the same sequence of intermediate results. Correctness follows from the fact that all tight typings of normal forms precisely measure their size, a *substitution lemma* for typing derivations and *subject reduction*. Completeness follows from the fact that every normal form admits a tight typing, an *anti-substitution lemma* for typing derivations, and *subject expansion*.
6. *Neutral terms*. We stress the relevance of neutral terms in normalisation proofs from a typing perspective. In particular, correctness theorems always rely on a lemma about them. Neutral terms are a common concept in the study of λ-calculus, playing a key role in, for instance, the reducibility candidate technique (Girard *et al*., 1989).

The proof schema is illustrated in the next sections on two standard reduction strategies, namely *head* and *leftmost(-outermost) evaluation*. It is then slightly adapted to deal with *maximal evaluation* in Section 7 and *linear head evaluation* in Section 8. A similar schema

[3] For the head, leftmost, and maximal systems, the $\mathtt{abs}_S(t)$ predicate is trivial; it simply holds when t is an abstraction. In the linear head system of Section 8, however, we use another predicate, which is why, for the sake of uniformity, we prefer to also make our approach parametric with respect to a $\mathtt{abs}_S(t)$ predicate.

is also followed in Section 5, where we study leftmost evaluation once again, this time with respect to multi types that are not necessarily tight.

Evaluation systems. Each case study treated in the paper relies on the same properties of the strategy $\to_S$ and the related predicates $\mathtt{normal}_S(t)$, $\mathtt{neutral}_S(t)$, and $\mathtt{abs}_S(t)$, that we collect under the notion of *evaluation system*.

Definition 2.1 (Evaluation system)**.** *Let $\mathcal{T}_S$ be a set of terms, $\to_S$ be an evaluation strategy and $\mathtt{normal}_S$, $\mathtt{neutral}_S$, and $\mathtt{abs}_S$ be predicates on $\mathcal{T}_S$. All together they form an* evaluation system *S if for all $t, p, p_1, p_2 \in \mathcal{T}_S$:*

1. Determinism of $\to_S$ *: If $t \to_S p_1$ and $t \to_S p_2$ then $p_1 = p_2$.*
2. Characterisation of S-normal terms*: t is $\to_S$ -normal if and only if $\mathtt{normal}_S(t)$.*
3. Characterisation of S-neutral terms*: $\mathtt{neutral}_S(t)$ if and only if $\mathtt{normal}_S(t)$ and $\neg\mathtt{abs}_S(t)$.*

Given a strategy $\to_S$ we use $\to_S^k$ for its kth iteration and $\to_S^*$ for its reflexive-transitive closure.

Summary. The following table clearly indicates the figures defining the key concepts needed to capture the essence of the evaluation strategies and the typing systems. Correctness and completeness theorems for each evaluation strategy with respect to the corresponding typing system are also indicated.

	Head	Leftmost	Shrinking	Maximal	Linear head
Neutral/normal terms	Figure 1	Figure 5	Figure 5	Figure 5	Figure 11
Size of terms	Figure 3	Figure 7	Figure 7	Figure 7	Figure 13
Evaluation rules	Figure 2	Figure 6	Figure 6	Figure 9	Figure 12
Typing rules	Figure 4	Figure 8	Figure 8	Figure 10	Figure 14
Correctness	Theorem 3.7	Theorem 4.7	Theorem 5.10	Theorem 7.8	Theorem 8.7
Completeness	Theorem 3.11	Theorem 4.11	Theorem 5.13	Theorem 7.12	Theorem 8.11

3 Head evaluation

In this section, we consider the head evaluation system, which is the simplest one, and gradually introduce the main concepts for multi types and for the tight technique.

The set of λ-terms Λ is given by ordinary λ-terms:

$$\lambda\text{-TERMS} \qquad t, p ::= x \mid \lambda x.t \mid tp$$

Normal, neutral, and abs predicates. The predicate $\mathtt{normal}_{hd}$ defining head normal terms is in Figure 1, and it is based on an auxiliary predicate $\mathtt{neutral}_{hd}$ defining neutral terms, that are simply terms of the form $xt_1 \ldots t_k$ with $k \geq 0$. The predicate $\mathtt{abs}_{hd}(t)$ is true simply when t is an abstraction.

Small-step semantics. The *head* strategies $\to_{hd}$ is defined in Figure 2.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$$\frac{}{\texttt{neutral}_{hd}(x)} \qquad \frac{\texttt{neutral}_{hd}(t)}{\texttt{neutral}_{hd}(tp)} \qquad \frac{\texttt{neutral}_{hd}(t)}{\texttt{normal}_{hd}(t)} \qquad \frac{\texttt{normal}_{hd}(t)}{\texttt{normal}_{hd}(\lambda y.t)}$$

Fig. 1. Head neutral and head normal terms.

$$\frac{}{(\lambda x.u)q \rightarrow_{hd} u\{x{\leftarrow}q\}} \qquad \frac{t \rightarrow_{hd} p}{\lambda x.t \rightarrow_{hd} \lambda x.p} \qquad \frac{\neg\texttt{abs}_{hd}(t) \quad t \rightarrow_{hd} p}{tu \rightarrow_{hd} pu}$$

Fig. 2. Head strategy.

$$|x|_{hd} := 0 \qquad |\lambda x.p|_{hd} := |p|_{hd} + 1 \qquad |pu|_{hd} := |p|_{hd} + 1$$

Fig. 3. Head size of terms.

Proposition 3.1 (Head evaluation systems). *(Λ, $\rightarrow_S$, $\texttt{neutral}_{hd}$, $\texttt{normal}_{hd}$, $\texttt{abs}_{hd}$) is an evaluation system.*

The proof is routine, and it is then omitted also from the Appendix.

Size of normal forms. The notions of *head size* $|t|_{hd}$ of a (head normal) term t is defined in Figure 3 (for simplicity, we define it over the structure of terms and not of head normal forms).

There are two unusual points:

1. *Arguments.* In the application case, the argument u does not contribute (1 accounts for the application constructor itself, that is, just the existence of an argument). Head evaluation does not enter arguments and so it is natural to not account for them. Another point of view is considering $|t|_{hd}$ as measuring the part of t that an algorithm has to explore in order to check that it is head normal.

   Note that head evaluation does suffer of size explosion, but only if the size of arguments is also taken into account – see Grabmayer (2018) for details.
2. *Variables*. Variables may be counted for the size of normal forms, but we do not count them for uniformity. In general, the first counter on typing judgements shall measure the dynamic aspect (the number of steps) of the computation, while the second counter is devoted to the static aspect (the size of normal forms). Counting variables for normal forms correspond to counting axioms in the typing system, which in turn accounts for the number of *single* (linear) variable replacements done by the strategy – this shall be done in Section 8, where we deal with *linear head evaluation*. But head evaluation is based on meta-level (non-linear) substitution and thus does not account for single variable replacements – thus variables must not be counted at the dynamic level. To be uniform, we do not count variables for the static aspect either, thus excluding them from the size of normal forms. Note that the same point applies for the leftmost and maximal strategies of the next sections, whose size of normal forms shall not count variables either.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$$\frac{}{x:[A]\vdash^{(0,0)} x:A}\ \mathsf{ax} \qquad \frac{(\Delta_i \vdash^{(b_i,r_i)} t:A_i)_{i\in I}}{\uplus_{i\in I}\Delta_i \vdash^{(+_{i\in I} b_i,+_{i\in I} r_i)} t:[A_i]_{i\in I}}\ \mathsf{many}$$

$$\frac{\Gamma \vdash^{(b,r)} t:A}{\Gamma \setminus\!\!\setminus x \vdash^{(b+1,r)} \lambda x.t:\Gamma(x)\to A}\ \mathsf{fun}_b \qquad \frac{\mathtt{tight}(\Gamma(x)) \quad \Gamma \vdash^{(b,r)} t:\mathtt{tight}}{\Gamma \setminus\!\!\setminus x \vdash^{(b,r+1)} \lambda x.t:\mathtt{abs}}\ \mathsf{fun}_r$$

$$\frac{\Gamma \vdash^{(b,r)} t:M\to A \quad \Delta \vdash^{(b',r')} p:M}{\Gamma \uplus \Delta \vdash^{(b+b',r+r')} tp:A}\ \mathsf{app}_b \qquad \frac{\Gamma \vdash^{(b,r)} t:\mathtt{neutral}}{\Gamma \vdash^{(b,r+1)} tp:\mathtt{neutral}}\ \mathsf{app}_r^{hd}$$

Fig. 4. Type system for head evaluation.

Multi types. We define the following notions about types:

- *Multi types* are defined by the following grammar:

TIGHT CONSTANTS	$\mathtt{tight} ::= \mathtt{neutral} \mid \mathtt{abs}$
TYPES	$A,B ::= \mathtt{tight} \mid X \mid M \to A$
MULTI-SETS	$M ::= [A_i]_{i\in I}$ (I a finite set)

  where X ranges over a non-empty set of *atomic types* and $[\ldots]$ denotes the multi-set constructor.
- *Multi-sets*. We use $[\,]$ to denote the empty multi-set, $\uplus$ for multi-set union, $\sqsubseteq$ for multi-set inclusion, and $\setminus$ for multi-set difference. An example of multi-set is $M = [A,A,B]$, which contains two occurrences of A and one occurrence of B. Then for example $M \uplus [A] = [A,A,A,B]$ and $M \setminus [A] = [A,B]$.
- A *typing context* Γ is a map from variables to finite multi-sets M of types such that only finitely many variables are not mapped to the empty multi-set $[\,]$. The empty typing context is written ϵ. We write $\mathtt{dom}(\Gamma)$ for the domain of Γ, i.e. the set $\{x \mid \Gamma(x) \neq [\,]\}$.
- *Tightness*. We use the notation $\mathtt{Tight}$ for a multi-set containing only tight constants. Moreover, we write $\mathtt{tight}(A)$ if A is of the form $\mathtt{tight}$, $\mathtt{tight}(M)$ if M is of the form $\mathtt{Tight}$, and $\mathtt{tight}(\Gamma)$ if $\mathtt{tight}(\Gamma(x))$ for all x, in which case we also say that Γ is *tight*.
- The *multi-set union* $\uplus$ is extended to typing contexts point-wise, i.e. $\Gamma \uplus \Delta$ maps each variable x to $\Gamma(x) \uplus \Delta(x)$. This notion is extended to several contexts as expected so that $\uplus_{i\in I}\Gamma_i$ denotes a finite union of contexts (when $I=\emptyset$ the notation is to be understood as the empty context). When $\mathtt{dom}(\Gamma)\cap\mathtt{dom}(\Delta)=\emptyset$, then $\Gamma \uplus \Gamma'$ may also be written $\Gamma;\Gamma'$. We write $x:M$ for the typing context assigning M to x and $[\,]$ to all the other variables. Then the notation $\Gamma; x:M$ combines the two previous ones.
- The *restricted* context Γ with respect to the variable x, written $\Gamma \setminus\!\!\setminus x$ is defined by $(\Gamma \setminus\!\!\setminus x)(x) := [\,]$ and $(\Gamma \setminus\!\!\setminus x)(y) := \Gamma(y)$ if $y \neq x$.

Head typing system. The typing rules of the head system *hd* are presented in Figure 4. Roughly, the intuitions behind the typing rules are (please ignore the indices b and r for the time being):

- *Rules* ax, fun_b, *and* app_b. These rules are essentially the traditional rules for multi types for head and LO evaluation (see, e.g., Bucciarelli *et al*., 2017), modulo the presence of the indices.
- *Rule* many. This is a structural rule allowing typing terms with a multi-set of types. In some presentations of multi types many is hardcoded in the right premiss of the app_b rule (that requires a multi-set). For technical reasons, it is preferable to separate it from app_b. Morally, it corresponds to the !-promotion rule in linear logic.
- *Rule* fun_r. t has already been tightly typed, and all the types associated to x are also tight constants, i.e. the typing context contains a pair of the form $x : \mathtt{Tight}$. Then $\lambda x.t$ receives the tight constant $\mathtt{abs}$ for abstractions. The consequence is that this abstraction can no longer be applied, because there are no rules to apply terms of type $\mathtt{abs}$. Therefore, the abstraction constructor cannot be consumed by evaluation and it ends up in the (head) normal form of the term, that has the form $\lambda x.t'$, where t' is the (head) normal form of t.
- *Rule* app_r^{hd}. t has already been tightly typed with $\mathtt{neutral}$ and so morally it head normalises to a term t' having neutral form $xu_1 \ldots u_k$. The rule adds a further argument p that cannot be consumed by evaluation, because t shall never become an abstraction. Therefore, p ends up in the head normal form $t'p$ of tp, that is still neutral – correctly, so that tp is also typed with $\mathtt{neutral}$. Note that there is no need to type p because head evaluation never enters into arguments.
- *Tight constants and predicates*. There is of course a correlation between the tight constants $\mathtt{neutral}$ and $\mathtt{abs}$ and the predicates $\mathtt{neutral}_{hd}$ and $\mathtt{abs}_{hd}$. Namely, a term t is hd-typable with $\mathtt{neutral}$ if and only if the hd-normal form of t verifies the predicate $\mathtt{neutral}_{hd}$, as we shall prove. For the tight constant $\mathtt{abs}$ and the predicate $\mathtt{abs}_{hd}$ the situation is similar but weaker: if the hd-normal form of t verifies $\mathtt{abs}_{hd}$ then t is typable with $\mathtt{abs}$, but not the other way around – variables are typable with $\mathtt{abs}$ without being abstractions.
- *Presentation of abstraction rules*. The presentation of the abstraction rules may seem unusual. We explain the relationship with the usual presentation in the *vacuous abstractions* paragraph given next.
- *The type systems is not syntax-directed*, e.g. given an abstraction (resp. an application), it can be typed with rule fun_r or fun_b (resp. app_r^{hd} or app_b), depending on whether the constructor typed by the rule ends up in the normal form or not. Thus, for example, given the term $\mathtt{II}$, where $\mathtt{I}$ is the identity function $\lambda z.z$, the second occurrence of $\mathtt{I}$ can be typed with $\mathtt{abs}$ using rule fun_r, while the first one can be typed with $[\mathtt{abs}] \to \mathtt{abs}$ using rule fun_b.

We write $\Phi \rhd_{hd} \Gamma \vdash^{(b,r)} t : A$, to stress that Φ is a *head* typing derivation ending in the judgement $\Gamma \vdash^{(b,r)} t : A$ when, in the next sections, we discuss also other type systems.

Indices. The roles of b and r can be described as follows:

- *b and β-steps*. b counts the abstraction rules of the derivation that may be used to form (head) β-redexes, i.e. the number of fun_b rules, because it is the only rule introducing an arrow type. The index b is at least the number of β-steps to normal form because typing a β-redex requires a fun_b rule. It may be greater than such a

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

number if some of the abstractions typed with fun_b are never applied, and therefore end up in the normal form. For tight typing derivations (introduced next), we are going to prove that b is *exactly* the length of the head evaluation of the typed term to normal form. Essentially, tightness shall force abstractions typed with fun_b to be applied.

- *r and size of the result.* r counts the rules typing constructors that cannot be consumed by β-reduction according to the head evaluation strategy, and that therefore shall end up in contributing to the normal form. It counts the number of fun_r and app_r^{hd}. These rules type the result of the evaluation, according to the head strategy, and measure the size of the result. Note that the type $\mathtt{abs}$ given by rule fun_r is not an arrow type and cannot therefore be composed. Essentially, tightness shall force all abstractions ending in the normal form to be typed with fun_r.

For system *hd*, the indices on typing judgements are not really needed, as b can be recovered as the number of fun_b rules, and r as the number of fun_r and app_r^{hd} rules. We prefer to make them explicit because (1) we want to stress the separate counting, and (2) for linear head evaluation in Section 8 the counting shall be more involved, and the indices shall not be recoverable.

Note that only some rules contribute to the indices b and r. The fact that ax is not counted shall change in Section 8, where we show that counting ax rules corresponds to measure evaluations in the linear substitution calculus. The fact that many is not counted, instead, is due to the fact that it does not correspond to any constructor on terms. A further reason is that the rule may be eliminated by absorbing it in the app_b rule, that is the only rule that uses multi-sets – it is however technically convenient to separate the two. The fact that app_b is not counted is because we already count fun_b for β-redexes, and counting app_b would provide a number twice the measure we are interested in.[4]

Typing size. We define the size $|\Phi|$ of a typing derivation Φ as the number of rules in Φ, not counting the occurrences of rule many. The size of a derivation gives an upper bound to the sum of the indices (b,r) on its final judgement: whenever $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} t : A$, we have $b + r \leq |\Phi|$.

Subtleties and easy facts. Let us overview some peculiarities and consequences of the definition of our type systems.

1. *Relevance*. No weakening is allowed in axioms. An easy induction on typing derivations shows that a variable declaration $x : M \neq [\,]$ appears explicitly in the typing context Γ of a type derivation for t only if x occurs free and *typed* in t. In system *hd*, arguments of applications might not be typed (because of rule app_r^{hd}), and so there may be $x \in \mathtt{fv}(t)$ but not appearing in Γ.
2. *Vacuous abstractions*. Our presentation of abstraction rules in Figure 1 precisely accounts for the case in which the abstraction binds a variable x not appearing in the type context Γ. Indeed, in the fun_b rule, if $x \notin \mathtt{dom}(\Gamma)$, then $\Gamma \setminus\!\!\setminus x$ is equal to

[4] In the conference version of this paper, we actually counted app_b rules and then obtained that b counted twice the number of β-steps. We believe that it is cleaner not to count app_b rules, as suggested by P. Vial.

Γ and $\Gamma(x)$ is $[\,]$, while in the fun_r rule, if $x \notin \mathtt{dom}(\Gamma)$, then $\Gamma(x)$ is $[\,]$ and thus $\mathtt{tight}([\,])$ holds. The alternative and equivalent presentation of these rules is

$$\frac{\Gamma; x : M \vdash^{(b,r)} t : A}{\Gamma \vdash^{(b+1,r)} \lambda x.t : M \to A}\ \mathsf{fun}_b \qquad \frac{\Gamma; x : \mathtt{Tight} \vdash^{(b,r)} t : \mathtt{tight}}{\Gamma \vdash^{(b,r+1)} \lambda x.t : \mathtt{abs}}\ \mathsf{fun}_r$$

In the proofs, we rather use this alternative formulation (that requires that x does appear in the type context, and implicitly assumes that $\Gamma; x : [\,] = \Gamma$) but we prefer to adopt the precise presentation in the formal definition, to avoid ambiguities.

3. *Head typings and applications*: note the app_r^{hd} rule types an application tp without typing the right subterm p. This matches the fact that tp is a head normal form when t is, independently of the status of p.

Tight derivations. A given term t may have many different typing derivations, indexed by different pairs (b, r). They always provide upper bounds on head evaluation lengths and lower bounds on the size of head normal forms. The interesting aspect of our type systems, however, is that there is a simple description of a class of typing derivations that provide *exact* bounds for these quantities, as we shall show. Their definition relies on tight constants.

Definition 3.2 (Tight head derivations)**.**
A derivation $\Phi \rhd_{hd}\ \Gamma \vdash^{(b,r)} t : B$ *is* tight *if* $\mathtt{tight}(B)$ *and* $\mathtt{tight}(\Gamma)$.

Remarkably, tightness is expressed as a property of the last judgement only. This is however not unusual: characterisations of weakly normalising terms via intersection/multi types also rely on properties of the last judgement only, as discussed in Section 5.

In Section 5, in particular, we show the size of a tight derivation for a normal term t is *minimal* among derivations for t. Moreover, it is also of the same size of the minimal derivations making no use of tight constants nor rules using them. Therefore, tight derivations may be thought as a characterisation of minimal derivations for normal terms – for non-normal terms the question is subtle and it is discussed at the end of Section 5.

Let us also refine the intuitions about tightness of the paragraph *indices* above, where we explain in particular how tightness forces the partitioning of abstractions. Dually, tightness also forces a partitioning of application rules. At some point of the evaluation, the left subterm of every application typed with app_b shall turn into an abstraction, forming a redex whose firing shall *consume* the app_b rule. Therefore, all applications in the normal form are typed with app_r^{hd}.

Example. Let $t_0 = (\lambda x_1.(\lambda x_0.x_0x_1)x_1)\mathtt{I}$, where $\mathtt{I}$ is the identity function $\lambda z.z$. The head evaluation of t_0 to hd normal-form is

$$(\lambda x_1.(\lambda x_0.x_0x_1)x_1)\mathtt{I} \to_{hd} (\lambda x_0.x_0\mathtt{I})\mathtt{I} \to_{hd} \mathtt{I}\mathtt{I} \to_{hd} \mathtt{I}$$

The evaluation sequence has length 3. The head normal form has size 1. To give a tight typing for the term t_0 let us write $\mathtt{abs}^{\mathtt{abs}}$ for $[\mathtt{abs}] \to \mathtt{abs}$. Then,

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$$
\begin{prooftree}
\AxiomC{$x_0 : [\mathtt{abs}^{\mathtt{abs}}] \vdash^{(0,0)} x_0 : \mathtt{abs}^{\mathtt{abs}}$}
\AxiomC{$x_1 : [\mathtt{abs}] \vdash^{(0,0)} x_1 : \mathtt{abs}$}
\UnaryInfC{$x_1 : [\mathtt{abs}] \vdash^{(0,0)} x_1 : [\mathtt{abs}]$}
\BinaryInfC{$x_0 : [\mathtt{abs}^{\mathtt{abs}}], x_1 : [\mathtt{abs}] \vdash^{(0,0)} x_0 x_1 : \mathtt{abs}$}
\UnaryInfC{$x_1 : [\mathtt{abs}] \vdash^{(1,0)} \lambda x_0.x_0 x_1 : [\mathtt{abs}^{\mathtt{abs}}] \rightarrow \mathtt{abs}$}
\AxiomC{$x_1 : [\mathtt{abs}^{\mathtt{abs}}] \vdash^{(0,0)} x_1 : \mathtt{abs}^{\mathtt{abs}}$}
\UnaryInfC{$x_1 : [\mathtt{abs}^{\mathtt{abs}}] \vdash^{(0,0)} x_1 : [\mathtt{abs}^{\mathtt{abs}}]$}
\BinaryInfC{$x_1 : [\mathtt{abs}, \mathtt{abs}^{\mathtt{abs}}] \vdash^{(1,0)} (\lambda x_0.x_0 x_1) x_1 : \mathtt{abs}$}
\UnaryInfC{$\vdash^{(2,0)} \lambda x_1.(\lambda x_0.x_0 x_1) x_1 : [\mathtt{abs}, \mathtt{abs}^{\mathtt{abs}}] \rightarrow \mathtt{abs}$}
\AxiomC{$\vdots$}
\UnaryInfC{$\vdash^{(1,1)} \mathtt{I} : [\mathtt{abs}, \mathtt{abs}^{\mathtt{abs}}]$}
\BinaryInfC{$\vdash^{(3,1)} (\lambda x_1.(\lambda x_0.x_0 x_1) x_1)\mathtt{I} : \mathtt{abs}$}
\end{prooftree}
$$

Indeed, the pair $(3, 1)$ represents three evaluation steps to *hd* normal-form and a head normal form of size 1.

3.1 Tight head correctness

Correctness of tight typings for head evaluation is the fact that whenever a term is *tightly* typable with indices (b, r), then b is exactly the number of head evaluation steps to head normal form while r is exactly the head size of the head normal form. The correctness theorem is always obtained via three intermediate steps.

First step: tight typings of normal forms. The first step is to show that, when a tightly typed term is a head normal form, then the first index b of its type derivation is 0, so that it correctly captures the number of steps, and the second index r coincides exactly with its head size. An interesting auxiliary lemma relating *hd*-neutral terms and tight typings is required.

Lemma 3.3 (Tight spreading on neutral terms). *Let t be such that $\mathtt{neutral}_{hd}(t)$ and $\Phi \rhd_{hd} \Gamma \vdash^{(b,r)} t : A$ be a typing derivation such that $\mathtt{tight}(\Gamma)$. Then $\mathtt{tight}(A)$ and the last rule of Φ is not app_b.*

Proof By induction on $\mathtt{neutral}_{hd}(t)$. Cases:

- *Variable*, i.e. $t = x$. Then $\Gamma = x : [A]$, and A is tight because Γ is tight by hypothesis.
- *Application*, i.e. $t = pu$ and $\mathtt{neutral}_{hd}(t)$ because $\mathtt{neutral}_{hd}(p)$. The last rule of Φ can only be app_b or app_r^{hd}. In both cases the left subterm p is typed by a sub-derivation $\Phi' \rhd \Gamma_p \vdash^{(b',r')} p : B$ such that all types in Γ_p appear in Γ, and so they are all tight by hypothesis. Since $\mathtt{neutral}_{hd}(p)$, we can apply the *i.h.* and obtain that B is tight. The only possible case is then $B = \mathtt{neutral}$ and the last rule of Φ is then app_r^{hd}. Then $A = B = \mathtt{neutral}$. □

The lemma expresses the fact that tightness of neutral terms only depends on their contexts. Morally, this fact is what makes tightness expressible as a property of the final judgement only. We shall see in Section 5 that a similar property is hidden in more traditional approaches to weak normalisation (see Lemma 5.6). Such a spreading property

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

appears repeatedly in our study, and we believe that its isolation is one of the contributions of our work, induced by the modular and comparative study of various strategies.

Proposition 3.4 (Properties of *hd* typings for normal forms)**.** *Let t be such that* $\mathtt{normal}_{hd}(t)$*, and* $\Phi \rhd_{hd} \ \Gamma \vdash^{(b,r)} t : A$ *be a typing derivation.*

1. Size bound*:* $|t|_{hd} \leq |\Phi|$.
2. Tight indices*: If Φ is tight then $b = 0$ and $r = |t|_{hd}$.*
3. Neutrality*: If $A = \mathtt{neutral}$ then* $\mathtt{neutral}_{hd}(t)$.

Proof By induction on t. Note that $\mathtt{neutral}_{hd}$ implies $\mathtt{normal}_{hd}$ and so we can apply the *i.h.* when $\mathtt{neutral}_{hd}$ holds on some subterm of t. The proof is mostly straightforward, there is only one interesting case, the one using the tight spreading on neutral terms (Lemma 3.3). The case is when $\mathtt{normal}_{hd}(t)$ because $\mathtt{neutral}_{hd}(t)$ and $t = pu$, that in turn implies $\mathtt{neutral}_{hd}(p)$. If the last rule of Φ is $\mathtt{app}_b$ then Φ has the form

$$\frac{\Phi_p \rhd_{hd} \ \Gamma_p \vdash^{(b_p,\, r_p)} p : M \to A \qquad \Phi_u \rhd_{hd} \ \Gamma_u \vdash^{(b_u,\, r_u)} u : M}{\Gamma_p \uplus \Gamma_u \vdash^{(b_p+b_u, r_p+r_u)} pu : A}\ \mathtt{app}_b$$

with $b = b_p + b_u$, $r = r_p + r_u$, and $\Gamma = \Gamma_p \uplus \Gamma_u$.

1. *Size bound*. By *i.h.*, $|p|_{hd} \leq |\Phi_p|$, from which it follows $|t|_{hd} = |p|_{hd} + 1 \leq_{i.h.} |\Phi_p| + 1 \leq |\Phi|$.
2. *Tight size bound*: we show that the pre-condition for this case is impossible. If Φ is tight then $\Gamma = \Gamma_p \uplus \Gamma_u$ is a tight typing context, and so is Γ_p. Since $\mathtt{neutral}_{hd}(t)$, the tight spreading on neutral terms (Lemma 3.3) implies that the type of p in Φ_p has to be tight, while it is $M \to A$ – absurd.
3. *Neutrality*. $\mathtt{neutral}_{hd}(t)$ holds by hypothesis.

If the last rule of Φ is $\mathtt{app}_r^{hd}$ the statement follows easily from the *i.h.* □

Note that Proposition 3.4.2 indirectly shows that all tight derivations of a same term carry the same indices, and essentially have the same size (it can be easily shown that they all have the same number of axioms). The only way in which two tight derivations can differ, in fact, is whether the variables in the type context are typed with `neutral` or `abs`, but the structure of different derivations is necessarily the same, which is also the structure of the head normal form itself.

Second step: substitution lemma. Then one has to show that types, typings, and indices behave well with respect to substitution, which is essential, given that β-reduction is based on it.

Lemma 3.5 (Substitution and typings for *hd*)**.** *Let* $\Phi_t \rhd_{hd} \ \Delta; x : M \vdash^{(b,r)} t : A$ *and* $\Phi_p \rhd_{hd} \ \Gamma \vdash^{(b',r')} p : M$*. Then there exists a derivation* $\Phi_{t\{x\leftarrow p\}} \rhd_{hd} \ \Gamma \uplus \Delta \vdash^{(b+b',r+r')} t\{x\leftarrow p\} : A$*, where* $|\Phi_{t\{x\leftarrow p\}}| = |\Phi_t| + |\Phi_p| - |M|$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Note that the lemma also holds for $M = [\,]$, in which case Γ is necessarily empty. As already pointed out, in system *hd* it can be that $M = [\,]$ and yet $x \in \mathtt{fv}(t)$ and $t\{x{\leftarrow}p\} \neq t$.

Proof Easy induction on the derivation of $\Delta; x : M \vdash^{(b,r)} t : A$, see Appendix A.1. □

Third step: quantitative subject reduction. Finally, one needs to shows a quantitative form of type preservation along evaluation. When the typing is tight, every evaluation step decreases the first index b by exactly 1, accounting for the abstraction constructor *consumed* by the firing of the redex.

Proposition 3.6 (Quantitative subject reduction for *hd*)**.** *If $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} t : A$ and $t \to_{hd} p$ then $b \geq 1$ and there exists a typing Φ' such that $\Phi' \triangleright_{hd} \Gamma \vdash^{(b-1,r)} p : A$ and $|\Phi| > |\Phi'|$.*

Proof By induction on $t \to_{hd} p$. The only case not following immediately from the *i.h.* is the one of reduction at the root of the term, that is when $t = (\lambda x.u)q \to_{hd} u\{x{\leftarrow}q\} = p$. Assume $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} (\lambda x.u)q : A$. The derivation Φ must end with rule $\mathtt{app}_b$, and the derivation of its premiss for $(\lambda x.u)$ must end with $\mathtt{fun}_b$. Hence, there are two derivations $\Phi_u \triangleright_{hd} \Gamma_u; x : M \vdash^{(b_u,r_u)} u : A$ and $\Phi_q \triangleright_{hd} \Gamma_q \vdash^{(b_q,r_q)} q : M$, with $(b,r) = (b_u + b_q + 1, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_q$. Applying the substitution lemma (Lemma 3.5), we obtain $\Phi' \triangleright_{hd} \Gamma \vdash^{(b_u+b_q,r_u+r_q)} u\{x{\leftarrow}q\} : A$ such that $|\Phi'| = |\Phi_u| + |\Phi_q| - |M| < |\Phi_u| + |\Phi_q| + 2 = |\Phi|$. □

Summing up. The tight correctness theorem is proved by a straightforward induction on the evaluation length relying on quantitative subject reduction (Proposition 3.6) for the inductive case, and the properties of tight typings for normal forms (Proposition 3.4) for the base case.

Theorem 3.7 (Tight correctness for *hd*)**.** *Let $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} t : A$ be a derivation. Then there exists p and k such that $t \to^k_{hd} p$, $k \leq b$, $\mathtt{normal}_{hd}(p)$, and $|p|_{hd} + k \leq |\Phi|$. Moreover, if Φ is tight, then $b = k$, $|p|_{hd} = r$ and ($A = \mathtt{neutral}$ implies $\mathtt{neutral}_{hd}(p)$).*

Proof By induction on $|\Phi|$. If t is a $\to_{hd}$ normal form – that covers the base case $|\Phi| = 1$, for which t is necessarily a variable – then by taking $p := t$ and $k := 0$ the first statement follows from Proposition 3.4.1, the tight statement follows from the *tight indices* and *neutrality* properties of tight typings of normal forms (Proposition 3.4:2–3).

Otherwise, $t \to_{hd} u$ and, by quantitative subject reduction (Proposition 3.6), there is a derivation $\Phi' \triangleright_{hd} \Gamma \vdash^{(b-1,r)} u : A$ such that $|\Phi'| < |\Phi|$. By *i.h.*, there exists p and k' such that $\mathtt{normal}_{hd}(p)$ and $u \to^{k'}_{hd} p$ and $k' \leq b-1$ and $|p|_{hd} + k' \leq |\Phi'|_{hd}$. Just note that $t \to_{hd} u \to^{k'}_{hd} p$, that is, $t \to^{k'+1}_{hd} p$, where $k'+1 \leq (b-1)+1 = b$. Therefore $|p|_{hd} + k' + 1 \leq |\Phi'| + 1 \leq |\Phi|$.

For the tight statement, we know by the *i.h.* that $k' = b-1$ so that $k := k'+1$ verifies $k = b$. The *i.h.* also gives $|p|_{hd} = r$ and ($A = \mathtt{neutral}$ implies $\mathtt{neutral}_{hd}(p)$), which concludes the proof. □

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

3.2 Tight head completeness

Completeness of tight head typings expresses the fact that every head normalising term has a tight derivation in system *hd*. As for correctness, the completeness theorem is always obtained via three intermediate steps, dual to those for correctness. Essentially, one shows that every head normal form has a tight derivation and then extends the result to head normalising terms by pulling typability back through evaluation, using a subject expansion property.

First step: normal forms are tightly typable. A simple induction on the structure of normal forms proves the following proposition.

Proposition 3.8 (Normal forms are tightly typable for *hd*)**.** *Let* t *be such that* $\mathtt{normal}_{hd}(t)$*. Then*

1. Existence*: there exists a tight derivation* $\Phi \triangleright_{hd} \Gamma \vdash^{(0,|t|_{hd})} t : A$*.*
2. Structure*: moreover, if* $\mathtt{neutral}_{hd}(t)$ *then* $A = \mathtt{neutral}$*, and if* $\mathtt{abs}_{hd}(t)$ *then* $A = \mathtt{abs}$.

In contrast to the proposition for normal forms of the correctness part (Proposition 3.4), here there are no auxiliary lemmas, so the property is simpler.

Proof Easy induction on $\mathtt{normal}_{hd}(t)$, see Appendix A.2. □

Second step: anti-substitution lemma. In order to pull typability back along evaluation sequences, we have to first show that typability can also be pulled back along substitutions.

Lemma 3.9 (Anti-substitution and typings for *hd*)**.** *Let* $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} t\{x{\leftarrow}p\} : A$*. Then there exist*

- *a multi-set* M*;*
- *a typing derivation* $\Phi_t \triangleright_{hd} \Gamma_t ; x : M \vdash^{(b_t,r_t)} t : A$*; and*
- *a typing derivation* $\Phi_p \triangleright_{hd} \Gamma_p \vdash^{(b_p,r_p)} p : M$

such that

- Typing context*:* $\Gamma = \Gamma_t \uplus \Gamma_p$*;*
- Indices*:* $(b,r) = (b_t + b_p, r_t + r_p)$*;*
- Size*:* $|\Phi| = |\Phi_t| + |\Phi_p| - |M|$.

Proof Easy induction on Φ, see Appendix A.2. □

Let us point out that the anti-substitution lemma holds also in the degenerated case in which x does not occur in t and p is not *hd*-normalising: rule many can indeed be used to type *any* term p with $\vdash^{(0,0)} p : [\,]$ by taking an empty set I of indices for the premisses. Note also that this is *forced* by the fact that $x \notin \mathtt{fv}(t)$, and so $\Gamma_t(x) = [\,]$. Finally, this fact does not contradict the correctness theorem, because here p is typed with a multi-set, while the theorem requires a type.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Third step: quantitative subject expansion. This property guarantees that typability can be pulled back along evaluation sequences.

Proposition 3.10 (Quantitative subject expansion for *hd*). *Let* $\Phi \rhd_{hd} \Gamma \vdash^{(b,r)} p : A$ *be a derivation. If* $t \to_{hd} p$ *then there exists a typing* Ψ *such that* $\Psi \rhd_{hd} \Gamma \vdash^{(b+1,r)} t : A$ *and* $|\Psi| > |\Phi|$.

Proof By induction on $t \to_{hd} p$. The only case not following immediately from the *i.h.* is the one of reduction at the root of the term, that is when $t = (\lambda x.u)q \to_{hd} u\{x \leftarrow q\} = p$. Assume $\Phi \rhd_{hd} \Gamma \vdash^{(b,r)} u\{x \leftarrow q\} : A$. By applying the anti-substitution lemma (Lemma 3.9), we obtain the premisses of the following derivation Ψ:

$$\dfrac{\dfrac{\Phi_u \rhd_{hd} \Gamma_u, x : M \vdash^{(b_u,r_u)} u : A}{\Gamma_u \vdash^{(b_u+1,r_u)} \lambda x.u : M \to A} \qquad \Phi_q \rhd_{hd} \Gamma_q \vdash^{(b_q,r_q)} q : M}{\Gamma_u \uplus \Gamma_q \vdash^{(b_u+b_q+1,r_u+r_q)} (\lambda x.u)q : A}$$

with $(b,r) = (b_u + b_q, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_q$. By the same Lemma 3.9 we have $|\Phi_u| + |\Phi_q| - |M| = |\Phi|$. Then $|\Psi| = |\Phi_u| + |\Phi_q| + 2 > |\Phi_u| + |\Phi_q| - |M| = |\Phi|$. Note that the difference between the sizes of Ψ and Φ is indeed between 2 and $2 + |M|$. □

Summing up. The tight completeness theorem is proved by a straightforward induction on the evaluation length relying on quantitative subject expansion (Proposition 3.10) for the inductive case, and the existence of tight typings for normal forms (Proposition 3.8) for the base case.

Theorem 3.11 (Tight completeness for *hd*). *Let* $t \to_{hd}^{k} p$ *with* $\mathtt{normal}_{hd}(p)$. *Then there exists a tight typing* $\Phi \rhd_{hd} \Gamma \vdash^{(k,|p|_{hd})} t : A$. *Moreover, if* $\mathtt{neutral}_{hd}(p)$ *then* $A = \mathtt{neutral}$, *and if* $\mathtt{abs}_{hd}(p)$ *then* $A = \mathtt{abs}$.

Proof By induction on $t \to_{hd}^{k} p$. If $k = 0$ the statement is given by the existence of tight typings for $\mathtt{normal}_{hd}$ terms (Proposition 3.8), that also provides the *moreover* part. Let $k > 0$ and $t \to_{hd} u \to_{hd}^{k-1} p$. By *i.h.*, there exists a tight typing derivation $\Psi \rhd \vdash_{\mathtt{tight}}^{(k-1,|p|_{hd})} u$. By subject expansion (Proposition 3.10) there exists a typing derivation Φ of u with the same types in the ending judgement of Ψ – then Φ is tight – and with indices $(k, |p|_{hd})$. □

4 Leftmost-outermost evaluation

In this section, we slightly modify the system for head evaluation to provide tight bounds for leftmost-outermost (shortened leftmost in the text and *lo* in mathematical symbols) evaluation, the iteration of head evaluation into arguments. Lo evaluation is an important strategy for two main reasons. First, it is a normalising strategy, that is, it reaches a normal form, whenever it exists. Second, its number of steps can be taken as a reasonable time cost model.

$$\frac{}{\mathtt{neutral}_{lo}(x)} \qquad \frac{\mathtt{neutral}_{lo}(t) \quad \mathtt{normal}_{lo}(p)}{\mathtt{neutral}_{lo}(tp)} \qquad \frac{\mathtt{neutral}_{lo}(t)}{\mathtt{normal}_{lo}(t)} \qquad \frac{\mathtt{normal}_{lo}(t)}{\mathtt{normal}_{lo}(\lambda y.t)}$$

Fig. 5. Leftmost-outermost neutral and normal terms.

$$\frac{}{(\lambda x.u)q \rightarrow_{lo} u\{x{\leftarrow}q\}} \qquad \frac{t \rightarrow_{lo} p}{\lambda x.t \rightarrow_{lo} \lambda x.p} \qquad \frac{\neg\mathtt{abs}_{lo}(t) \quad t \rightarrow_{lo} p}{tu \rightarrow_{lo} pu} \qquad \frac{\mathtt{neutral}_{lo}(u) \quad t \rightarrow_{lo} p}{ut \rightarrow_{lo} up}$$

Fig. 6. Leftmost-outermost strategy.

$$|x|_{lo} := 0 \qquad |\lambda x.p|_{lo} := |p|_{lo} + 1 \qquad |pu|_{lo} := |p|_{lo} + |u|_{lo} + 1$$

Fig. 7. Leftmost size of terms.

The development follows the same lines of the head case. There are however subtle and important differences. Essentially, the typing system is tweaked, as discussed next, so that arguments of typed applications are always typed, and the tight hypothesis has to be added to various properties, that would otherwise not hold. In particular, it has to be added to subject reduction (and, of course, subject expansion) whose proof then becomes subtler, because tightness has to somehow spread to sub-derivations, while it is defined as a property of the final judgement only.

Basic definitions. The predicates $\mathtt{normal}_{lo}$ and $\mathtt{neutral}_{lo}$ defining normal and neutral terms are in Figure 5 – note the case $\mathtt{neutral}_{lo}(tp)$ which is the one distinguishing the leftmost case from the head one. As in the head case, the predicates $\mathtt{abs}_{lo}(t)$ is true simply when t is an abstraction. The *leftmost-outermost* strategy $\rightarrow_{lo}$ is defined in Figure 6 – note the case $ut \rightarrow_{lo} up$.

Proposition 4.1 (Leftmost evaluation system). *$(\Lambda, \rightarrow_{lo}, \mathtt{neutral}_{lo}, \mathtt{normal}_{lo}, \mathtt{abs}_{lo})$ is an evaluation system.*

The proof is routine, and it is then omitted also from the Appendix.

Size of normal forms. The notion of leftmost(-outermost) size $|t|_{lo}$ of a term t is defined in Figure 7 – the difference with the head size is on applications.

Note that $|t|_{lo}$ counts the number of internal nodes of the syntax tree of t. Variable occurrences – that are ignored – are the leaves of the syntax tree and thus are at most $|t|_{lo} + 1$, that is, the size of the syntax tree of t is bound by $2|t|_{lo} + 1$. Therefore, the considerations about size explosion in the introduction are unaffected by considering $|t|_{lo}$ rather than the size of its syntax tree.

The leftmost type system. The typing rules are in Figure 8, where the only difference with the head case is in rule app_r^{lo}, that replaces rule app_r^{hd}: leftmost evaluation enters into arguments and so the added argument p now also has to be typed, and with a tight constant. Note also a key difference between app_r^{lo} and app_b: in the former the argument p is typed exactly

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$$\frac{}{x : [A] \vdash^{(0,0)} x : A}\ \mathsf{ax} \qquad \frac{(\Delta_i \vdash^{(b_i,r_i)} t : A_i)_{i \in I}}{\uplus_{i\in I} \Delta_i \vdash^{(+_{i\in I} b_i, +_{i\in I} r_i)} t : [A_i]_{i\in I}}\ \mathsf{many}$$

$$\frac{\Gamma \vdash^{(b,r)} t : A}{\Gamma \setminus\!\!\setminus x \vdash^{(b+1,r)} \lambda x.t : \Gamma(x) \to A}\ \mathsf{fun}_b \qquad \frac{\mathtt{tight}(\Gamma(x)) \quad \Gamma \vdash^{(b,r)} t : \mathtt{tight}}{\Gamma \setminus\!\!\setminus x \vdash^{(b,r+1)} \lambda x.t : \mathtt{abs}}\ \mathsf{fun}_r$$

$$\frac{\Gamma \vdash^{(b,r)} t : M \to A \quad \Delta \vdash^{(b',r')} p : M}{\Gamma \uplus \Delta \vdash^{(b+b',r+r')} tp : A}\ \mathsf{app}_b \qquad \frac{\Gamma \vdash^{(b,r)} t : \mathtt{neutral} \quad \Delta \vdash^{(b',r')} p : \mathtt{tight}}{\Gamma \uplus \Delta \vdash^{(b+b',r+r'+1)} tp : \mathtt{neutral}}\ \mathsf{app}_r^{lo}$$

Fig. 8. Type system for leftmost evaluation.

once (that is, the type is not a multi-set), because its leftmost normal form p' appears exactly once in the leftmost normal form $t'p'$ of tp (where t' is the leftmost normal form of t), while in the latter it can be typed any number of times, depending on the cardinality of M.

We write $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$, to stress that the type derivation is built out of the rules of the leftmost type system.

As in the head case, the size of a typing derivation $|\Phi|$ is the number of rules in Φ, not counting the occurrences of rule many. Here again, $|\Phi|$ gives an upper bound to the sum of the indices (b, r) on its final judgement: whenever $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$, we have $b + r \leq |\Phi|$.

Definition 4.2 (Tight and traditional derivations). *A derivation $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : B$ is* tight *if* $\mathtt{tight}(B)$ *and* $\mathtt{tight}(\Gamma)$. *A derivation Ψ is* traditional *if it is tight-free, i.e. no tight type occurs in Φ (and therefore it does not use rules* fun_r *nor* app_r^{lo}*).*

Traditional derivations do not really play a role in this section, only in the next one. We introduce them here because the two sections share the same anti-substitution lemma.

4.1 Tight leftmost correctness

The proof of tight correctness of the type system follows exactly the same structure in sub-properties than for head evaluation. There are however two relevant differences. The first one is that without the tight hypothesis there is no bound on the leftmost size of normal forms – this shall be refined in Section 5. The second one is that quantitative subject reduction also holds only for tight derivations. Moreover, its proof is subtler and more involved.

Lemma 4.3 (Tight spreading on neutral terms). *Let t be such that* $\mathtt{neutral}_{hd}(t)$ *and $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$ be a typing derivation such that* $\mathtt{tight}(\Gamma)$. *Then* $\mathtt{tight}(A)$ *and the last rule of Φ is not* app_b.

Note that the lemma assumes $\mathtt{neutral}_{hd}(t)$ and not $\mathtt{neutral}_{lo}(t)$, and this is not a typo: as $\mathtt{neutral}_{lo}(t)$ implies $\mathtt{neutral}_{hd}(t)$, the lemma is stronger than if stated with

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$\mathtt{neutral}_{lo}(t)$. The proof of this lemma is the same of that of Lemma 3.3, but for the fact that app_r^{lo} replaces app_r^{hd}.

Proposition 4.4 (Properties of *lo* typings for normal forms)**.** *Let t be such that* $\mathtt{normal}_{lo}(t)$*, and* $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} t : A$ *be a tight type derivation. Then*

1. Tight indices*:* $b = 0$ *and* $r = |t|_{lo}$*. As a consequence* $|t|_{lo} \leq |\Phi|$*.*
2. Neutrality*: If* $A = \mathtt{neutral}$ *then* $\mathtt{neutral}_{lo}(t)$*.*

Proof The proof follows the same lines of the one for the head case. In particular, it uses the tight spreading on neutral terms exactly in the same way. See Appendix B.1. □

The substitution lemma also follows the same pattern of the head case.

Lemma 4.5 (Substitution and typings for *lo*)**.** *Let* $\Phi_t \triangleright_{lo} \Delta; x : M \vdash^{(b,r)} t : A$ *and* $\Phi_p \triangleright_{lo} \Gamma \vdash^{(b',r')} p : M$*. Then there exists a derivation* $\Phi_{t\{x\leftarrow p\}} \triangleright_{lo} \Gamma \uplus \Delta \vdash^{(b+b',r+r')} t\{x\leftarrow p\} : A$ *where* $|\Phi_{t\{x\leftarrow p\}}| = |\Phi_t| + |\Phi_p| - |M|$*. Moreover, if* Φ_t *and* Φ_p *are traditional, then* $\Phi_{t\{x\leftarrow p\}}$ *is traditional too.*

Proof See Appendix B.1. □

Leftmost subject reduction. To obtain the quantitative version of subject reduction, for which the b index decreases of exactly one, we are now forced to add the tight hypothesis. Basically, when reduction takes place in the argument, tightness ensures that the argument is typed only once. An example of argument that may go untyped is given by the following non-tight derivation Φ:

$$\dfrac{y : [[\,] \to X] \vdash^{(0,0)} y : [\,] \to X \qquad \dfrac{}{\vdash^{(0,0)} \Omega : [\,]}\,\mathsf{many}}{y : [[\,] \to X] \vdash^{(0,0)} y\Omega : X}\,\mathsf{app}_b$$

Note that $y\Omega \to_{lo} y\Omega$ and so the reduct is typed again by the derivation Φ and nothing has changed. Incidentally, this example shows that the type system is not even correct for $\to_{lo}$ termination. We then need a predicate restricting the set of derivations. Tightness is one such predicate, but in Section 5 we shall see another possible predicate.

Arguments may also be typed more than once. Consider the following non-tight derivation:

$$\dfrac{y : [[X,X] \to X] \vdash^{(0,0)} y : [X,X] \to X \qquad \dfrac{(z : [X] \vdash^{(0,0)} \mathtt{I}z : X)_{i=1,2}}{z : [X,X] \vdash^{(2,0)} \mathtt{I}z : [X,X]}\,\mathsf{many}}{y : [[X,X] \to X], z : [X,X] \vdash^{(2,0)} y(\mathtt{I}z) : X}\,\mathsf{app}_b \tag{1}$$

Now, $y(\mathtt{I}z) \to_{lo} yz$ and the corresponding derivation for yz is

$$\dfrac{y : [[X,X] \to X] \vdash^{(0,0)} y : [X,X] \to X \qquad \dfrac{(z : [X] \vdash^{(0,0)} z : X)_{i=1,2}}{z : [X,X] \vdash^{(0,0)} z : [X,X]}\,\mathsf{many}}{y : [[X,X] \to X], z : [X,X] \vdash^{(0,0)} yz : X}\,\mathsf{app}_b$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Both sub-derivations inside the many rule have been reduced, and so the first counter on the final judgement decreases by 2 and not by 1.

In both examples, if the derivation were tight then y would be typed with `neutral` and the last application rule would be app_r^{lo}, which requires the argument to be typed only once, avoiding both examples of inaccurate counting.

Adding the tight hypothesis impacts on the proof of subject reduction. The inductive cases of the proof change, because to apply the *i.h.* on a sub-derivation Φ one now needs to show that Φ is tight. In fact, since tightness is a global property not necessarily true for all sub-derivations, the proof actually proves a strengthened statement. Moreover, the last case relies crucially on the tight spreading on neutral terms (Lemma 4.3).

Proposition 4.6 (Quantitative tight subject reduction for lo)**.** *If $\Phi \rhd_{lo} \ \Gamma \vdash^{(b,r)} t : A$ is tight and $t \rightarrow_{lo} p$ then $b \geq 1$ and there exists a typing Φ' such that $\Phi' \rhd_{lo} \ \Gamma \vdash^{(b-1,r)} p : A$ and $|\Phi| > |\Phi'|$.*

Proof We prove the following stronger statement (tightness is decomposed in two predicates $\mathtt{tight}(\Gamma)$ and $\mathtt{tight}(A)$, and the second is paired together with a further assumption).

Let $t \rightarrow_{lo} p$, $\Phi \rhd_{lo} \ \Gamma \vdash^{(b,r)} t : A$, $\mathtt{tight}(\Gamma)$, and ($\mathtt{tight}(A)$ or $\neg\mathtt{abs}_{lo}(t)$). Then $b \geq 1$ and there exists a typing $\Phi' \rhd_{lo} \ \Gamma \vdash^{(b-1,r)} p : A$ and $|\Phi| > |\Phi'|$.

By induction on $t \rightarrow_{lo} p$. Cases:

- Rule

$$\frac{}{(\lambda x.u)q \rightarrow_{lo} u\{x{\leftarrow}q\}}$$

Assume $\Phi \rhd_{lo} \ \Gamma \vdash^{(b,r)} (\lambda x.u)q : A$ and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule app_b, and the derivation of its premiss for $\lambda x.u$ must end with fun_b. Hence, there are two derivations $\Phi_u \rhd_{lo} \ \Gamma_u; x : M \vdash^{(b_u,r_u)} u : A$ and $\Phi_q \rhd_{lo} \ \Gamma_q \vdash^{(b_q,r_q)} q : M$, with $(b,r) = (b_u + b_q + 1, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_q$. Applying the substitution Lemma 4.5, we obtain $\Phi' \rhd_{lo} \ \Gamma \vdash^{(b_u+b_q,r_u+r_q)} u\{x{\leftarrow}q\} : A$ and $|\Phi'| = |\Phi_u| + |\Phi_q| - |M| < |\Phi_u| + |\Phi_q| + 2 = |\Phi|$.
- Rule

$$\frac{u \rightarrow_{lo} q}{\lambda x.u \rightarrow_{lo} \lambda x.q}$$

Assume $\Phi \rhd_{lo} \ \Gamma \vdash^{(b,r)} \lambda x.u : A$ and $\mathtt{tight}(\Gamma)$. Since $\mathtt{abs}_{lo}(\lambda x.u)$ we must have hypothesis $\mathtt{tight}(A)$. Then, the last rule of Φ has to be with fun_r, and we must have a subderivation $\Phi_u \rhd_{lo} \ \Gamma, x : \mathtt{Tight} \vdash^{(b,r-1)} u : \mathtt{tight}$. As $\mathtt{tight}(\Gamma, x : \mathtt{Tight})$ we can apply the *i.h.* and get the premiss of the derivation Φ' in the following:

$$\frac{\Phi_q \rhd_{lo} \ \Gamma, x : \mathtt{Tight} \vdash^{(b-1,r-1)} q : \mathtt{tight}}{\Gamma \vdash^{(b-1,r)} \lambda x.q : \mathtt{abs}}$$

We conclude $|\Phi| = |\Phi_u| + 1 > |\Phi_q| + 1 = |\Phi'|$ thanks to the *i.h.* $|\Phi_u| > |\Phi_q|$.
- Rule

$$\frac{\neg\mathtt{abs}_{lo}(u) \quad u \rightarrow_{lo} q}{t = um \rightarrow_{lo} qm = p}$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Assume $\Phi \triangleright_{lo} \ \Gamma \vdash^{(b,r)} um : A$ and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule app_b or app_r^{lo}.

In both cases there are derivations $\Phi_u \triangleright_{lo} \ \Gamma_u \vdash^{(b_u,r_u)} u : A_u$ and $\Phi_m \triangleright_{lo} \ \Gamma_m \vdash^{(b_m,r_m)} m : A_m$ (A_m may be a multi-set), with $\Gamma = \Gamma_u \uplus \Gamma_m$. Since $\mathtt{tight}(\Gamma)$ we have $\mathtt{tight}(\Gamma_u)$, and since $\neg\mathtt{abs}_{lo}(u)$ we can apply the *i.h.* (even if A_u is not tight) obtaining the derivation $\Phi_q \triangleright_{lo} \ \Gamma_u \vdash^{(b_u-1,r_u)} q : A_u$ such that $|\Phi_q| < |\Phi_u|$. Now, using the same rule app_b or app_r^{lo} at the end of Φ, we build the following derivation Φ':

$$\frac{\Phi_q \triangleright_{lo} \ \Gamma_u \vdash^{(b_u-1,r_u)} q : A_u \qquad \Phi_m \triangleright_{lo} \ \Gamma_m \vdash^{(b_m,r_m)} m : A_m}{\Gamma \vdash^{(b-1,r)} qm : A}$$

that satisfies $|\Phi'| = |\Phi_q| + |\Phi_m| + 1 < |\Phi_u| + |\Phi_m| + 1 = |\Phi|$.

- Rule

$$\frac{\mathtt{neutral}_{lo}(m) \qquad u \to_{lo} q}{t = mu \to_{lo} mq = p}$$

Assume $\Phi \triangleright_{lo} \ \Gamma \vdash^{(b,r)} mu : A$ and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule app_b or app_r^{lo}, and therefore there are two derivations $\Phi_m \triangleright_{lo} \ \Gamma_m \vdash^{(b_m,r_m)} m : A_m$ and $\Phi_u \triangleright_{lo} \ \Gamma_u \vdash^{(b_u,r_u)} u : A_u$, for some types A_m and A_u (A_u may be a multi-set), with $\Gamma = \Gamma_m \uplus \Gamma_u$. Since $\mathtt{tight}(\Gamma)$ we have $\mathtt{tight}(\Gamma_m)$ and $\mathtt{tight}(\Gamma_u)$. By the *tight spreading on neutral terms* (Lemma 4.3), from $\mathtt{tight}(\Gamma_m)$ and $\mathtt{neutral}_{lo}(m)$ it follows $\mathtt{tight}(A_m)$. Therefore, the last rule of Φ must be app_r^{lo}, whence $A_m = A = \mathtt{neutral}$ and $A_u = \mathtt{tight}$. Now, the sub-derivation Φ_u is tight ($\mathtt{tight}(\Gamma_u)$ and $A_u = \mathtt{tight}$) and we can apply the *i.h.*, obtaining the derivation $\Phi_q \triangleright_{lo} \ \Gamma_u \vdash^{(b_u-1,r_u)} q : A_u$ such that $|\Phi_q| < |\Phi_u|$. Now, using the same rule app_b or app_r^{lo} at the end of Φ, we build the following derivation Ψ:

$$\frac{\Phi_m \triangleright_{lo} \ \Gamma_m \vdash^{(b_m,r_m)} m : A_m \qquad \Phi_q \triangleright_{lo} \ \Gamma_u \vdash^{(b_u-1,r_u)} q : A_u}{\Gamma \vdash^{(b-1,r)} mq : A}$$

that satisfies $|\Psi| = |\Phi_q| + |\Phi_m| + 1 < |\Phi_u| + |\Phi_m| + 1 = |\Phi|$. □

Correctness then follows by the same reasoning used for tight head derivations.

Theorem 4.7 (Tight correctness for *lo*). *Let $\Phi \triangleright_{lo} \ \Gamma \vdash^{(b,r)} t : A$ be a tight derivation. Then there exists p such that $t \to_{lo}^{b} p$, $\mathtt{normal}_{lo}(p)$, and $|p|_{lo} = r$. Moreover, if $A = \mathtt{neutral}$ then $\mathtt{neutral}_{lo}(p)$.*

Proof See Appendix B.1. □

Note that the statement of the correctness theorem is different from the one for head evaluation (Theorem 3.7), because here nothing is said about derivations that are not tight. The whole of Section 5 is devoted to the study of such derivations. Let us just sketch the point. Consider the following derivation that types the argument exactly once, as the tight derivation would. Consider now the non-tight type $A = [X] \to X$ and the following (non-tight) derivation:

$$\dfrac{\dfrac{z:[A]\vdash^{(0,0)} z:A}{\vdash^{(1,0)} \mathtt{I}:[A]\rightarrow A}\,\mathsf{fun}_b \qquad \dfrac{z:[X]\vdash^{(0,0)} z:X}{\vdash^{(1,0)} \mathtt{I}:[X]\rightarrow X}\,\mathsf{fun}_b}{\vdash^{(2,0)} \mathtt{I}\mathtt{I}:A}\,\mathsf{app}_b \tag{2}$$

Note that neither 2 is the length of the evaluation to normal form $\mathtt{I}$, nor 0 is the size of this normal form $\mathtt{I}$. Nonetheless, the derivation has the same structure as the tight one, and it contains the same basic information. In Section 5, we shall work out how to extract it.

4.2 Tight leftmost completeness

The proof of completeness of tight derivations for leftmost evaluation follows the same structure of completeness for the head case. There are some differences that are exactly the same differences between the correctness parts of the two systems. In particular, the proof of subject expansion is refined along the same lines of the proof of subject reduction.

Proposition 4.8 (Normal forms are tightly typable for lo)**.** *Let t be such that $\mathtt{normal}_{lo}(t)$. Then*

1. Existence*: there exists a tight derivation $\Phi \rhd_{lo}\ \Gamma \vdash^{(0,|t|_{lo})} t : A$.*
2. Structure*: moreover, if $\mathtt{neutral}_{lo}(t)$ then $A = \mathtt{neutral}$, and if $\mathtt{abs}_{lo}(t)$ then $A = \mathtt{abs}$.*
3. Unique size*: if Ψ is another tight derivation for t then $|\Phi| = |\Psi|$.*

Proof See Appendix B.2. □

Lemma 4.9 (Anti-substitution and typings for lo)**.** *Let $\Phi \rhd_{lo}\ \Gamma \vdash^{(b,r)} t\{x{\leftarrow}p\} : A$. Then there exist*

- *a multi-set M;*
- *a typing derivation $\Phi_t \rhd_{lo}\ \Gamma_t; x : M \vdash^{(b_t,r_t)} t : A$; and*
- *a typing derivation $\Phi_p \rhd_{lo}\ \Gamma_p \vdash^{(b_p,r_p)} p : M$*

such that

- Typing context*: $\Gamma = \Gamma_t \uplus \Gamma_p$;*
- Indices*: $(b,r) = (b_t + b_p, r_t + r_p)$.*
- Sizes*: $|\Phi| = |\Phi_t| + |\Phi_p| - |M|$.*
- *If Φ is traditional, then Φ_t and Φ_p are traditional too.*

Proof See Appendix B.2. □

The proof of quantitative subject expansion mimicks the elaborated one for subject reduction: it uses anti-substitution in the base case, it needs a strengthened hypothesis for the inductive cases, and it makes use of the tight spreading on neutral terms in the last inductive case. The unsurprising details are in the Appendix.

Proposition 4.10 (Quantitative tight subject expansion for lo)**.** *Let $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} p : A$ be a tight derivation. If $t \to_{lo} p$ then there exists a (tight) typing Ψ such that $\Psi \triangleright_{lo} \Gamma \vdash^{(b+1,r)} t : A$ and $|\Psi| > |\Phi|$.*

Proof See Appendix B.2. □

Theorem 4.11 (Tight completeness for lo)**.** *Let $t \to_{lo}^{k} p$ with $\mathtt{normal}_{lo}(p)$. Then*

1. Existence*: there exists a tight typing $\Phi \triangleright_{lo} \Gamma \vdash^{(k,|p|_{lo})} t : A$.*
2. Structure*: moreover, if $\mathtt{neutral}_{lo}(p)$ then $A = \mathtt{neutral}$, and if $\mathtt{abs}_{lo}(p)$ then $A = \mathtt{abs}$.*

Proof See Appendix B.2. □

5 Leftmost-outermost evaluation and shrinking typings

This section focuses on the leftmost(-outermost) evaluation system and on the relationship between tight and tight-free – deemed *traditional* in Definition 4.2 – typings. Contributions are manyfold:

1. *Leftmost normalisation, revisited*: we revisit the characterisation of leftmost normalising terms as those typable with *shrinking* typings, that is, those where the empty multi-set has no negative occurrences (Krivine, 1993). The insight is that the shrinking and tight constraints are of a very similar nature, showing that our technique is natural rather than ad-hoc. Moreover, our notion of shrinking derivation can also include the tight constants, thus we provide a strict generalisation of the characterisation in the literature.
2. *Proof technique*: the technical development follows the schema of the previous sections and differs considerably by others in the literature. The literature always relies, necessarily, on shrinking typings. Krivine uses the reducibility technique (because he deals with idempotent intersection types and cannot use the simpler size-decreasing technique allowed by multi types) (Krivine, 1993), de Carvalho also uses the reducibility technique (despite studying multi types) (de Carvalho, 2018), and Kesner and co-authors use the size-decreasing technique and rely on *typed redex occurrences* (Kesner & Ventura, 2014; Bucciarelli *et al.*, 2017). We use the size-decreasing technique and replace typed redex occurrences with a detailed study of how shrinking typings propagate, based on properties of neutral terms.
3. *Unitary shrinking typings*: we study a notion of minimal traditional typings, deemed *unitary shrinking*, that is a slight variation over principal types or de Carvalho's 1-typings (de Carvalho, 2018), playing a role akin to that of tight typings in the absence of the tight constants. The insight here is that tight typings are simply a device to focalise what traditional types can already observe in a somewhat more technical way.
4. *Type bound*: we show that for traditional shrinking derivations, the types in the last judgement provide a bound on the size of normal forms – with no reference to the type derivation – and this bound is exact if the typing is unitary shrinking and

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

minimal. This is a reformulation of a key point of de Carvalho's work (de Carvalho, 2018). The insight is then the inherent inadequacy of multi types as a tool for reliable complexity measures for the leftmost strategy, because of size explosion.

This study is done with respect to leftmost evaluation because among the case studies of the paper it is the most relevant one for reasonable cost models. It may however be easily adapted, *mutatis mutandis*, to the other systems.

Shrinking typings. It is standard to characterise leftmost normalising terms as those typable with intersection types without negative occurrences of Ω (Krivine, 1993), or, those typable with multi types without occurrence of the empty multi-set $[\,]$ (Bucciarelli *et al.*, 2017). We call this constraint *shrinking*. To explain it, let's recall the examples we considered before tight subject reduction for the leftmost strategy (Proposition 4.6). Consider the derivation of end sequent:

$$y : [[\,] \to X] \vdash^{(0,0)} y\Omega : X \tag{3}$$

Since $y\Omega$ is $\to_{lo}$-diverging, this derivation has to be excluded somehow. The problem here is that since y has an erasing type – that is an arrow type with $[\,]$ on the left – then the diverging subterm Ω does not get typed. Excluding the use of $[\,]$ is too drastic, because the paradigmatic erasing term $\lambda x.y$ is normal and can be typed only with:

$$y : [X] \vdash^{(1,0)} \lambda x.y : [\,] \to X$$

The idea is that only some occurrences of $[\,]$ are dangerous. The given examples seem to suggest that if $[\,]$ occurs on the right side of $\vdash$ is fine, while if it occurs in the typing context it is not. Things in fact are subtler. Extending example (3) with an abstraction, one obtains the $\to_{lo}$-diverging term $\lambda y.y\Omega$ and the typing

$$\vdash^{(1,0)} \lambda y.y\Omega : [[\,] \to X] \to X$$

that show that $[\,]$ can be dangerous also on the right. The correct shrinking constraint takes into account the polarity of the occurrence $[\,]$, which is the number of arrows for which the occurrence is in the left branch. The types associated to reducing sub-terms are exactly those of even polarity on the right of $\vdash$ and those of odd polarity on the left of $\vdash$, which then must not be $[\,]$.

Shrinking allows to capture termination (Krivine, 1993). Counting exactly the number of steps, however, requires a slight refinement. Not only all reducing subterms have to be typed (that is, they cannot be typed with $[\,]$), but they have to be typed *exactly once* – this is the *unitary* shrinking constraint. Let's go back to the example (1) of inaccurate counting of the previous section (p. 22), for instance.

$$y : [[X, X] \to X], z : [X, X] \vdash^{(2,0)} y(\mathtt{I}z) : X$$

Here the redex $\mathtt{I}z$ is typed twice, which can be seen by $[X, X]$ in the type of y, even if it is not duplicated. A unitary shrinking typing of $y(\mathtt{I}z)$ necessarily has a different end sequent and types $\mathtt{I}z$ only once:

$$\dfrac{y : [[X] \to X] \vdash^{(0,0)} y : [X] \to X \qquad \dfrac{z : [X] \vdash^{(1,0)} \mathtt{I}z : X}{z : [X] \vdash^{(1,0)} \mathtt{I}z : [X]}\,\mathsf{many}}{y : [[X] \to X], z : [X] \vdash^{(1,0)} y(\mathtt{I}z) : X}\,\mathsf{app}_b$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

We need some basic notions. We use the notation T to denote a *(multi)-type*, that is, either a type A or a multi-set of types M.

Definition 5.1 (Positive and negative occurrences)**.** *Let T be a (multi)-type. The sets of positive and negative occurrences of T in a type/multi-set of types/typing context are defined by mutual induction as follows:*

$$\frac{}{A \in \mathtt{Occ}_+(A)} \qquad \frac{\exists B \in M \text{ such that } T \in \mathtt{Occ}_+(B)}{T \in \mathtt{Occ}_+(M)} \qquad \frac{\exists B \in M \text{ such that } T \in \mathtt{Occ}_-(B)}{T \in \mathtt{Occ}_-(M)}$$

$$\frac{}{M \in \mathtt{Occ}_+(M)} \qquad \frac{T \in \mathtt{Occ}_-(M) \text{ or } T \in \mathtt{Occ}_+(A)}{T \in \mathtt{Occ}_+(M \to A)} \qquad \frac{T \in \mathtt{Occ}_+(M) \text{ or } T \in \mathtt{Occ}_-(A)}{T \in \mathtt{Occ}_-(M \to A)}$$

$$\frac{T \in \mathtt{Occ}_+(M) \text{ or } T \in \mathtt{Occ}_+(\Gamma)}{T \in \mathtt{Occ}_+(x : M, \Gamma)} \qquad \frac{T \in \mathtt{Occ}_-(M) \text{ or } T \in \mathtt{Occ}_-(\Gamma)}{T \in \mathtt{Occ}_-(x : M, \Gamma)}$$

Shrinking typings are defined by imposing a condition on the final judgement of the derivation, similarly to tight typings. It is technically convenient to also define its dual predicate, *being (unitary) co-shrinking*.

Definition 5.2 ((Unitary) shrinking typing)**.** *Let $\Phi \rhd_{lo} \ \Gamma \vdash^{(b,r)} t : A$ be a typing derivation.*

- *A is* shrinking *if $|M| \geq 1$ for all $M \in \mathtt{Occ}_+(A)$, and it is* unitary shrinking *if $|M| = 1$;*
- *A is* co-shrinking *if $|M| \geq 1$ for all $M \in \mathtt{Occ}_-(A)$, and it is* unitary co-shrinking *if $|M| = 1$;*
- *M is* shrinking/co-shrinking/unitary shrinking/unitary co-shrinking *if every $A \in M$ is;*
- *Γ is* co-shrinking *(resp. unitary co-shrinking) if M is co-shrinking (resp. unitary co-shrinking) for all type declarations $x : M$ in Γ;*
- *Φ is* shrinking *(resp.* unitary shrinking*) if A is shrinking (resp. unitary shrinking) and Γ is co-shrinking (resp. unitary co-shrinking).*

For example, $[\,] \in \mathtt{Occ}_-([\,] \to A)$, $[\,] \in \mathtt{Occ}_-([[\,] \to A, A])$ and $[[\,] \to A, A]$ is shrinking but not unitary, $[\,] \in \mathtt{Occ}_-(x : [[\,] \to A])$, and $[\,] \in \mathtt{Occ}_+([[\,] \to A] \to A)$ and $[[\,] \to A] \to A$ is unitary co-shrinking.

Note that

- *Final judgement*: being shrinking is a *local condition*, which depends only on the final judgement of a typing derivation, and that
- *Tight implies unitary shrinking*: a tight typing derivation is always shrinking, and even unitary shrinking.

In this section, we also have a close look to *traditional derivations*, that is, derivations without tight constants, see Definition 4.2 on p. 21.

We shall need a natural property of type occurrences (used in Lemma 5.6).

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Lemma 5.3 (Transitivity of polarities). *Let T, U, V be (multi)-types and $a, b \in \{+, -\}$. If $U \in \mathtt{Occ}_a(T)$ and $V \in \mathtt{Occ}_b(U)$ then $V \in \mathtt{Occ}_{\delta(a,b)}(T)$, where*

$$\delta(+,+) := + \qquad \delta(-,+) := - \qquad \delta(-,-) := + \qquad \delta(+,-) := -$$

Proof Easy induction on $U \in \mathtt{Occ}_a(T)$. See Appendix C for details. □

Type sizes. One of our results is that the types appearing in the final judgement of a derivation bound the size of *lo* normal forms, for traditional typings, according to the a notion of type size given next, and independently of the derivation itself. To give an idea, consider the easily derivable (unitary shrinking) typing

$$\vdash^{(1,0)} \lambda y.yy : [[X] \to X, X] \to X \tag{4}$$

There are 2 arrows in the type (judgement) and the normal form has leftmost size 2. Of course, one also has to take into account the arrow symbols appearing in the typing judgement, when present.

Note, however, that types in general only give an upper bound: taking the derivation of (4) and replacing X with $[X] \to X$ produces the derivable (unitary shrinking) judgement

$$\vdash^{(1,0)} \lambda y.yy : [[[X] \to X] \to [X] \to X, [X] \to X] \to [X] \to X$$

which has many more arrows than the size of the term.

Definition 5.4 (Type size). *The size $\#(\cdot)$ of types, multi-sets, and typing contexts is defined as follows:*

$$\begin{array}{rlrl} \#(X) := 0 & & \#(\mathtt{tight}) := 0 \\ \#(M) := \sum_{A \in M} \#(A) & & \#(M \to A) := \#(M) + \#(A) + 1 \\ \#(\epsilon) := 0 & & \#(x : M; \Gamma) := \#(M) + \#(\Gamma) \end{array}$$

Exact measures via unitary shrinking traditional typings. One of the aims of this section is to show how to exactly measure the number of steps and the size of normal forms without using tight constants, that is, using traditional derivations only. Essentially, this is done using unitary shrinking typings. The measurements however are more involved than in the tight case, as they have to be extracted from other information that can be found in the typing derivations. In particular, for traditional derivations, the index r is always 0 (it is only incremented by rules fun_r and app_r^{lo}), so that all the information is collapsed on the b index. The basic ideas are the following:

- The size $\#(\Gamma) + \#(A)$ in a unitary shrinking traditional typing $\Gamma \vdash^{(b,0)} t : A$ provides the size of the normal form of t.
- Being unitary shrinking ensures that b decreases by exactly 1 at every $\to_{lo}$ step (as in the tight case).
- Then $b - \#(\Gamma) - \#(A)$ gives the exact number of $\to_{lo}$ steps to normal form.

Things are however slightly more complex than as just described, for the following reasons:

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

1. *Unitary shrinking traditional type sizes are lax in general*. The first item above is not really true, only unitary shrinking traditional derivations giving *minimal* types do capture exactly the size of normal forms (otherwise they only provide a bound, as the example above shows). We build such typings in the completeness subsection (and so in the correctness part there are no exact measures).
2. *Sizes mismatch*: The third item above is also not really true, because, even when $\#(\Gamma) + \#(A)$ does match the size of normal forms, we have that b counts abstractions, so that we should subtract from it only the number of abstractions in the normal form (and not the applications). This is done by replacing the size of types with a polarised size extracting from types the number of abstractions, defined next.

Extracting the number of abstractions from types. Polarity as used for the shrinking predicate is the key concept to isolate the number of abstractions. Re-consider the following example:

$$\vdash^{(1,0)} \lambda y.yy : [[X] \to X, X] \to X$$

Note that arrows of positive polarity (that is, on the left branch of an even number of abstractions) count abstractions, and arrows of negative polarity count applications. Of course, things are reversed for types in the typing judgement, as the next example shows:

$$z : [[[X] \to X, X] \to X] \to X \vdash^{(1,0)} z\,(\lambda y.yy) : X$$

We use $|t|_\lambda$ to denote the number of abstractions in a term. The following refined notion of size for types shall be used to count the number of abstractions in the normal form.

Definition 5.5 (Polarised type size)**.** *The polarised sizes $\#^P(\cdot)$ and $\#^N(\cdot)$ of types, multi-sets, and typing contexts are defined as follows:*

$$\begin{array}{rcl@{\qquad}rcl}
\#^P(X) & := & 0 & \#^P(\mathtt{tight}) & := & 0 \\
\#^P(M) & := & \sum_{A \in M} \#^P(A) & \#^P(M \to A) & := & \#^N(M) + \#^P(A) + 1 \\
\#^P(\epsilon) & := & 0 & \#^P(x : M; \Gamma) & := & \#^P(M) + \#^P(\Gamma)
\end{array}$$

$$\begin{array}{rcl@{\qquad}rcl}
\#^N(X) & := & 0 & \#^N(\mathtt{tight}) & := & 0 \\
\#^N(M) & := & \sum_{A \in M} \#^N(A) & \#^N(M \to A) & := & \#^P(M) + \#^N(A) \\
\#^N(\epsilon) & := & 0 & \#^N(x : M; \Gamma) & := & \#^N(M) + \#^N(\Gamma)
\end{array}$$

5.1 Shrinking correctness

Here we show that shrinking typability is preserved by leftmost evaluation and that the size of shrinking typings decreases along it – hence the name – so that every shrinkingly typable term is leftmost normalising. Moreover, the b index of unitary shrinking typings decreases by exactly 1, as for tight typings (for shrinking derivations it may decrease of an arbitrary positive amount). For the sake of completeness, we also show that typability is always preserved, but if the typing is not shrinking then its size may not decrease.

Once more, we follow the abstract schema of the other sections, but replacing *tight* with (unitary) *shrinking*.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

We start, as usual, with a spreading property on neutral terms, expressed by the following lemma.

Lemma 5.6 (Occurrences spreading on neutral terms)**.** *Let t be such that $\mathtt{neutral}_{hd}(t)$ and $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$ be a typing derivation. Then A is a positive occurrence of Γ. Moreover, if Γ is co-shrinking (resp unitary co-shrinking) then A is co-shrinking (resp unitary co-shrinking).*

Proof See Appendix C.1. □

Two observations:

1. *Subsumption of tight spreading*. This lemma subsumes the *tight spreading on neutral terms* (Lemma 3.3). Indeed, if the typing context Γ is tight, the fact that A is a positive occurrence of Γ implies that A is tight.
2. *Being co-shrinking spreads*. Note that in the tight case the corresponding lemma allows to conclude that the derivation is tight, while here we cannot conclude that the derivation is shrinking, because it is *being co-shrinking* that spreads, giving that A is co-shrinking, while to obtain that Φ is shrinking we would instead need that A is shrinking.

Properties of normal forms. For normal forms, we prove two properties. First, the shrinking hypothesis allows to use type derivations to bound the size of normal forms. Moreover, the index b provides a bound to the number of abstractions in the normal form.

Similarly to the case of tight subject reduction for system lo (Proposition 4.6), the next three propositions require a slightly strengthened statement, having as particular case what we are actually interested in, that is, that the derivation is shrinking.

Proposition 5.7 (Shrinking derivations bound the size of normal forms)**.** *Let $\mathtt{normal}_{lo}(t)$ and $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$ be a derivation, and let $|\Phi|_{\mathsf{ax}}$ denote the number of axiom rules in Φ.*

1. *If Γ is co-shrinking and (A is shrinking or t is not an abstraction) then $|t|_{lo} \leq |\Phi| - |\Phi|_{\mathsf{ax}}$. Moreover, if Φ is traditional then $|t|_\lambda \leq b$.*
2. *If Γ is unitary co-shrinking and (A is unitary shrinking or t is not an abstraction) then $|t|_{lo} = |\Phi| - |\Phi|_{\mathsf{ax}}$. Moreover, if Φ is traditional then $|t|_\lambda = b$.*

Proof By induction on t. Note that $\mathtt{neutral}_{lo}$ implies $\mathtt{normal}_{lo}$ and so we can apply the *i.h.* when $\mathtt{neutral}_{lo}$ holds on some subterm of t. See Appendix C.1. □

The second property of normal forms is relative to traditional derivations, for which (size of) the types in the final judgement – rather than the type derivation – bound the size of the normal form. Moreover, the index b is bound by the polarised sizes of such types. As in the previous sections, neutral terms play a key role, showing that our isolation of the relevance of neutral terms for characterisation via multi types is not specific to tight types.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proposition 5.8 (Traditional types bounds the size of neutral and normal terms). *Let $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} t : A$ be a traditional derivation such that Γ is co-shrinking. Then*

1. *if $\mathtt{neutral}_{lo}(t)$ then $\#(A) + |t|_{lo} \leq \#(\Gamma)$ and $\#^N(A) + b \leq \#^N(\Gamma)$;*
2. *if $\mathtt{normal}_{lo}(t)$ and A is shrinking then $|t|_{lo} \leq \#(\Gamma) + \#(A)$ and $b \leq \#^N(\Gamma) + \#^P(A)$.*

Proof See Appendix C.1. □

The substitution lemma for the *lo* system has already been proved in Section 3 (Lemma 4.5).

As usual, shrinking correctness is based on a subject reduction property. Note that for unitary shrinking derivations b decreases by exactly 1.

Proposition 5.9 (Shrinking subject reduction). *Let $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} t : A$. If $t \rightarrow_{lo} p$ then $b \geq 1$ and there exists Ψ such that $\Psi \triangleright_{lo} \Gamma \vdash^{(b',r)} p : A$ with $b' \leq b$ and $|\Psi| \leq |\Phi|$. Moreover, Φ traditional implies Ψ traditional, and if Φ is shrinking (resp. unitary shrinking) then $b' < b$ (resp. $b' = b - 1$) and $|\Psi| < |\Phi|$.*

Proof See Appendix C.1. □

Note that a leftmost diverging term like $x(\delta\delta)$ is typable in system *lo* by assigning to x the type $[\,] \rightarrow X$ and typing $\delta\delta$ with $[\,]$, and that its type is preserved by leftmost evaluation, by Proposition 5.9. Note however that the resulting judgement is not shrinking – only shrinkingly typable terms are leftmost normalising, in fact.

Theorem 5.10 (Shrinking correctness). *Let $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} t : A$ be a shrinking derivation. Then there exists p such that $\mathtt{normal}_{lo}(p)$ and $k \leq b$ such that*

1. Steps*: t $\rightarrow_{lo}$-evaluates to p in k steps, i.e. $t \rightarrow_{lo}^{k} p$;*
2. Size bound*: $|p|_{lo} + k \leq |\Phi|$;*

Moreover, if Φ is traditional then $|p|_{lo} \leq \#(\Gamma) + \#(A)$ and $|p|_{\lambda} \leq \#^N(\Gamma) + \#^P(A)$, and if Φ is also unitary shrinking then $|p|_{\lambda} = b - k$.

Proof See Appendix C.1. □

Note that when Φ is unitary shrinking it does not follow that $|p|_{lo} = \#(\Gamma) + \#(A)$ and $|p|_{\lambda} = \#^N(\Gamma) + \#^P(A)$. The equalities indeed hold only if additionally the types in the last judgement of Φ are minimal. Such minimal derivations are built in the proof of Proposition 5.11.

5.2 Shrinking completeness

The proof of completeness for shrinking typings also follows, *mutatis mutandis*, the usual schema. Normal forms and anti-substitution have already been treated (Proposition 3.8 and Lemma 3.9). Again, however, we repeat the study of (the existence of typings for) leftmost normal forms focussing now on traditional typings and on the bound provided by types. Their study is yet another instance of *spreading on (leftmost) neutral terms*, in this case of the size bound provided by types: for neutral terms the size of the typing context Γ allows bounding both the size of the term and the size of its type, which is stronger than what happens for general leftmost normal terms.

One of the key point of the following proposition is that its proofs builds typing judgements having types of minimal size, refining Proposition 5.8.

Proposition 5.11 (Neutral and normal terms have minimal traditional shrinking typings)**.**

1. *If* $\mathtt{neutral}_{lo}(t)$ *then for every unitary co-shrinking type A there exists a traditional derivation* $\Phi \rhd_{lo} \Gamma \vdash^{(b,0)} t : A$ *such that Γ is unitary co-shrinking,* $\#(A) + |t|_{lo} = \#(\Gamma)$*, and* $\#^N(A) + b = \#^N(\Gamma)$*.*
2. *If* $\mathtt{normal}_{lo}(t)$ *then there exists a traditional unitary shrinking derivation* $\Phi \rhd_{lo} \Gamma \vdash^{(b,0)} t : A$ *such that* $|t|_{lo} = \#(\Gamma) + \#(A)$ *and* $b = \#^N(\Gamma) + \#^P(A)$*.*

Proof By mutual induction on $\mathtt{neutral}_{lo}(t)$ and $\mathtt{normal}_{lo}(t)$. Point 1 is along the lines of the case of Proposition 5.8. Here we only show the proof of Point 2; see Appendix C.2 for the full proof. Cases of $\mathtt{normal}_{lo}(t)$:

1. $\mathtt{neutral}_{lo}(t)$. By *i.h.* (point 1), *for every* unitary co-shrinking type A there exists a traditional typing $\Phi \rhd_{lo} \Gamma \vdash^{(b,0)} t : A$ such that Γ is unitary co-shrinking. It is then enough to pick $A := X$, that is both unitary shrinking and unitary co-shrinking, so that Φ is unitary shrinking, $\#(A) = 0$, and the statement trivially holds, because then $|t|_{lo} = \#(A) + |t|_{lo} =_{i.h.} \#(\Gamma) = \#(\Gamma) + \#(A)$. Moreover, $\#^P(A) = \#^N(A) = 0$, so that by *i.h.* $\#^N(A) + b = \#^N(\Gamma)$, which is equivalent to $b = \#^N(\Gamma) + \#^P(A)$, as required.
2. *Abstraction*, i.e. $t = \lambda y.p$ and $\mathtt{normal}_{lo}(p)$. By *i.h.* (point 2), there exists a unitary shrinking traditional typing $\Phi_p \rhd_{lo} \Gamma_p \vdash^{(b_p,0)} p : B$ with $|p|_{lo} = \#(\Gamma_p) + \#(B)$.

   Then let $y : M$ (M possibly $[\,]$) the declaration of y in Γ_p and set Γ be Γ_p without $y : M$. Then let Φ be the derivation

$$\frac{\Phi_p \rhd_{lo} y : M; \Gamma \vdash^{(b_p,0)} p : B}{\Gamma \vdash^{(b_p+1,0)} \lambda y.p : M \to B}\,\mathtt{fun}_b$$

   which is traditional and unitary shrinking because Φ_p is. We have

$$\begin{array}{rcl} |\lambda y.p|_{lo} & = & |p|_{lo} + 1 \\ & =_{i.h.} & \#(y : M; \Gamma) + \#(B) + 1 \\ & = & \#(\Gamma) + \#(M) + \#(B) + 1 \\ & = & \#(\Gamma) + \#(M \to B) \end{array}$$

   and

$$\begin{array}{rcl} b_p + 1 & =_{i.h.} & \#^N(y : M; \Gamma) + \#^P(B) + 1 \\ & = & \#^N(\Gamma) + \#^N(M) + \#^P(B) + 1 \\ & = & \#^N(\Gamma) + \#^P(M \to B) \end{array}$$

□

The last bit is a subject expansion property. Note in particular that since β-redexes are typed using traditional rules, the expansion preserves traditional typings.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proposition 5.12 (Shrinking subject expansion)**.** *If $t \to_{lo} p$ and $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} p : A$ then there exists Ψ such that $\Psi \triangleright_{lo} \Gamma \vdash^{(b',r)} t : A$ with $b' \geq b$. Moreover, if Φ is shrinking (resp. unitary shrinking) then $b' \geq b+1$ (resp. $b' = b+1$) and $|\Psi| > |\Phi|$, and if Φ is traditional then Ψ is traditional.*

Proof The proof is along the lines of the one for shrinking subject reduction, requiring the same kind of strengthened statement, see Appendix C.2. □

The completeness theorem then follows. We are finally able to measure exactly and separately both the number of steps and the size of the leftmost normal form via a traditional unitary shrinking derivation.

Theorem 5.13 (Shrinking completeness)**.** *Let $t \to_{lo}^{k} p$ with p such that $\texttt{normal}_{lo}(p)$. Then there exists a traditional unitary shrinking typing $\Phi \triangleright_{lo} \Gamma \vdash^{(b,0)} t : A$ such that $k = b - \#^{N}(\Gamma) - \#^{P}(A)$ and $|p|_{lo} = \#(\Gamma) + \#(A)$.*

Proof See Appendix C.2. □

Minimality. The minimality (with respect to size) of both tight and unitary shrinking derivations is implicitly contained in the statement of Proposition 5.7. For shrinking derivations, one has $|t|_{lo} \leq |\Phi| - |\Phi|_{\mathsf{ax}}$ and the equality holds exactly when the derivation is tight or unitary shrinking. The part about axioms is harmless: it is easily seen that for tight and unitary shrinking derivations the number of axioms is exactly the number of variable occurrences in the term (and so they all have the same size), and for shrinking derivations it is greater or equal to such number.

It is expected that the result holds more generally for all tight and unitary shrinking derivations, not just those for normal terms. Proving it, however, requires an (even more) involved study. Intuition tells that minimality can be pulled back to all typable terms via subject expansion. The problem is that subject expansion is formulated as an existential property (*there exists a derivation...*) and establishing minimality requires to compare the obtained expanded derivation with all the derivations for the expanded term, that may bear no similarity with the derivation in the hypothesis of subject expansion. A possible approach is to formalise subject reduction and expansion as *operations* over derivations (and not as existential properties). The precise definition of these operations is however very technical, because they can rewrite multi-sets of sub-derivations at once, if the rewriting step takes place in some arguments (in the term).

We estimated that the technical effort is not worth the minor additional result, given that this paper already has its good amount of technical material.

Type bounds and relational denotational semantics. The fact that for traditional typings the types in the final judgements provide a bound on the size of the normal form is a strong property. It is in particular the starting point for de Carvalho's transfer of the study of bounds to the relational semantics of terms (de Carvalho, 2007, 2018) – a term is interpreted as the set of its possible typings (thus including the typing context), that is a notion independent of the typing derivations themselves.

As we said in the introduction, multi types can be seen as a syntactic presentation of relational denotational semantics, which is the model obtained by interpreting the λ-calculus into the relational model of linear logic (Girard, 1988; Bucciarelli & Ehrhard, 2001; de Carvalho, 2007, 2016), often considered as a canonical model.

The idea is that the interpretation (or semantics) of a term is simply the set of its types, together with their typing contexts. More precisely, let t be a term and $x_1, \ldots, x_n$ (with $n \geq 0$) be pairwise distinct variables. If $\mathtt{fv}(t) \subseteq \{x_1, \ldots, x_n\}$, 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 of* t *for* $\vec{x}$ is

$$[\![t]\!]_{\vec{x}} := \{((M_1, \ldots, M_n), A) \mid \exists\, \Phi \rhd_{lo} x_1 : M_1, \ldots, x_n : M_n \vdash^{(b,r)} t : A \text{ such that } \Phi \text{ is shrinking}\}\,.$$

By subject reduction and expansion, the interpretation $[\![t]\!]_{\vec{x}}$ is an invariant of evaluation, and by correctness and completeness it is non-empty if and only if t is leftmost normalisable. Said differently, shrinking multi typing judgements provide an adequate denotational model with respect to the leftmost strategy. If the interpretation is restricted to traditional typing derivations, then it coincides with the one in the relational model in the literature. General derivations still provide a relational model, but a slightly different one, with the two new types `abs` and `neutral`, whose categorical semantics still has to be studied.

6 Extensions

In the rest of the paper, we are going to further explore the properties of the tight approach to multi types along two independent axes:

1. *Maximal evaluation*. We adapt the tight methodology to the case of maximal evaluation, which relates to strong normalisation in that the maximal evaluation strategy terminates only if the term being evaluated is strongly normalising. This case is a simplification of Bernadet & Graham-Lengrand (2013a) that can be directly related to the head and leftmost evaluation cases. It is in fact very close to leftmost evaluation but for the fact that, during evaluation, typing contexts are not necessarily preserved and the size of the terms being erased has to be taken into account. The statements of the properties have to be adapted accordingly.
2. *Linear head evaluation*. We reconsider head evaluation in the linear substitution calculus obtaining exact bounds on the number of steps and on the size of normal forms. The surprise here is that the type system is essentially unchanged and that it is enough to count also axiom rules (that are ignored for head evaluation in the λ-calculus) in order to exactly bound also the number of *linear substitution* steps.

Let us stress that these two variations on a theme can be read independently.

7 Maximal evaluation

In this section, we consider the maximal strategy, which gives the longest evaluation sequence from any strongly normalising term to its normal form. The maximal evaluation strategy is *perpetual* in that, if a term t has a diverging evaluation path then the maximal strategy diverges on t. Therefore, its termination subsumes the termination of any other

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

strategy, which is why it is often used to reason about strong normalisation property (van Raamsdonk *et al.*, 1999).

Strong normalisation and erasing steps. It is well-known that in the framework of relevant (i.e. without weakening) multi types it is technically harder to deal with strong normalisation (all evaluations terminate) – which is equivalent to the termination of the maximal strategy – than with weak normalisation (there exists a terminating evaluation) – which is equivalent to the termination of the leftmost strategy. The reason is that one has to ensure that all subterms that are erased along any evaluation are themselves strongly normalising.

The simple proof technique that we used in the previous section does not scale up – in general – to strong normalisation (or to the maximal strategy), because subject reduction breaks for erasing steps, as they change the final typing judgement. Of course the same is true for subject expansion. There are at least three ways of circumventing this problem:

1. *Memory*. To add a memory constructor, as in Klop's calculus (Klop, 1980), that records the erased terms and allows evaluation inside the memory, so that diverging subterms are preserved. Subject reduction then is recovered.
2. *Subsumption/weakening*. Adding a simple form of sub-typing, that allows stabilising the final typing judgement in the case of an erasing step, or more generally, adding a strong form of weakening, that essentially removes the empty multi type.
3. *Big-step subject reduction*: Abandon the preservation of the typing judgement in the erasing cases, and rely on a more involved big-step subject reduction property relating the term directly to its normal form, stating in particular that the normal form is typable, potentially by a different type.

Surprisingly, the tight characterisation of the maximal strategy that we are going to develop does not need any of these workarounds: in the case of tight typings subject reduction for the maximal strategy holds, and the simple proof technique used before adapts smoothly. To be precise, an evaluation step may still change the final typing judgement, but the key point is that the judgement stays tight. Morally, we are employing a form of subsumption of tight contexts, but an extremely light one, that in particular does not require a sub-typing relation. We believe that this is a remarkable feature of tight multi types.

Maximal evaluation and predicates. The maximal strategy shares with leftmost evaluation the predicates $\mathtt{neutral}_{lo}$, $\mathtt{normal}_{lo}$, $\mathtt{abs}_{lo}$, and the notion of term size $|t|_{lo}$, which we respectively write $\mathtt{neutral}_{max}$, $\mathtt{normal}_{max}$, $\mathtt{abs}_{max}$, and $|t|_{max}$. We actually define, in Figure 9, a version of the maximal strategy, denoted $\xrightarrow{r}_{max}$, that is indexed by an integer r representing the size of what is erased by the evaluation step. We define the transitive closure of $\xrightarrow{r}_{max}$ as follows:

$$\frac{}{t \xrightarrow{0}{}^{0}_{max} t} \qquad \frac{t \xrightarrow{r_1}_{max} p \quad p \xrightarrow{r_2}{}^{k}_{max} u}{t \xrightarrow{r_1+r_2}{}^{k+1}_{max} u} \qquad \frac{t \xrightarrow{r}{}^{k}_{max} p}{t \xrightarrow{r}{}^{*}_{max} p}$$

Proposition 7.1 (*max* evaluation system). *(*Λ, $\rightarrow_{max}$, $\mathtt{neutral}_{max}$, $\mathtt{normal}_{max}$, $\mathtt{abs}_{max}$*) is an evaluation system.*

$$\frac{x \in \mathtt{fv}(u)}{(\lambda x.u)q \xrightarrow{0}_{max} u\{x \leftarrow q\}} \qquad \frac{x \notin \mathtt{fv}(u) \quad \mathtt{normal}_{max}(q)}{(\lambda x.u)q \xrightarrow{|q|_{max}}_{max} u} \qquad \frac{x \notin \mathtt{fv}(u) \quad t \xrightarrow{r}_{max} p}{(\lambda x.u)t \xrightarrow{r}_{max} (\lambda x.u)p}$$

$$\frac{t \xrightarrow{r}_{max} p}{\lambda x.t \xrightarrow{r}_{max} \lambda x.p} \qquad \frac{\neg\mathtt{abs}_{max}(t) \quad t \xrightarrow{r}_{max} p}{tu \xrightarrow{r}_{max} pu} \qquad \frac{\mathtt{neutral}_{max}(u) \quad t \xrightarrow{r}_{max} p}{ut \xrightarrow{r}_{max} up}$$

Fig. 9. Maximal strategy.

Typing rules $\{\mathsf{ax}, \mathsf{fun}_b, \mathsf{fun}_r, \mathsf{app}_b, \mathsf{app}_r^{lo}\}$ in Figure 8 plus

$$\frac{|I| > 0 \quad (\Delta_i \vdash^{(b_i, r_i)} t : A_i)_{i \in I}}{\uplus_{i \in I} \Delta_i \vdash^{(+_{i \in I} b_i, +_{i \in I} r_i)} t : [A_i]_{i \in I}}\ \mathsf{many}_{>0} \qquad \frac{\Delta \vdash^{(b,r)} t : A}{\Delta \vdash^{(b,r)} t : [\,]}\ \mathsf{none}$$

Fig. 10. Type system for maximal evaluation.

Also in this case the proof is routine.

Multi types. Multi types are defined exactly as in Section 3. The type system *max* for *max*-evaluation is defined in Figure 10. Rules $\mathsf{many}_{>0}$ and none (which is a special 0-ary version of many), are both used to prevent an argument p in rule app_b to be *untyped*: either it is typed by means of rule $\mathsf{many}_{>0}$ – and thus it is typed with at least one type – or it is typed by means of rule none – and thus it is typed with exactly one type: the type itself is then forgotten, but requiring the premiss to have a type forces the term to be $\to_{max}$ normalising. The fact that arguments are always typed, even those that are erased during reduction, is essential to guarantee strong normalisation: system *max* cannot type anymore a term like $x\Omega$.

The next lemma expresses the *relevance* property of system *max*, that distinguishes it from the head and leftmost cases, and that can be proved by a straightforward induction on Φ.

Lemma 7.2 (Relevance). *Let $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : A$. Then $x \in \mathtt{fv}(t)$ if and only if $x \in \mathtt{dom}(\Gamma)$.*

The *size* $|\Phi|$ of a typing derivation Φ is naturally adapted to system *max*, counting all rule applications in Φ, except those of rules $\mathsf{many}_{>0}$ and none. And again if $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : A$ then $b + r \leq |\Phi|$.

Similarly to the head and leftmost cases, the quantitative information in typing derivations is used to characterise evaluation lengths and sizes of normal forms, as captured by the correctness and completeness theorems that we now present.

7.1 Tight correctness

The correctness theorem is proved following the same schema used for head and leftmost evaluations. Most proofs are similar, and are therefore omitted.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

We start with the properties of typed normal forms. The proof of the tight spreading on neutral terms (Lemma 4.3) also applies to typing system *max*, providing the following lemma.

Lemma 7.3 (Tight spreading on neutral terms for *max*)**.** *If* $\mathtt{neutral}_{hd}(t)$ *and* $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : A$ *such that* $\mathtt{tight}(\Gamma)$*, then* $\mathtt{tight}(A)$ *and the last rule of* Φ *is not* app_b*.*

The general properties of typed normal forms hold as well, using the same notion of tightness as in Definition 3.2.

Proposition 7.4 (Properties of typings for normal forms)**.** *Given* $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : A$ *with* $\mathtt{normal}_{max}(t)$*,*

1. Size bound*:* $|t| \leq |\Phi|$.
2. Tight indices*: if* Φ *is tight then* $b = 0$ *and* $r = |t|$.
3. Neutrality*: if* $A = \mathtt{neutral}$ *then* $\mathtt{neutral}_{max}(t)$.

Proof See Appendix D.1. □

We now turn to the typing derivations of terms that are not necessarily in normal form. The case of maximal evaluation starts differing from head and leftmost evaluations: indeed, rule none is not used in tight derivations of normal forms but must be used to type terms that are erased by reduction. If the types of such terms are left unconstrained, then precision is lost regarding the quantitative information that typing derivations contain about erasable terms. For maximal evaluation we must therefore strengthen the notion of tightness for typing derivations, which becomes a global condition because it is no longer a property of the final judgment only:

Definition 7.5 (Max-tight derivations)**.** *A derivation* $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : B$ *is* garbage-tight *if in every instance of rule* (none) *in* Φ *we have* $\mathtt{tight}(A)$*. It is* max-tight *if also* Φ *is tight, in the sense of Definition 3.2.*

Then we can type substitutions:

Lemma 7.6 (Substitution and typings for *max*)**.** *Let* $M \neq [\,]$*,* $\Phi_t \triangleright_{max} \Delta; x : M \vdash^{(b,r)} t : A$*, and* $\Phi_p \triangleright_{max} \Gamma \vdash^{(b,r)} p : M$*. Then there exists a derivation* $\Phi_{t\{x \leftarrow p\}} \triangleright_{max} \Gamma \uplus \Delta \vdash^{(b+b',r+r')} t\{x \leftarrow p\} : A$ *where* $|\Phi_{t\{x \leftarrow p\}}| = |\Phi_t| + |\Phi_p| - |M|$*. Moreover if* Φ_t *and* Φ_p *are garbage-tight, then so is* $\Phi_{t\{x \leftarrow p\}}$*.*

Note that the substitution lemma differs in two points with respect to the those for the head and leftmost cases (Lemmas 3.5 and 4.5):

1. *Relevance*: we assume that the multi-set M is not empty, so that the typing hypothesis for p is derived with rule $\mathsf{many}_{>0}$ rather than none. Note that indeed meta-level substitution is used in the definition of $t \xrightarrow{r}_{max} p$ only when the substituted variable x does occur, that by relevance (Lemma 7.2), corresponds to having x assigned to a non-empty multi-set M in the type context typing the body of the abstraction.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

2. *Garbage-tightness*: the substitution lemma has to ensure that garbage-tightness is preserved. This has no analogous on the head and leftmost cases because their notions of tightness only concern the final judgement, while here tightness has also an internal component, given precisely by garbage-tightness.

Nonetheless, the proof of the substitution lemma follows exactly the same schemas in the head and leftmost cases, and is therefore omitted.

Subject reduction. The statement of the subject reduction property here slightly differs from the corresponding ones for the head and leftmost cases. Indeed, if $t \xrightarrow{r}_{max} p$ then the typing environment Γ for term t is not necessarily preserved when typing p, because the evaluation step may erase a subterm of t. Consider for instance term $t = (\lambda x.x')(yy)$. In any *max*-typing derivation of t, the typing context must declare y with an appropriate type that ensures that, when applying a well-typed substitution to t, the resulting term is still normalising for $\rightarrow_{max}$. For instance, the context should declare $y : [[A] \rightarrow A, A]$, or even $y : [\mathtt{neutral}]$ if the typing derivation for t is max-tight. However, as $t \xrightarrow{1}_{max} x'$, the typing derivation for x' will clearly have a typing environment Γ' that maps y to $[\,]$. Hence, the subject reduction property has to take into account the change of typing context, as shown next. In what follows we write $\Gamma \sqsubseteq \Gamma'$ if $\Gamma(x) \sqsubseteq \Gamma'(x)$ for every variable x, where $\sqsubseteq$ denotes multi-set inclusion.

Proposition 7.7 (Quantitative tight subject reduction for *max*). *If $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : A$ is max-tight and $t \xrightarrow{e}_{max} p$, then there exist $\Gamma' \sqsubseteq \Gamma$ and an max-tight typing Ψ such that $\Psi \triangleright_{max} \Gamma' \vdash^{(b-1,r-e)} p : A$ and $|\Phi| > |\Psi|$.*

Proof As for the leftmost case (Proposition 4.6) we need to strengthen the statement, as follows:

Let $t \xrightarrow{e}_{max} p$, $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : A$ is garbage-tight, $\mathtt{tight}(\Gamma)$, and ($\mathtt{tight}(A)$ or $\neg\mathtt{abs}_{max}(t)$). Then there exist Γ' and a garbage-tight typing $\Psi \triangleright_{max} \Gamma' \vdash^{(b-1,r-e)} p : A$ such that $\mathtt{tight}(\Gamma')$.

We give here the two interesting cases of evaluation at top level: the non-erasing one, that requires the strengthen substitution lemma, and the erasing one, that modifies the type context. The full proof is in Appendix D.1.

- Non-erasing top-level step:

$$\frac{x \in \mathtt{fv}(u)}{(\lambda x.u)q \xrightarrow{0}_{max} u\{x \leftarrow q\}}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} (\lambda x.u)q : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule $\mathtt{app}_b$, the derivation of its premiss for $\lambda x.u$ must end with $\mathtt{fun}_b$. Hence, there are two garbage-tight derivations $\Phi_u \triangleright_{max} \Gamma_u; x : M \vdash^{(b_u,r_u)} u : A$ and $\Phi_p \triangleright_{max} \Gamma_p \vdash^{(b_p,r_p)} p : M$, with $(b,r) = (b_u + b_q + 1, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_p$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Moreover, by hypothesis $x \in \mathtt{fv}(u)$, and so $M \neq [\,]$ by relevance (Lemma 7.2). Then, the substitution lemma (Lemma 7.6) gives a garbage-tight derivation $\Psi \triangleright_{max} \Gamma \vdash^{(b_u+b_q, r_u+r_q)} u\{x{\leftarrow}q\} : A$ such that $|\Psi| = |\Phi_u| + |\Phi_q| - |M| < |\Phi_u| + |\Phi_q| + 2 = |\Phi|$.

- Erasing top-level step:

$$\frac{x \notin \mathtt{fv}(u) \qquad \mathtt{normal}_{max}(q)}{(\lambda x.u)q \xrightarrow{|q|_{max}}_{max} u}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} (\lambda x.u)q : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule app_b, and the derivation of its premiss for $(\lambda x.u)$ must end with fun_b. Moreover, since $x \notin \mathtt{fv}(u)$, then by relevance (Lemma 7.2) the derivation of its premiss q must end with rule none:

$$\dfrac{\dfrac{\Phi_u \triangleright_{max} \Gamma_u \vdash^{(b_u, r_u)} u : A}{\Gamma_u \vdash^{(b_u+1, r_u)} \lambda x.u : [\,] \to A}\,\mathsf{fun}_b \qquad \dfrac{\Phi_q \triangleright_{max} \Gamma_q \vdash^{(b_q, r_q)} q : A_q}{\Gamma_q \vdash^{(b_q, r_q)} q : [\,]}\,\mathsf{none}}{\Gamma_u \uplus \Gamma_q \vdash^{(b_u+b_q+1, r_u+r_q)} (\lambda x.u)q : A}\,\mathsf{app}_b$$

with $(b, r) = (b_u + b_q + 1, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_p$. Since Φ is garbage-tight, then Γ_q is tight and A_q must be $\mathtt{tight}$, and since $\mathtt{normal}_{max}(q)$, we can apply the tight indices property of normal forms (Proposition 7.4) and obtain $(b_q, r_q) = (0, |q|_{max})$, so that $(b_u, r_u) = (b - 1, r - |q|_{max})$. Since $\mathtt{tight}(\Gamma_u \uplus \Gamma_q)$ we have $\mathtt{tight}(\Gamma_u)$, so Φ_u is the desired garbage-tight derivation. Moreover, $|\Phi_u| < |\Phi_u| + |\Phi_q| + 2 = |\Phi|$. □

Correctness theorem. Now the correctness theorem easily follows. It differs from the corresponding theorem in Section 3.1 in that the second index in the max-tight typing judgement does not only measure the size of the normal form but also the sizes of all the terms erased during evaluation (and necessarily in normal form).

Theorem 7.8 (Tight correctness for *max*-evaluation). *Let $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : A$ be a max-tight derivation. Then there is an integer e and a term p such that $\mathtt{normal}_{max}(p)$, $t \xrightarrow{e}{}^{b}_{max} p$ and $|p|_{max} + e = r$. Moreover, if $A = \mathtt{neutral}$ then $\mathtt{neutral}_{max}(p)$.*

Proof See Appendix D.1. □

On removing the measure of erased terms. It is possible to slightly modify the definition of system *max* so that the second counter r is exactly the size $|p|_{max}$ of the normal form. Simply, one needs to modify the none rule as follows:

$$\frac{\Delta \vdash^{(b,r)} t : A}{\Delta \vdash^{(b,0)} t : [\,]}\,\mathsf{none}^0$$

Indeed, by setting the second counter to 0, rule none^0 ignores the size of erasable arguments.

7.2 Tight completeness

Completeness is again similar to completeness in the head and leftmost cases, and differs from them in the same way as correctness differs from their correctness. Namely, the second index in the completeness theorem also accounts for the size of erased terms, and the appendix provides the proof of the subject expansion property. The completeness statement follows.

Proposition 7.9 (Normal forms are tightly typable in max)**.** *Let t be such that* $\mathtt{normal}_{max}(t)$*. Then*

1. Existence*: there exists a max-tight derivation $\Phi \rhd_{max} \Gamma \vdash^{(0,|t|_{max})} t : A$.*
2. Structure*: Moreover, if $\mathtt{neutral}_{max}(t)$ then $A = \mathtt{neutral}$, and if $\mathtt{abs}_{max}(t)$ then $A = \mathtt{abs}$.*

Lemma 7.10 (Anti-substitution and typings for *max*)**.** *If $\Phi \rhd_{max} \Gamma \vdash^{(b,r)} t\{x{\leftarrow}p\} : A$ and $x \in \mathtt{fv}(t)$, then there exist*

- *a multi-set M different from $[\,]$;*
- *a typing derivation $\Phi_t \rhd_{max} \Gamma_t; x : M \vdash^{(b_t,r_t)} t : A$; and*
- *a typing derivation $\Phi_p \rhd_{max} \Gamma_p \vdash^{(b_p,r_p)} p : M$*

such that

- Typing context*: $\Gamma = \Gamma_t \uplus \Gamma_p$;*
- Indices*: $(b, r) = (b_t + b_p, r_t + r_p)$;*
- Sizes*: $|\Phi| = |\Phi_t| + |\Phi_p| - |M|$.*

Moreover, if Φ is garbage-tight then so are Φ_t and Φ_p.

Proposition 7.11 (Quantitative tight subject expansion for *max*)**.** *If $\Phi \rhd_{max} \Gamma \vdash^{(b,r)} p : A$ is max-tight and $t \xrightarrow{e}_{max} p$, then there exist $\Gamma' \sqsupseteq \Gamma$ and a max-tight typing Ψ such that $\Psi \rhd_{max} \Gamma' \vdash^{(b+1,r+e)} t : A$ and $|\Phi| < |\Psi|$. $\Psi \rhd_{max} \Gamma' \vdash^{(b+1,r+e)} t : A$ and $|\Phi| < |\Psi|$.*

Proof See Appendix D.2. □

Theorem 7.12 (Tight completeness for *max*)**.** *If $t \xrightarrow{e}{}^{k}_{max} p$ with $\mathtt{normal}_{max}(p)$, then there exists an max-tight typing $\Phi \rhd_{max} \Gamma \vdash^{(k,|p|_{max}+e)} t : A$. Moreover, if $\mathtt{neutral}_{max}(p)$ then $A = \mathtt{neutral}$, and if $\mathtt{abs}_{max}(p)$ then $A = \mathtt{abs}$.*

Proof See Appendix D.2. □

8 Linear head evaluation

In this section, we consider the linear version of the head evaluation system, where *linear* comes from the *linear substitution calculus* (LSC), a formalism that is a subtle reformulation of Milner's calculus with explicit substitutions (Milner, 2007; Kesner & Conchúir, 2008), which is inspired from the structural lambda-calculus (Accattoli & Kesner, 2010).

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

The linear substitution calculus has all the good properties one expect from a calculus with explicit substitutions, inherited from those of Milner's calculus (Kesner & Conchúir, 2008). It also has properties that no other calculus with explicit substitutions has, such as a residual system and a theory of standardisation (Accattoli *et al*., 2014).

Concretely, the LSC is a refinement of the λ-calculus where the language is extended with an explicit substitution constructor $t[x\backslash p]$, and *linear substitution* is a micro-step rewriting rule replacing one occurrence at a time – therefore, *linear* does not mean that variables have at most one occurrence, only that their occurrences are replaced one by one. Linear head evaluation – first studied in Mascari & Pedicini (1994) and Danos & Regnier (2004) – admits various presentations. The one in the LSC adopted here is the simplest one and has been introduced in Accattoli (2012).

The insight here is that switching from head to linear head, and from the λ-calculus to the LSC only requires counting $\mathtt{ax}$ rules for the size of typings and the head variable for the size of terms – the type system, in particular is the same. The correspondence between the two system is spelled out in the last subsection of this part. Of course, switching to the LSC some details have to be adapted: a further index traces linear substitution steps, there is a new typing rule to type the new explicit substitution constructor, and the proof schema slightly changes, as the (anti-)substitution lemma is replaced by a *linear* (anti-)substitution one – these are unavoidable and yet inessential modifications.

Thus, the main point of this section is to split the complexity measure among the multiplicative steps (beta steps) and the exponential ones (substitutions). Moreover, linear logic proof nets are known to simulate the λ-calculus, and LSC is known to be isomorphic to the proof-nets used in the simulation (Accattoli, 2018b). Therefore, the results of this section directly apply to those proof nets.

Explicit substitutions. We start by introducing the syntax of our language, which is given by the following set $\Lambda_{\mathtt{lsc}}$ of terms, where $t[x\backslash p]$ is a new constructor called *explicit substitution* (shortened ES), that is equivalent to $\mathtt{let}\ x = p\ \mathtt{in}\ t$:

$$\text{LSC TERMS} \qquad t, p \ ::= \ x \mid \lambda x.t \mid tp \mid t[x\backslash p]$$

The notion of *free* variable is defined as expected, in particular, $\mathtt{fv}(t[x\backslash p]) := (\mathtt{fv}(t) \setminus \{x\}) \cup \mathtt{fv}(p)$. *(List of) substitutions* and *linear head contexts* are given by the following grammars:

$$\begin{array}{rrcl} \text{(LIST OF) SUBSTITUTION CONTEXTS} & L & ::= & \langle\cdot\rangle \mid L[x\backslash t] \\ \text{LINEAR HEAD CONTEXTS} & H & ::= & \langle\cdot\rangle \mid \lambda x.H \mid Ht \mid H[x\backslash t] \end{array}$$

We write $L\langle t\rangle$ (resp. $H\langle t\rangle$) for the term obtained by replacing the hole $\langle\cdot\rangle$ in context L (resp. H) by the term t. This *plugging* operation, as usual with contexts, can capture variables. We write $H\langle\!\langle t\rangle\!\rangle$ when we want to stress that the context H does not capture the free variables of t.

Normal, neutral, and abs predicates. The predicate $\mathtt{normal}_{lhd}$ defining linear head normal terms and $\mathtt{neutral}_{lhd}$ defining linear head neutral terms are introduced in Figure 11. They are a bit more involved than before, because switching to the micro-step granularity of the LSC the study of normal forms requires a finer analysis. The predicates are now based on

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

LINEAR HEAD ABSTRACTIONS

$$\frac{}{\mathtt{abs}_{lhd}(\lambda x.t)} \qquad \frac{\mathtt{abs}_{lhd}(t)}{\mathtt{abs}_{lhd}(t[x\backslash p])}$$

LINEAR HEAD NORMAL FORMS

$$\frac{}{\mathtt{neutral}^{x}_{lhd}(x)}\ \mathtt{lhnvar} \qquad \frac{\mathtt{neutral}^{x}_{lhd}(t)}{\mathtt{neutral}^{x}_{lhd}(tp)}\ \mathtt{lhnapp} \qquad \frac{y \neq x \quad \mathtt{neutral}^{x}_{lhd}(t)}{\mathtt{neutral}^{x}_{lhd}(t[y\backslash p])}\ \mathtt{lhnsubx}$$

$$\frac{\mathtt{neutral}^{x}_{lhd}(t)}{\mathtt{normal}^{x}_{lhd}(t)}\ \mathtt{lhnno} \qquad \frac{y \neq x \quad \mathtt{normal}^{x}_{lhd}(t)}{\mathtt{normal}^{x}_{lhd}(\lambda y.t)}\ \mathtt{lhnolamx} \qquad \frac{y \neq x \quad \mathtt{normal}^{x}_{lhd}(t)}{\mathtt{normal}^{x}_{lhd}(t[y\backslash p])}\ \mathtt{lhnosubx}$$

$$\frac{\mathtt{normal}^{x}_{lhd}(t)}{\mathtt{normal}^{\#}_{lhd}(\lambda x.t)}\ \mathtt{lhnolamx} \qquad \frac{\mathtt{normal}^{\#}_{lhd}(t)}{\mathtt{normal}^{\#}_{lhd}(\lambda x.t)}\ \mathtt{lhnolam} \qquad \frac{\mathtt{normal}^{\#}_{lhd}(t)}{\mathtt{normal}^{\#}_{lhd}(t[y\backslash p])}\ \mathtt{lhnosub}$$

$$\frac{\mathtt{neutral}^{x}_{lhd}(t)}{\mathtt{neutral}_{lhd}(t)}\ \mathtt{lhnx} \qquad \frac{\mathtt{normal}^{x}_{lhd}(t)}{\mathtt{normal}_{lhd}(t)}\ \mathtt{lhnox} \qquad \frac{\mathtt{normal}^{\#}_{lhd}(t)}{\mathtt{normal}_{lhd}(t)}\ \mathtt{lhno}$$

Fig. 11. Linear head neutral and normal forms.

$$\frac{}{L\langle \lambda x.t\rangle u \to_{lhd} L\langle t[x\backslash u]\rangle}\ lhd_{\mathtt{m}} \qquad \frac{}{H\langle\!\langle x\rangle\!\rangle[x\backslash t] \to_{lhd} H\langle\!\langle t\rangle\!\rangle[x\backslash t]}\ lhd_{\mathtt{e}}$$

$$\frac{t \to_{lhd} u}{\lambda x.t \to_{lhd} \lambda x.u}\ lhd_{\lambda} \qquad \frac{\neg\mathtt{abs}_{lhd}(t) \quad t \to_{lhd} u}{tv \to_{lhd} uv}\ lhd_{@} \qquad \frac{t \neq H\langle\!\langle x\rangle\!\rangle \quad t \to_{lhd} u}{t[x\backslash v] \to_{lhd} u[x\backslash v]}\ lhd_{s}$$

Fig. 12. Linear-head strategy.

three auxiliary predicates $\mathtt{neutral}^{x}_{lhd}$, $\mathtt{normal}^{x}_{lhd}$, and $\mathtt{normal}^{\#}_{lhd}$: the first two characterise neutral and normal terms whose head variable x is free, the third instead characterises normal forms whose head variable is bound. Note also that the abstraction predicate $\mathtt{abs}_{lhd}$ is now defined *modulo* ES, that is, a term such as $(\lambda x.t)[z\backslash p][y\backslash u]$ satisfies the predicate. It is worth noticing that a term t of the form $H\langle\!\langle y\rangle\!\rangle$ does not necessarily verify $\mathtt{normal}_{lhd}(t)$, e.g. $(\lambda z.(yx)[x\backslash y])p$, because it has a multiplicative redex (defined next). Examples of linear head normal forms are $\lambda x.xy$ and $(yx)[x\backslash z](II)$.

Micro-step semantics. Linear head evaluation is often specified by means of a non-deterministic strategy having the diamond property (Accattoli, 2012). Here, however, we present a minor deterministic variant, in order to follow the general schema presented in the introduction. The deterministic notion of linear head evaluation *lhd* is in Figure 12. An example of $\to_{lhd}$-sequence is

$$\begin{array}{llll} ((\lambda z.(xx)[x\backslash y])p)[y\backslash w] & \to_{lhd} & (xx)[x\backslash y][z\backslash p][y\backslash w] & \to_{lhd} \\ (yx)[x\backslash y][z\backslash p][y\backslash w] & \to_{lhd} & (wx)[x\backslash y][z\backslash p][y\backslash w] & \end{array}$$

From now on, we split the evaluation relation $\to_{lhd}$ in two relations, *multiplicative* $\to_{\mathtt{m}}$ and *exponential* $\to_{\mathtt{e}}$ evaluation, where $\to_{\mathtt{m}}$ (resp. $\to_{\mathtt{e}}$) is generated by the base case

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$$\begin{array}{llll} |x|_{lhd} & := 1 & |\lambda x.t|_{lhd} & := |t|_{lhd} + 1 \\ |tu|_{lhd} & := |t|_{lhd} + 1 & |t[x\backslash u]|_{lhd} & := |t|_{lhd} \end{array}$$

Fig. 13. Head linear size of terms.

($lhd_{\mathtt{m}}$) (resp. ($lhd_{\mathtt{e}}$)) and closed by the three rules (lhd_λ), ($lhd_@$), (lhd_s). The terminology *multiplicative* and *exponential* comes from the linear logic interpretation of the LSC. The literature contains also an alternative terminology, using $\mathtt{B}$ *at a distance* for $\rightarrow_{\mathtt{m}}$ (or *distant* $\mathtt{B}$, where $\mathtt{B}$ is a common name for the variant of β introducing an ES instead of using meta-level substitution) and *linear substitution* for $\rightarrow_{\mathtt{e}}$.

Proposition 8.1 (linear head evaluation system)**.**
($\Lambda_{\mathtt{lsc}}, \rightarrow_{lhd}, \mathtt{neutral}_{lhd}, \mathtt{normal}_{lhd}, \mathtt{abs}_{lhd}$) is an evaluation system.

Proof In the linear case, the proof is subtler than for the head, LO, and maximal cases. It is in Appendix E. □

Size of normal forms. In Figure 13, the notion of *linear head size* $|t|_{lhd}$ extends the head size to terms with ES by counting 1 for variables – note that ES do not contribute to the linear head size. One way to understand why ES do not count for $|t|_{lhd}$ is having in mind the cost of recognizing a linear head normal form. In contrast to the previous cases, however, here one has to consider terms represented as proof nets, or, equivalently, a pointer-based representation of variables. In these cases, in the representation of $t[x\backslash p]$ the topmost constructor of t and the ES $[x\backslash p]$ are not next to each other, and there might not even be a path between the two. To check whether t is linear head normal then amounts to go through the left branch of the syntax tree, as for the head case, and once arrived on the head variable, verifying that the variable does not point to an ES (that is, it points to an abstraction or nowhere, if it is a free variable).

Multi types. We consider the same multi types of Section 3, but now typing judgements are of the form $\Gamma \vdash^{(b,e,r)} t : A$, where (b, e, r) is a *triple* of integers whose intended meaning is explained in the next paragraph. The typing system *lhd* is defined in Figure 14. By abuse of notation, we use for all the typing rules – except $\mathtt{ES}$ which is a new rule – the same names used for *hd*. As in the case of *hd* and *lo*, there is an alternative way to specify the functional rules, which is also applicable now to rule $\mathtt{ES}$. These formulations are often used in the technical proofs, they look as follows:

$$\frac{\Gamma; x : M \vdash^{(b,e,r)} t : A}{\Gamma \vdash^{(b+1,e+|M|,r-|M|)} \lambda x.t : M \rightarrow A}\ \mathsf{fun}_b \qquad \frac{\Gamma; x : \mathtt{Tight} \vdash^{(b,e,r)} t : \mathtt{tight}}{\Gamma \vdash^{(b,e,r+1)} \lambda x.t : \mathtt{abs}}\ \mathsf{fun}_r$$

$$\frac{\Gamma; x : M \vdash^{(b,e,r)} t : A \qquad \Delta \vdash^{(b',e',r')} u : M}{\Gamma \uplus \Delta \vdash^{(b+b',e+e'+|M|,r+r'-|M|)} t[x\backslash u] : A}\ \mathtt{ES}$$

As in the head and leftmost case, the size of a typing derivation $|\Phi|$ is the number of rules in Φ, not counting the occurrences of rule many.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$$\frac{}{x : [A] \vdash^{(0,0,1)} x : A}\ \mathsf{ax} \qquad \frac{(\Delta_i \vdash^{(b_i,e_i,r_i)} t : A_i)_{i\in I}}{\uplus_{i\in I}\Delta_i \vdash^{(+_{i\in I} b_i, +e_i, +_{i\in I} r_i)} t : [A_i]_{i\in I}}\ \mathsf{many}$$

$$\frac{\Gamma \vdash^{(b,e,r)} t : A}{\Gamma \setminus\!\!\setminus x \vdash^{(b+1,e+|\Gamma(x)|,r-|\Gamma(x)|)} \lambda x.t : \Gamma(x) \to A}\ \mathsf{fun}_b \qquad \frac{\mathtt{tight}(\Gamma(x)) \quad \Gamma \vdash^{(b,e,r)} t : \mathtt{tight}}{\Gamma \setminus\!\!\setminus x \vdash^{(b,e,r+1)} \lambda x.t : \mathtt{abs}}\ \mathsf{fun}_r$$

$$\frac{\Gamma \vdash^{(b,e,r)} t : M \to A \quad \Delta \vdash^{(b',e',r')} p : M}{\Gamma \uplus_{i\in I} \Delta_i \vdash^{(b+b',e+e',r+r')} tp : A}\ \mathsf{app}_b \qquad \frac{\Gamma \vdash^{(b,e,r)} t : \mathtt{neutral}}{\Gamma \vdash^{(b,e,r+1)} tu : \mathtt{neutral}}\ \mathsf{app}_r^{hd}$$

$$\frac{\Gamma \vdash^{(b,e,r)} t : A \qquad \Delta \vdash^{(b',e',r')} u : \Gamma(x)}{(\Gamma \setminus\!\!\setminus x) \uplus \Delta \vdash^{(b+b',e+e'+|\Gamma(x)|,r+r'-|\Gamma(x)|)} t[x\backslash u] : A}\ \mathsf{ES}$$

Fig. 14. Type system for linear head evaluation.

Indices. The roles of the three components of (b, e, r) in a typing derivation $\Gamma \vdash^{(b,e,r)} t : A$ can be described as follows:

- *b and multiplicative steps*. Similarly to the head case, b is supposed to bound the number of multiplicative redexes, i.e. the number of subterms of the form $L\langle\lambda x.t\rangle u$ that are reduced during an evaluation to normal form.
- *e and exponential steps*. The index e is supposed to bound the number of exponential redexes, i.e. subterms of the form $H\langle\langle x\rangle\rangle[x\backslash t]$ that are reduced during an evaluation to normal form. Note that e is incremented by axioms, and thus it counts the number of times an axioms is replaced by an exponential step. The ES typing rule does not change the index because a single ES can be involved in many exponential steps along an evaluation sequence.
- *r and size of the result*. r counts the rules typing variables, abstractions, and applications (i.e. ax, fun_r and app_r^{hd}) that cannot be consumed by *lhd* evaluation, so that they appear in the linear head normal form of a term. Note that the ES constructor is not considered the part of the head of terms.

Note also that the typing rules assume that variable occurrences (corresponding to ax rules) end up in the result, by having the third index set to 1. When a variable x becomes bound by an ES (rule ES) or by an abstraction destined to be applied (fun_b), the number of uses of x, expressed by the multiplicity of the multi-set M typing it, is subtracted from the size of the result, because those uses of x correspond to the times that it shall be replaced via a linear substitution step, and thus they should no longer be considered as contributing to the result. Coherently, that number instead contributes to the index tracing linear substitution steps.

Definition 8.2 (Tight derivations). *A derivation $\Phi \rhd_{lhd}\ \Gamma \vdash^{(b,e,r)} t : B$ is* tight *if* $\mathtt{tight}(B)$ *and* $\mathtt{tight}(\Gamma)$.

Example. Consider again the term $t_0 = (\lambda x_1.(\lambda x_0.x_0x_1)x_1)\mathtt{I}$, where $\mathtt{I}$ is the identity function $\lambda x_3.x_3$. The linear head evaluation sequence from t_0 to *lhd* normal-form is given next,

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

in which we distinguish the multiplicative steps from the exponential ones.

$$\begin{array}{llll}
(\lambda x_1.(\lambda x_0.x_0x_1)x_1)\mathtt{I} & \rightarrow_{\mathtt{m}} ((\lambda x_0.x_0x_1)x_1)[x_1\backslash\mathtt{I}] & \rightarrow_{\mathtt{m}} \\
(x_0x_1)[x_0\backslash x_1][x_1\backslash\mathtt{I}] & \rightarrow_{\mathtt{e}} (x_1x_1)[x_0\backslash x_1][x_1\backslash\mathtt{I}] & \rightarrow_{\mathtt{e}} \\
(\mathtt{I}x_1)[x_0\backslash x_1][x_1\backslash\mathtt{I}] & \rightarrow_{\mathtt{m}} x_3[x_3\backslash x_1][x_0\backslash x_1][x_1\backslash\mathtt{I}] & \rightarrow_{\mathtt{e}} \\
x_1[x_3\backslash x_1][x_0\backslash x_1][x_1\backslash\mathtt{I}] & \rightarrow_{\mathtt{e}} \mathtt{I}[x_3\backslash x_1][x_0\backslash x_1][x_1\backslash\mathtt{I}] &
\end{array}$$

The evaluation sequence has length 7:3 multiplicative steps and 4 exponential steps. The linear head normal form has size 2. We now give a tight typing for the term t_0, by writing again $\mathtt{abs}^{\mathtt{abs}}$ for $[\mathtt{abs}] \rightarrow \mathtt{abs}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{x_0 : [\mathtt{abs}^{\mathtt{abs}}] \vdash^{(0,0,1)} x_0 : \mathtt{abs}^{\mathtt{abs}} \qquad \dfrac{x_1 : [\mathtt{abs}] \vdash^{(0,0,1)} x_1 : \mathtt{abs}}{x_1 : [\mathtt{abs}] \vdash^{(0,0,1)} x_1 : [\mathtt{abs}]}}{x_0 : [\mathtt{abs}^{\mathtt{abs}}], x_1 : [\mathtt{abs}] \vdash^{(0,0,2)} x_0x_1 : \mathtt{abs}}}{x_1 : [\mathtt{abs}] \vdash^{(1,1,1)} \lambda x_0.x_0x_1 : [\mathtt{abs}^{\mathtt{abs}}] \rightarrow \mathtt{abs}} \qquad \dfrac{x_1 : [\mathtt{abs}^{\mathtt{abs}}] \vdash^{(0,0,1)} x_1 : \mathtt{abs}^{\mathtt{abs}}}{x_1 : [\mathtt{abs}^{\mathtt{abs}}] \vdash^{(0,0,1)} x_1 : [\mathtt{abs}^{\mathtt{abs}}]}}{\dfrac{x_1 : [\mathtt{abs}, \mathtt{abs}^{\mathtt{abs}}] \vdash^{(1,1,2)} (\lambda x_0.x_0x_1)x_1 : \mathtt{abs}}{\vdash^{(2,3,0)} \lambda x_1.(\lambda x_0.x_0x_1)x_1 : [\mathtt{abs}, \mathtt{abs}^{\mathtt{abs}}] \rightarrow \mathtt{abs}}} \qquad \dfrac{\vdots}{\vdash^{(1,1,2)} \mathtt{I} : [\mathtt{abs}, \mathtt{abs}^{\mathtt{abs}}]}}{\vdash^{(3,4,2)} (\lambda x_1.(\lambda x_0.x_0x_1)x_1)\mathtt{I} : \mathtt{abs}}$$

Indeed, the pair $(3, 4, 2)$ represents 3 (resp. 4) multiplicative (resp. exponential) evaluation steps to *lhd* normal-form, and a linear head normal form of size 2.

8.1 Tight correctness

As in the case of head and LO evaluation, the correctness proof is based on three main properties: properties of normal forms – themselves based on a lemma about neutral terms – the interaction between (linear head) substitution and typings, and subject reduction.

Neutral terms and properties of normal forms. As for the head case, the properties of tight typing of *lhd* normal forms depend on a spreading property of *lhd* neutral terms. Additionally, they also require a characterisation of the shape of type contexts for tight derivations of neutral and normal terms.

Lemma 8.3 (Tight spreading on neutral terms, plus typing contexts)**.** *Let* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} t : A$ *be a derivation.*

1. *If* $\mathtt{neutral}^{x}_{lhd}(t)$ *then* $x \in \mathtt{dom}(\Gamma)$*. Moreover, if* $\mathtt{tight}(\Gamma(x))$ *then* $\mathtt{tight}(A)$ *and* $\mathtt{dom}(\Gamma) = \{x\}$.
2. *If* $\mathtt{normal}^{x}_{lhd}(t)$ *then* $x \in \mathtt{dom}(\Gamma)$*. Moreover, if* $\mathtt{tight}(\Gamma(x))$ *then* $\mathtt{dom}(\Gamma) = \{x\}$.
3. *If* $\mathtt{normal}^{\#}_{lhd}(t)$ *and* $\mathtt{tight}(A)$ *then* $A = \mathtt{abs}$ *and* Γ *is empty.*

In all the cases, if $\mathtt{tight}(\Gamma)$*, then the last rule of* Φ *is not* $\mathtt{app}_b$.

Note that the points 1 and 2 together imply that if $\Gamma \vdash^{(b,e,r)} H\langle\langle x\rangle\rangle : A$ then $x \in \mathtt{dom}(\Gamma)$, i.e. $\Gamma = \Gamma'; x : M$ with $M \neq [\,]$.

Proof By induction on Φ. See Appendix E.1. □

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proposition 8.4 (Properties of *lhd* tight typings for normal forms)**.** *Let t be such that* $\mathtt{normal}_{lhd}(t)$*, and* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} t : A$ *be a typing derivation.*

1. Size bound*:* $|t|_{lhd} \leq |\Phi|$.
2. Tightness*: if* Φ *is tight then* $b = e = 0$ *and* $r = |t|_{lhd}$.
3. Neutrality*: if* $A = \mathtt{neutral}$ *then* $\mathtt{neutral}_{lhd}(t)$.

Proof The proof is by induction on Φ. We only show here the interesting case which allows to understand the use of Lemma 8.3, the full proof can be found in Appendix E.1. Let $t = p[x\backslash u]$, whose derivation Φ has the following form:

$$\dfrac{\Psi \triangleright_{lhd} \Delta; x : M \vdash^{(b',e',r')} p : A \qquad \Pi \vdash^{(b'',e'',r'')} u : M}{\Delta \uplus \Pi \vdash^{(b'+b'',e'+e''+|M|,r'+r''-|M|)} p[x\backslash u] : A}\,\mathtt{ES}$$

with $b = b' + b''$, $e = e' + e''$, $r = r' + r''$, and $\Gamma = \Delta \uplus \Pi$.

1. *Size bound*: by *i.h.* $|p|_{lhd} \leq |\Psi|$. Then $|t|_{lhd} = |p|_{lhd} \leq_{i.h.} |\Psi| < |\Phi|$.
2. *Tight bound*: There are two cases:

   - $\mathtt{normal}^{y}_{lhd}(p)$ *for some* $y \neq x$. By Lemma 8.3.2 $y \in \mathtt{dom}(\Delta)$. All assignments in Δ are $\mathtt{Tight}$ because Φ is tight, and so applying Lemma 8.3.2 again we obtain that $\mathtt{dom}(\Delta) = \{y\}$, that is, that $M = [\,]$. Two consequences: first, the $\mathtt{ES}$ has no right premiss, that is, it rather has the following shape:

     $$\dfrac{\Psi \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} p : A}{\Gamma \vdash^{(b,e,r)} p[x\backslash u] : A}\,\mathtt{ES}$$

     second, Ψ is tight, and so by *i.h.* $b = e = 0$ and $r = |p|_{lhd}$. The statement follows from the fact that $|p|_{lhd} = |p[x\backslash u]|_{lhd}$.
   - $\mathtt{normal}^{\#}_{lhd}(p)$. If Φ is tight then $A = \mathtt{tight}$ and by Lemma 8.3.3 the context $\Delta; x : M$ is empty, that is, $M = [\,]$. Two consequences: first, the $\mathtt{ES}$ has no right premiss, that is, it rather has the following shape:

     $$\dfrac{\Psi \triangleright_{lhd} \vdash^{(b,e,r)} p : A}{\vdash^{(b,e,r)} p[x\backslash u] : A}\,\mathtt{ES}$$

     second, Ψ is tight, and so by *i.h.* $b = e = 0$ and $r = |p|_{lhd}$. The statement follows from the fact that $|p|_{lhd} = |p[x\backslash u]|_{lhd}$. □

Linear substitution lemma. The main difference in the proof schema with respect to the head case is about the substitution lemma, that is now expressed differently, because evaluation no longer relies on meta-level substitution. Linear substitutions consume one type at a time: performing a linear head substitution on a term of the form $H\langle\!\langle x\rangle\!\rangle[x\backslash t]$ consumes *exactly* one type resource associated to the variable x, and all the other ones remain in the typing context after the partial substitution.

Lemma 8.5 (Linear substitution and typings for *lhd*)**.** *Let* $\Phi \triangleright_{lhd} x : M; \Gamma \vdash^{(b,e,r)} H\langle\!\langle x\rangle\!\rangle : A$. *Then there exists* $B \in M$ *such that for all* $\Phi_t \triangleright_{lhd} \Gamma_t \vdash^{(b_t,e_t,r_t)} t : B$ *there exists a derivation* $\Psi \triangleright_{lhd} x : M \setminus [B]; \Gamma \uplus \Gamma_t \vdash^{(b+b_t,e+e_t,r+r_t-1)} H\langle\!\langle t\rangle\!\rangle : A$*. Moreover,* $|\Psi| = |\Phi| + |\Phi_t| - 1$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proof By induction on H. See Appendix E.1. □

Subject reduction. Quantitative subject reduction is also refined, by taking into account the fact that now there are two evaluation steps, whose numbers are traced by two different indices.

Proposition 8.6 (Quantitative subject reduction for *lhd*)**.** *If* $\Phi \triangleright \Gamma \vdash^{(b,e,r)} t : A$, *then*

1. *if* $t \to_{\mathtt{m}} u$ *then* $b \geq 1$ *and there is a typing* Φ' *such that* $\Phi' \triangleright \Gamma \vdash^{(b-1,e,r)} u : A$ *and* $|\Phi'| = |\Phi| - 1.$;
2. *if* $t \to_{\mathtt{e}} u$ *then* $e \geq 1$ *and there is a typing* Φ' *such that* $\Phi' \triangleright \Gamma \vdash^{(b,e-1,r)} u : A$ *and* $|\Phi'| = |\Phi| - 1$.

Proof The proof is by induction on $t \to_{\mathtt{m}} u$ and $t \to_{\mathtt{e}} u$, using Lemma 8.5. See Appendix E.1. □

Note that quantitative subject reduction does not assume that the typing derivation is tight: as for the head case, the tight hypothesis is only used for the study of normal forms – it is needed for subject reduction/expansion only if evaluation can take place inside arguments, as in the leftmost and maximal cases.

Note also that the size of derivations decreases of *exactly* 1 – it follows from the *moreover* part of the linear substitution lemma. This fact contrasts strikingly with respect to the other subject reduction properties in the paper, where it is not possible to have such an uniform bound, because they adopt an operational semantics based on meta-level (full) substitution, that may replace many or no variable occurrences at the same time. This is one of the reasons behind our slogan that multi types more naturally measure evaluation in the LSC rather than in the λ-calculus.

Correctness. According to the spirit of tight typings, linear head correctness does not only provide the size of (linear head) normal forms, but also the lengths of evaluation sequences to (linear head) normal form: the two first integers b and e in the final judgement count exactly the total number of evaluation steps to (linear-head) normal form.

Theorem 8.7 (Tight correctness for *lhd*)**.** *Let* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} t : A$ *be a tight derivation. Then there exists* p *such that* $t \to_{lhd}^{b+e} p$, $\mathtt{normal}_{lhd}(p)$ *and* $|p|_{lhd} = r$. *Moreover, if* $A = \mathtt{neutral}$ *then* $\mathtt{neutral}_{lhd}(p)$.

Proof See Appendix E.1. □

8.2 Tight completeness

As in the case of head and LO evaluation, the completeness proof is based on the following properties: typability of linear head normal forms, interaction between (linear head) anti-substitution and typings, and subject expansion. The proofs are analogous to those of the completeness for head and LO evaluation, up to the changes for the linear case, that are instead analogous to those of the correctness of the previous subsection. The statements follow.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proposition 8.8 (Linear head normal forms are tightly typable for *lhd*)**.** *Let t be such that* $\mathtt{normal}_{lhd}(t)$*. Then there exists a tight typing* $\Phi \triangleright_{lhd} \Gamma \vdash^{(0,0,|t|_{lhd})} t : A$*. Moreover, if* $\mathtt{neutral}_{lhd}(t)$ *then* $A = \mathtt{neutral}$*, and if* $\mathtt{abs}_{lhd}(t)$ *then* $A = \mathtt{abs}$.

Proof See Appendix E.2. □

Lemma 8.9 (Linear anti-substitution and typings for *lhd*)**.** *Let* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} H\langle\langle u\rangle\rangle : A$*, where* $x \notin u$*. Then there exists*

- *a type* B
- *a typing derivation* $\Phi_u \triangleright_{lhd} \Gamma_u \vdash^{(b_u,e_u,r_u)} u : B$
- *a typing derivation* $\Phi_{H\langle\langle x\rangle\rangle} \triangleright_{lhd} \Gamma' \uplus x{:}[B] \vdash^{(b',e',r')} H\langle\langle x\rangle\rangle : A$

such that

- Typing contexts*:* $\Gamma = \Gamma' \uplus \Gamma_u$*.*
- Indices*:* $(b, e, r) = (b' + b_u, e' + e_u, r' + r_u - 1)$*.*
- Sizes*:* $|\Phi| = |\Phi_u| + |\Phi_{H\langle\langle x\rangle\rangle}| - 1$*.*

Proof By induction on H. See Appendix E.2. □

Proposition 8.10 (Quantitative subject expansion for *lhd*)**.** *If* $\Phi' \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} t' : A$ *then*

1. *if* $t \rightarrow_{\mathtt{m}} t'$ *then there is a derivation* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b+1,e,r)} t : \tau$ *and* $|\Phi'| = |\Phi| + 1$*;*
2. *if* $t \rightarrow_{\mathtt{e}} t'$ *then there is a derivation* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e+1,r)} t : A$ *and* $|\Phi'| = |\Phi| + 1$*.*

Proof See Appendix E.2. □

As for linear head correctness, linear head completeness also *refines* the information provided about the lengths of the evaluation sequences: the number k of evaluation steps to (linear head) normal form is now split into two integers k_1 and k_2 representing, respectively, the multiplicative and exponential steps in such evaluation sequence.

Theorem 8.11 (Tight completeness for *lhd*)**.** *Let* $t \rightarrow^{k}_{lhd} p$*, where* $\mathtt{normal}_{lhd}(p)$*. Then there exists a tight type derivation* $\Phi \triangleright_{lhd} \Gamma \vdash^{(k_1,k_2,|p|_{lhd})} t : A$*, where* $k = k_1 + k_2$*. Moreover, if* $\mathtt{neutral}_{lhd}(p)$*, then* $A = \mathtt{neutral}$*, and if* $\mathtt{abs}_{lhd}(p)$ *then* $A = \mathtt{abs}$.

Proof See Appendix E.2. □

8.3 Relationship between head and linear head

The head and linear head strategies are specifications at different granularities of the same notion of evaluation. Their type systems are also closely related – in a sense that we now make explicit, they are the same system.

In order to formalise this relationship we define the transformation $\mathcal{L}$ of *hd*-derivations into (linear, hence the notation) *lhd*-derivations as: ax in *hd* is mapped to ax in *lhd*, fun_r in *hd* is mapped to fun_r in *lhd*, fun_b in *hd* is mapped to fun_b in *lhd*, and so on. This transformation preserves the context and the type of all the typing judgements. Of course, if one

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

restricts the *lhd* system to λ-terms, there is an inverse transformation $\mathcal{N}$ of *lhd*-derivations into (non-linear, hence the notation) *hd*-derivations, defined as expected. Together, the two transformation realise an isomorphism.

Proposition 8.12 (Head isomorphism)**.** *Let t be a λ-term without explicit substitutions. Let* $|\cdot|_{\mathsf{ax}}$ *denote the number of axiom rules in a derivation. Then*

1. Non-linear to linear*: if* $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} t : A$ *then there exists* $e \geq 0$ *such that* $\mathcal{L}(\Phi) \triangleright_{lhd} \Gamma \vdash^{(b,e,r')} t : A$*, where* $r' = r - e + |\Phi|_{\mathsf{ax}}$*. Moreover,* $\mathcal{N}(\mathcal{L}(\Phi)) = \Phi$*.*
2. Linear to non-linear*: if* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e,r')} t : A$ *then* $\mathcal{N}(\Phi) \triangleright_{hd} \Gamma \vdash^{(b,r)} t : A$*, where* $r = e + r' - |\Phi|_{\mathsf{ax}}$*. Moreover,* $\mathcal{L}(\mathcal{N}(\Phi)) = \Phi$*.*

The proof is straightforward.

Morally, the same type system measures both head and linear head evaluations. The difference is that to measure head evaluation and head normal forms one forgets the number of axiom typing rules, that coincides exactly with the number of linear substitution steps, plus 1 for the head variable of the linear normal form. In this sense, multi types more naturally measure linear head evaluation. Roughly, a tight multi-type derivation for a term is nothing else but a coding of the evaluation in the LSC, including the normal form itself.

On the number of substitution steps. It is natural to wonder how the index e introduced by $\mathcal{L}$ in Proposition 8.12 is related to the other indices b and r. This kind of questions has been studied at length in the literature about reasonable cost models. It is known that $e = \mathcal{O}(b^2)$ for *any* λ-term, even for untypable ones, see Accattoli & Dal Lago (2012) for details. The bound is typically reached by the diverging term $\delta\delta$, which is untypable, but also by the following terminating (and therefore typable) term $t_n := (\lambda x_n. \ldots (\lambda x_1.(\lambda x_0.(x_0 x_1 \ldots x_n))x_1)x_2 \ldots x_n)I$. Indeed, t_n evaluates in $2n$ multiplicative steps (one for turning each β-redex into an ES, and one for each time that the identity comes in head position) and $\Omega(n^2)$ exponential steps.

On terms with ES. Relating typing judgements for λ-terms with ES to judgements for ordinary λ-terms is a bit trickier – we only sketch the idea. One needs to introduce the *unfolding* operation $(\cdot)\!\downarrow : \Lambda_{\mathtt{lsc}} \to \Lambda$ on λ-terms with ES, that turns all ES into meta-level substitutions, producing the underlying ordinary λ-term. For instance, $(x[x\backslash y][y\backslash z])\!\downarrow = z$. As in Proposition 8.12, types are preserved:

Lemma 8.13 (Unfolding and *lhd* derivations)**.** *Let* $t \in \Lambda_{\mathtt{lsc}}$*. If* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} t : A$ *then there exists* $\Psi \triangleright_{hd} \Gamma \vdash^{(b,r-1)} t\!\downarrow : A$*.*

Note that the indices are also preserved. It is possible to also spell out the relationship between Φ and Ψ (as done in Kesner *et al*., 2018), that simply requires a notion of unfolding of typing derivations, and that collapses on the transformation $\mathcal{N}$ in the case of ordinary λ-terms.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

9 Conclusions

Type systems provide guarantees both *internally* and *externally*. Internally, a typing discipline ensures that a program in isolation has a given desired property. Externally, the property is ensured *compositionally*: plugging a typed program in a typed environment preserves the desired property. Multi types (aka non-idempotent intersection types) are used in the literature to quantify the resources that are needed to produce normal forms. Minimal typing derivations provide exact upper bounds on the number of β-steps plus the size of the normal form – this is the internal guarantee. Unfortunately, such minimal typings provide almost no compositionality, as they essentially force the program to interact with a linear environment. Non-minimal typings allow compositions with less trivial environments, at the price of laxer bounds.

In this paper, we have engineered typing so that, via the use of *tight constants* among base types, some typing judgements express compositional properties of programs while other typing judgements, namely the *tight* ones, provide exact and separate bounds on the lengths of evaluation sequences on the one hand, and on the sizes of normal forms on the other hand. The distinction between the two counts is motivated by the size explosion problem, where the size of terms can grow exponentially with respect to the number of evaluation steps.

We conducted this study building on some of the ideas in Bernadet & Graham-Lengrand (2013a), by presenting a flexible and parametric typing framework, which we systematically applied to three evaluation strategies of the pure λ-calculus: head, leftmost-outermost, and maximal.

In the case of leftmost-outermost evaluation, we have also developed the traditional shrinking approach which does not make use of tight constants. One of the results is that the number of (leftmost) evaluation steps can be measured using only the (sizes) of the types of the final typing judgement, in contrast to the size of the *whole* typing derivation. Another point, is the connection between tight typings and minimal unitary shrinking typings without tight constants.

In the case of maximal evaluation, we have circumvented the traditional techniques to show strong normalisation: by focusing on the maximal deterministic strategy, we do not require any use of memory operator or subtyping for abstractions to recover subject reduction.

We have also extended our (pure) typing framework to linear head evaluation, presented in the linear substitution calculus (LSC). The result is that tight typings naturally encode evaluation in the LSC, which can be seen as the natural computing device behind multi types. In particular, and surprisingly, exact bounds for head and linear head evaluation rely on the same type system.

Acknowledgments

We are grateful to Alexis Bernadet, Antonio Bucciarelli, Giulio Guerrieri, Maico Leberle, Pierre Vial, Andrés Viso, and the anonymous reviewers for useful discussions and comments. This work has been partially funded by the ANR JCJC grant COCA HOLA (ANR-16-CE40-004-01).

Conflicts of Interest

None.

References

Accattoli, B. (2012) An abstract factorization theorem for explicit substitutions. In RTA'12. LIPIcs, Tiwari, A. (ed.), vol. 15, Nagoya, Japan. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 6–21.

Accattoli, B. (2018a) (In)Efficiency and reasonable cost models. *Electr. Notes Theor. Comput. Sci.* **338**, 23–43. https://doi.org/10.1016/j.entcs.2018.10.003

Accattoli, B. (2018b) Proof nets and the linear substitution calculus. In Theoretical Aspects of Computing - ICTAC 2018 - 15th International Colloquium, October 16–19, 2018, Proceedings, Stellenbosch, South Africa. Springer, pp. 37–61. https://doi.org/10.1007/978-3-030-02508-3_3

Accattoli, B., Bonelli, E., Kesner, D. & Lombardi, C. (2014) A nonstandard standardization theorem. In POPL 2014, Jagannathan, S. & Sewell, P. (eds), San Diego, California, USA. ACM, pp. 659–670.

Accattoli, B. & Dal Lago, U. (2012) On the invariance of the unitary cost model for head reduction. In RTA'12. LIPIcs, Tiwari, A. (ed.), vol. 15, Nagoya, Japan. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 22–37.

Accattoli, B. & Dal Lago, U. (2016) (Leftmost-Outermost) Beta-reduction is invariant, indeed. *Log. Methods Comput. Sci.* **12**(1), 1–46.

Accattoli, B., Graham-Lengrand, S. & Kesner, D. (2018). Tight typings and split bounds. *PACMPL* **2**(ICFP), 94:1–94:30. https://doi.org/10.1145/3236789

Accattoli, B. & Guerrieri, G. (2018) Types of fireballs. In Programming Languages and Systems - 16th Asian Symposium, APLAS 2018, December 2–6, 2018, Proceedings, Wellington, New Zealand. Springer, pp. 45–66. https://doi.org/10.1007/978-3-030-02768-1_3

Accattoli, B., Guerrieri, G. & Leberle, M. (2019). Types by need. In Programming Languages and Systems - 28th European Symposium on Programming, ESOP 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, April 6–11, 2019, Proceedings, Prague, Czech Republic. Springer, pp. 410–439. https://doi.org/10.1007/978-3-030-17184-1_15

Accattoli, B. & Kesner, D. (2010) The structural λ-calculus. In CSL'10, Brno, Czech Republic. Springer, pp. 381–395.

Alves, S., Kesner, D. & Ventura, D. (2019). A Quantitative Understanding of Pattern Matching. CoRR abs/1912.01914. arXiv:1912.01914. http://arxiv.org/abs/1912.01914

Avanzini, M. & Dal Lago, U. (2017). Automating sized-type inference for complexity analysis. *PACMPL* **1**, ICFP, 43:1–43:29.

Benedetti, E. D. & Ronchi Della Rocca, S. (2016). A type assignment for λ-calculus complete both for FPTIME and strong normalization. *Inf. Comput.* **248**, 195–214.

Bernadet, A. & Graham-Lengrand, S. (2013a). *A Big-Step Operational Semantics via Non-idempotent Intersection Types*. Technical report. http://www.lix.polytechnique.fr/~lengrand/Work/Reports/BigStep.pdf

Bernadet, A. & Graham-Lengrand, S. (2013b) Non-idempotent intersection types and strong normalisation. *Log. Methods Comput. Sci.* **9**(4), 1–46.

Bernadet, A. & Lengrand, S. (2011). Complexity of strongly normalising lambda-terms via non-idempotent intersection types. In FoSSaCS'11. Lecture Notes in Computer Science, Hofmann, M. (ed.), vol. 6604, Saarbrücken, Germany. Springer, pp. 88–107.

Blelloch, G. E. and Greiner, J. (1995) Parallelism in sequential functional languages. In FPCA'95, Williams, J. (ed.), La Jolla, California, USA. ACM, pp. 226–237.

Bucciarelli, A. & Ehrhard, T. (2001). On phase semantics and denotational semantics: The exponentials. *Ann. Pure Appl. Logic* **109**(3), 205–241.

Bucciarelli, A., Ehrhard, T. & Manzonetto, G. (2012). A relational semantics for parallelism and non-determinism in a functional setting. *Ann. Pure Appl. Log.* **163**(7), 918–934.

Bucciarelli, A., Kesner, D. and Ronchi Della Rocca, S. (2014) The inhabitation problem for non-idempotent intersection types. In IFIP-TCS'14. Lecture Notes in Computer Science, Díaz, J., Lanese, I. & Sangiorgi, D. (eds), vol. 8705, Rome, Italy. Springer, pp. 341–354.

Bucciarelli, A., Kesner, D. & Ventura, D. (2017) Non-idempotent intersection types for the Lambda-Calculus. *Log. J. IGPL* **25**(4), 431–464.

Carraro, A. & Guerrieri, G. (2014) A semantical and operational account of call-by-value solvability. In FoSSaCS 2014. Lecture Notes in Computer Science, Muscholl, A. (ed.), vol. 8412, Grenoble, France. Springer, pp. 103–118.

Coppo, M. and Dezani-Ciancaglini, M. (1978). A new type assignment for lambda-terms. *Arch. Math. Log.* **19**, 139–156.

Coppo, M. and Dezani-Ciancaglini, M. (1980). An extension of the basic functionality theory for the λ-calculus. *Notre Dame J. Formal Log.* **4**, 685–693.

Dal Lago, U. & Gaboardi, M. (2011). Linear dependent types and relative completeness. *Log. Methods Comput. Sci.* **8**(4). https://doi.org/10.2168/LMCS-8(4:11)2012

Dal Lago, U. & Petit, B. (2013) The geometry of types. In The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL'13, Rome, Italy - January 23–25, 2013, pp. 167–178. https://doi.org/10.1145/2429069.2429090

Dal Lago, U. & Petit, B. (2014). Linear dependent types in a call-by-value scenario. *Sci. Comput. Program.* **84**, 77–100. https://doi.org/10.1016/j.scico.2013.07.010

Danos, V. & Regnier, L. (2004) *Head Linear Reduction*. Technical report.

de Carvalho, D. (2007) *Sémantiques de la logique linéaire et temps de calcul*. Thèse de Doctorat. Université Aix-Marseille II.

de Carvalho, D. (2009). Execution Time of lambda-Terms via Denotational Semantics and Intersection Types. CoRR abs/0905.4251, pp. 1–36.

de Carvalho, D. (2016) The relational model is injective for multiplicative exponential linear logic. In CSL 2016. LIPIcs, Talbot, J.-M. & Regnier, L. (eds), vol. 62, Marseille, France. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 41:1–41:19.

de Carvalho, D. (2018) Execution time of λ-terms via denotational semantics and intersection types. *Math. Struct. Comput. Sci.* **28**(7), 1169–1203. https://doi.org/10.1017/S0960129516000396

de Carvalho, D., Pagani, M. & Tortora de Falco, L. (2011) A semantic measure of the execution time in linear logic. *Theor. Comput. Sci.* **412**(20), 1884–1902.

de Carvalho, D. & Tortora de Falco, L. (2016) A semantic account of strong normalization in linear logic. *Inf. Comput.* **248**, 104–129.

Díaz-Caro, A., Manzonetto, G. & Pagani, M. (2013) Call-by-value non-determinism in a linear logic type discipline. In LFCS 2013. Lecture Notes in Computer Science, Artëmov, S. N. & Nerode, A. (eds.), vol. 7734, San Diego, California, USA. Springer, pp. 164–178.

Dudenhefner, A. & Rehof, J. (2017) Intersection type calculi of bounded dimension. In POPL'17, Castagna, G. & Gordon, A. D. (eds.), Paris, France. ACM, pp. 653–665.

Ehrhard, T. (2012) Collapsing non-idempotent intersection types. In CSL'12. LIPIcs, Cégielski, P. & Durand, A. (eds.), vol. 16, Fontainebleau, France. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 259–273.

Ehrhard, T. & Guerrieri, G. (2016) The Bang Calculus: An untyped lambda-calculus generalizing call-by-name and call-by-value. In PPDP 2016, Edinburgh, UK. ACM, pp. 174–187.

Gardner, P. (1994) Discovering needed reductions using type theory. In TACS'94. Lecture Notes in Computer Science, Hagiya, M. and Mitchell, J. C. (eds), vol. 789, Sendai, Japan. Springer, pp. 555–574.

Girard, J.-Y. (1988) Normal functors, power series and the λ-calculus. *Ann. Pure Appl. Log.* **37**, 129–177.

Girard, J.-Y., Taylor, P. & Lafont, Y. (1989) *Proofs and Types*. New York, NY, USA: Cambridge University Press.

Grabmayer, C. (2018) Modeling terms by graphs with structure constraints (two illustrations). In Proceedings Tenth International Workshop on Computing with Terms and Graphs, TERMGRAPH@FSCD 2018, Oxford, UK, 7th July 2018, pp. 1–13. https://doi.org/10.4204/EPTCS.288.1

Guerrieri, G., Pellissier, L. and Tortora de Falco, L. (2016) Computing connected proof(-structure)s from their Taylor expansion. In FSCD 2016. LIPIcs, Kesner, D. & Pientka, B. (eds), vol. 52, Porto, Portugal. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 20:1–20:18.

Hoffmann, J., Aehlig, K. & Hofmann, M. (2012) Resource aware ML. In CAV 2012. Lecture Notes in Computer Science, Madhusudan, P. & Seshia, S. A. (eds), vol. 7358, Berkeley, CA, USA. Springer, pp. 781–786.

Hoffmann, J. & Hofmann, M. (2010) Amortized resource analysis with polynomial potential - a static inference of polynomial bounds for functional programs. In ESOP'10. Lecture Notes in Computer Science, Gordon, A. D. (ed.), vol. 6012, Paphos, Cyprus. Springer, pp. 287–306.

Hofmann, M. & Jost, S. (2003) Static prediction of heap space usage for first-order functional programs. In POPL 2003, Aiken, A. & Morrisett, G. (eds), New Orleans, Louisisana, USA. ACM, pp. 185–197.

Hughes, J., Pareto, L. & Sabry, A. (1996) Proving the correctness of reactive systems using sized types. In POPL'96, Boehm, H.-J. & Steele Jr., G. L. (eds), St. Petersburg Beach, Florida, USA. ACM, pp. 410–423.

Jost, S., Vasconcelos, P. B., Florido, M. and Hammond, K. (2017) Type-based cost analysis for lazy functional languages. *J. Autom. Reason.* **59**(1), 87–120.

Kesner, D. & Conchúir, S. (2008) *Milner's Lambda Calculus with Partial Substitutions*. Technical report. Paris 7 University. http://www.pps.univ-paris-diderot.fr/~kesner/papers/shortpartial.pdf

Kesner, D. & Ventura, D. (2014) Quantitative types for the linear substitution calculus. In IFIP-TCS'14. Lecture Notes in Computer Science, Díaz, J., Lanese, I. & Sangiorgi, D. (eds), vol. 8705, Rome, Italy. Springer, pp. 296–310.

Kesner, D. & Ventura, D. (2015) A resource aware computational interpretation for Herbelin's syntax. In ICTAC 2015. Lecture Notes in Computer Science, Leucker, M., Rueda, C. & Valencia, F. D. (eds), vol. 9399, Cali, Colombia. Springer, 388–403.

Kesner, D. & Vial, P. (2017) Types as resources for classical natural deduction. In FSCD 2017. LIPIcs, Miller, D. (ed.), vol. 84, Oxford, UK. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 24:1–24:17.

Kesner, D. & Vial, P. (2019) Non-idempotent types for classical calculi in natural deduction style. *Log. Methods Comput. Sci.* **16**(1). https://lmcs.episciences.org/6028

Kesner, D., Viso, A. & Ríos, A. (2018) Call-by-need, neededness and all that. In FoSSaCS'18. Lecture Notes in Computer Science, Baier, C. & Dal Lago, U. (eds.), vol. 10803, Thessaloniki, Greece. Springer, pp. 241–257.

Kfoury, A. J. (2000) A linearization of the Lambda-calculus and consequences. *J. Logic Comput.* **10**(3), 411–436.

Klop, J. W. (1980) *Combinatory Reduction Systems*. PhD thesis. Utrecht University.

Krivine, J.-L. (1993) *Lambda-Calculus, Types and Models*. Ellis Horwood.

Mascari, G. & Pedicini, M. (1994) Head linear reduction and pure proof net extraction. *Theor. Comput. Sci.* **135**(1), 111–137.

Mazza, D., Pellissier, L. & Vial, P. (2018) Polyadic approximations, fibrations and intersection types. *PACMPL* **2**(POPL), 6:1–6:28.

Milner, R. (2007) Local bigraphs and confluence: Two conjectures. *Electr. Notes Theor. Comput. Sci.* **175**(3), 65–73.

Neergaard, P. M. & Mairson, H. G. (2004) Types, potency, and idempotency: Why nonlinearity and amnesia make a type system work. In ICFP 2004, Okasaki, C. & Fisher, K. (eds), Snowbird, Utah, USA. ACM, pp. 138–149.

Ong, C.-H. L. (2017) Quantitative semantics of the lambda calculus: Some generalisations of the relational model. In LICS 2017, Ouaknine, J. (ed.), Reykjavik, Island. IEEE Computer Society, pp. 1–12.

Paolini, L., Piccolo, M. & Rocca, S. R. D. (2017) Essential and relational models. *Math. Struct. Comput. Sci.* **27**(5), 626–650.

Portillo, Á. J. R., Hammond, K., Loidl, H.-W. & Vasconcelos, P. B. (2002) Cost analysis using automatic size and time inference. In IFL 2002. Lecture Notes in Computer Science, Pena, R. & Arts, T. (eds), vol. 2670, Madrid, Spain. Springer, pp. 232–248.

Pottinger, G. (1980) A type assignment for the strongly normalizable λ-terms. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Seldin, J. P. & Hindley, J. R. (eds). Academic Press, pp. 561–578.

Simões, H. R., Hammond, K., Florido, M. & Vasconcelos, P. B. (2007) Using intersection types for cost-analysis of higher-order polymorphic functional programs. In TYPES 2006, Revised Selected Papers. Lecture Notes in Computer Science, Altenkirch, T. & McBride, C. (eds), vol. 4502, Nottingham, UK. Springer, pp. 221–236.

Urzyczyn, P. (1999) The emptiness problem for intersection types. *J. Symb. Logic* **64**(3), 1195–1215.

Urzyczyn, P. (2009) Inhabitation of low-rank intersection types. In TLCA 2009. Lecture Notes in Computer Science, Curien, P.-L. (ed.), vol. 5608. Springer, pp. 356–370.

van Raamsdonk, F., Severi, P., Sørensen, M. H. & Xi, H. (1999) Perpetual reductions in lambda-calculus. *Inf. Comput.* **149**(2), 173–225.

Vasconcelos, P. B. & Hammond, K. (2004) Inferring cost equations for recursive, polymorphic and higher-order functional programs. In IFL 2003, Revised Papers. Lecture Notes in Computer Science, vol. 3145, Edinburgh, UK. Springer, pp. 86–101.

Appendix A Head evaluation

A.1 Tight head correctness

Lemma A.1 (Multi-set decomposition for hd). *Let* $M = \uplus_{k\in K} M_k$. *Then* $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} t : M$ *if and only if there exist* $(\Phi_k)_{k\in K}$, $(\Gamma_k)_{k\in K}$, $(b_k)_{k\in K}$ *and* $(r_k)_{k\in K}$ *and such that* $\Phi_k \triangleright_{hd} \Gamma_k \vdash^{(b_k,r_k)} t : M_k$, *where* $\Gamma = \uplus_{k\in K}\Gamma_k$, $b = +_{k\in K} b_k$ *and* $r = +_{k\in K} r_k$. *Moreover,* $|\Phi|_{hd} = +_{k\in K}|\Phi_k|_{hd}$.

Proof By induction on the size of K. □

Lemma 3.5 (Substitution and typings for *hd*). *Let* $\Phi_t \triangleright_{hd} \Delta; x : M \vdash^{(b,r)} t : A$ *and* $\Phi_p \triangleright_{hd} \Gamma \vdash^{(b',r')} p : M$. *Then there exists a derivation* $\Phi_{t\{x\leftarrow p\}} \triangleright_{hd} \Gamma \uplus \Delta \vdash^{(b+b',r+r')} t\{x\leftarrow p\} : A$, *where* $|\Phi_{t\{x\leftarrow p\}}| = |\Phi_t| + |\Phi_p| - |M|$.

Proof Let Φ_p (resp. Φ_t) be the typing derivation of $\Gamma \vdash^{(b',r')} p : M$ (resp. $\Delta; x : M \vdash^{(b,r)} t : A$) in system *hd*. We prove that there exists a typing $\Phi_{t\{x\leftarrow p\}} \triangleright_{hd} \Gamma \uplus \Delta \vdash^{(b+b',r+r')} t\{x\leftarrow p\} : A$. The proof is by induction on Φ_t. Let us write M as $[B_i]_{i\in I}$ for some (potentially empty) set of indices I. We reason by cases of the last rule of Φ_t:

- *Rule* ax. Two cases:

  1. $t = x$, and so $t\{x{\leftarrow}p\} = x\{x{\leftarrow}p\} = p$ and $\Phi_t \rhd_{hd}\ x : [A] \vdash^{(0,0)} x : A$. Thus, $|I| = 1$ and $M = [A]$, and the hypothesis $\Phi_p \rhd_{hd}\ \Gamma \vdash^{(b',r')} p : [A]$ is necessarily obtained by applying a unary many rule to a derivation of the form $\Psi_p \rhd_{hd}\ \Gamma \vdash^{(b',r')} p : A$. Given that $x\{x{\leftarrow}p\} = p$, $r + r' = 0 + r' = r'$, and $b + b' = b'$, the typing derivation $\Phi_{t\{x\leftarrow p\}} := \Psi_p$ satisfies the requirements.
  2. $t = y$, and so $M = [\,]$, $b' = r' = 0$ and $t\{x{\leftarrow}p\} = y\{x{\leftarrow}p\} = y$. Then it is enough to take $\Phi_{t\{x\leftarrow p\}} := \Phi_t$.

- *Rule* fun_b. Then $t = \lambda y.u$, and Φ_t is such that $b = b_u + 1$, and it has the following form:

$$\frac{\Phi_u \rhd_{hd}\ \Delta; x : M; y : N \vdash^{(b_u,r)} u : B}{\Delta; x : M \vdash^{(b_u+1,r)} \lambda y.u : N \to B}\ \mathsf{fun}_b$$

  By *i.h.* there exists $\Phi_{u\{x\leftarrow p\}}$ such that

$$\Phi_{u\{x\leftarrow p\}} \rhd_{hd}\ \Gamma \uplus \Delta; y : N \vdash^{(b_u+b',r+r')} u\{x{\leftarrow}p\} : B$$

  from which by applying fun_b we obtain:

$$\Phi_{t\{x\leftarrow p\}} \rhd_{hd}\ \Gamma \uplus \Delta \vdash^{(b_u+b'+1,r+r')} \lambda y.u\{x{\leftarrow}p\} : N \to B$$

  The derivation $\Phi_{t\{x\leftarrow p\}}$ satisfies the requirements because $b_u + b' + 1 = b + b'$.
- *Rule* fun_r. Then $t = \lambda y.u$, and Φ_t is such that $r = r'' + 1$ and it has the following form:

$$\frac{\Phi_u \rhd_{hd}\ \Delta; x : M; y : \mathtt{Tight} \vdash^{(b,r'')} u : \mathtt{tight}}{\Delta; x : M \vdash^{(b,r''+1)} \lambda y.u : \mathtt{abs}}\ \mathsf{fun}_b$$

  By *i.h.* there exists $\Phi_{u\{x\leftarrow p\}}$ such that

$$\Phi_{u\{x\leftarrow p\}} \rhd_{hd}\ \Gamma \uplus \Delta; y : \mathtt{Tight} \vdash^{(b+b',r''+r')} u\{x{\leftarrow}p\} : \mathtt{tight}$$

  from which by applying fun_r we obtain

$$\Phi_{t\{x\leftarrow p\}} \rhd_{hd}\ \Gamma \uplus \Delta \vdash^{(b+b',r''+r'+1)} \lambda y.u\{x{\leftarrow}p\} : \mathtt{abs}$$

  that satisfies the requirements because $r'' + r' + 1 = r + r'$.
- *Rule* app_b. Then $t = uq$. The left premiss of the app_b rule in Φ_t assigns a type $u : N \to A$ and the right premiss is a many rule with $k := |N|$ premisses. The multi-set M assigned to x can be partioned in $k + 1$ (potentially empty) multi-sets $M_1, \ldots, M_k$ and M_u, to be distributed among the premisses of the app_b rule of Φ as follows (if k=0 then the many rule has no premisses):

$$\frac{\Phi_u \rhd_{hd}\ \Delta_u; x : M_u \vdash^{(b'',r'')} u : N \to A \qquad \dfrac{(\Phi_q^j \rhd_{hd}\ \Delta_q^j; x : M_j \vdash^{(b_j,r_j)} q : C_j)_{j=1,\ldots,k}}{\Phi_q \rhd_{hd}\ \Delta_q; x : M^\circ \vdash^{(b^\circ,r^\circ)} q : N}\ \mathsf{many}}{\Delta_u \uplus \Delta_q; x : M \vdash^{(b''+b^\circ,r''+r^\circ)} uq : A}\ \mathsf{app}_b$$

  where the notations satisfy the following:

$$b^\circ = +_{j=1\ldots k} b_j,\ r^\circ = +_{j=1\ldots k} r_j,\ \Delta_q = \uplus_{j=1}^k \Delta_q^j, \text{ and } M^\circ = \uplus_{j=1}^k M_j,$$

$\Delta = \Delta_u \uplus \Delta_q$,

$b = b'' + b^\circ$, and

$r = r'' + r^\circ$.

Moreover, given the partition of M into $M_1, \ldots, M_k, M_u$, the derivation $\Phi_p \triangleright_{hd} \Gamma \vdash^{(b',r')} p : M$ of the first hypothesis gives rise, by Lemma A.1 to derivations $\Psi_u \triangleright_{hd} \Gamma_u \vdash^{(b_u,r_u)} p : M_u$ and $(\Psi_q^j \triangleright_{hd} \Gamma_j \vdash^{(b'_j,r'_j)} p : M_j)_{j=1,\ldots,k}$ with

$\Gamma = \Gamma_u \uplus_{j=1\ldots k} \Gamma_j$,

$b' = b_u +_{j=1\ldots k} b'_j$, and

$r' = r_u +_{j=1\ldots k} r'_j$.

Now, by *i.h.* we can substitute these derivations Ψ_u and Ψ_q^j into the premisses of the app_b rule, obtaining the derivations $\Phi_{u\{x\leftarrow p\}}$, $\Phi^j_{q\{x\leftarrow p\}}$, and $\Upsilon_{q\{x\leftarrow p\}}$ such that

$$\Phi_{u\{x\leftarrow p\}} \triangleright_{hd} \Delta_u \uplus \Gamma_u \vdash^{(b''+b_u, r''+r_u)} u\{x\leftarrow p\} : N \to A$$

$$\frac{(\Phi^j_{q\{x\leftarrow p\}} \triangleright_{hd} \Delta^j_q \uplus \Gamma_j \vdash^{(b_j+b'_j, r_j+r'_j)} q\{x\leftarrow p\} : C_j)_{j=1,\ldots,k}}{\Upsilon_{q\{x\leftarrow p\}} \triangleright_{hd} \Delta_q \uplus \Gamma_q \vdash^{(+_{j=1\ldots k}(b_j+b'_j), +_{j=1\ldots k}(r_j+r'_j))} q\{x\leftarrow p\} : [C_j]_{j=1,\ldots,k}} \;\mathsf{many}$$

where Γ_q stands for $\uplus_{j=1}^k \Gamma_j$. By applying app_b we obtain

$$\Psi \triangleright_{hd} \Delta_u \uplus \Delta_q \uplus \Gamma_u \uplus \Gamma_q \vdash^{(b^*, r^*)} u\{x\leftarrow p\}q\{x\leftarrow p\} = (uq)\{x\leftarrow p\} : A$$

We conclude since

- $\Delta_u \uplus \Delta_q \uplus \Gamma_u \uplus \Gamma_q = \Delta \uplus \Gamma$.
- The resulting first counter is as required:
$$\begin{array}{rcl} b^* & = & b'' + b_u +_{j=1\ldots k} (b_j + b'_j) \\ & = & b'' +_{j=1\ldots k} b_j + b_u +_{j=\ldots k} b'_j \\ & = & b'' + b^\circ + b' \\ & = & b + b' \end{array}$$
- The resulting second counter is as required:
$$\begin{array}{rcl} r^* & = & r'' + r_u +_{j=1\ldots k} (r_j + r'_j) \\ & = & r'' +_{j=1\ldots k} r_j + r_u +_{j=1\ldots k} r'_j \\ & = & r + r' \end{array}$$

- *Rule* app^{hd}_r. Then $t = uq$ and Φ_t is such that $r = r_u + 1$ and it has the following form:

$$\frac{\Phi_u \triangleright_{hd} \Delta; x : M \vdash^{(b,r_u)} u : \mathtt{neutral}}{\Delta; x : M \vdash^{(b,r_u+1)} uq : \mathtt{neutral}} \;\mathsf{app}^{hd}_r$$

By *i.h.* we can substitute Φ_p into Φ_u obtaining $\Phi_{u\{x\leftarrow p\}}$ such that

$$\Upsilon \triangleright_{hd} \Gamma \uplus \Delta \vdash^{(b+b', r_u+r')} u\{x\leftarrow p\} : \mathtt{neutral}$$

By applying app^{hd}_r we obtain

$$\Theta \triangleright_{hd} \Gamma \uplus \Delta \vdash^{(b+b', r_u+r'+1)} u\{x\leftarrow p\}q\{x\leftarrow p\} = (uq)\{x\leftarrow p\} : \mathtt{neutral}$$

that satisfies the requirements because $r_u + r' + 1 = r + r'$. □

A.2 Tight head completeness

Proposition 3.8 (Normal forms are tightly typable for hd)**.** *Let t be such that* $\mathtt{normal}_{hd}(t)$. *Then*

1. Existence*: there exists a tight derivation* $\Phi \triangleright_{hd} \Gamma \vdash^{(0,|t|_{hd})} t : A$.
2. Structure*: moreover, if* $\mathtt{neutral}_{hd}(t)$ *then* $A = \mathtt{neutral}$*, and if* $\mathtt{abs}_{hd}(t)$ *then* $A = \mathtt{abs}$.

Proof

By induction on $\mathtt{normal}_{hd}(t)$. Cases:

1. *Variable*, i.e. $t = x$. Then the following derivation evidently satisfies all points of the statement:
$$\dfrac{}{x : [\mathtt{neutral}] \vdash^{(0,0)} x : \mathtt{neutral}}\ \mathsf{ax}$$
2. *Abstraction*, i.e. $t = \lambda y.p$ with $\mathtt{normal}_{hd}(p)$. By *i.h.* there is a tight derivation $\Phi_p \triangleright_{hd} \Delta \vdash^{(0,|p|_{hd})} p : \mathtt{tight}$. Since the derivation Φ_p is tight, the typing context Δ has the shape $\Gamma; y : \mathtt{Tight}$ (potentially, $y : [\,]$). Then the following is a tight derivation for $\lambda y.p$:
$$\dfrac{\Phi_p \triangleright_{hd} \Gamma; y : \mathtt{Tight} \vdash^{(0,|p|_{hd})} p : \mathtt{tight}}{\Gamma \vdash^{(0,|p|_{hd}+1)} \lambda y.p : \mathtt{abs}}\ \mathsf{fun}_r$$
Moreover, t is not neutral so the part about neutral terms is trivially true, while it is an abstraction and it is indeed typed with $\mathtt{abs}$.
3. *Application*, i.e. $t = pu$ and $\mathtt{normal}_{hd}(t)$ implies $\mathtt{neutral}_{hd}(t)$, that implies $\mathtt{neutral}_{hd}(p)$, that implies $\mathtt{normal}_{hd}(p)$. By *i.h.*, there is a tight derivation $\Psi \triangleright_{hd} \Gamma \vdash^{(0,|p|_{hd})} p : \mathtt{tight}$ typing p with $\mathtt{neutral}$. Then the following is a tight derivation Φ types $t = pu$ with $\mathtt{neutral}$, and having as second index satisfies $|p|_{hd} + 1 = |pu|_{hd} = |t|_{hd}$, as required:
$$\dfrac{\Psi \triangleright_{hd} \Gamma \vdash^{(0,|p|_{hd})} p : \mathtt{neutral}}{\Gamma \vdash^{(0,|p|_{hd}+1)} pu : \mathtt{neutral}}\ \mathsf{app}_r^{hd}$$
Moreover, $\mathtt{neutral}_{hd}(t)$ and Φ does indeed type t with $\mathtt{neutral}$. Dually, t is not an abstraction and so that point trivially holds. □

Lemma 3.9 (Anti-substitution and typings for hd)**.** *Let* $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} t\{x{\leftarrow}p\} : A$. *Then there exist*

- *a multi-set M;*
- *a typing derivation* $\Phi_t \triangleright_{hd} \Gamma_t; x : M \vdash^{(b_t,r_t)} t : A$*; and*
- *a typing derivation* $\Phi_p \triangleright_{hd} \Gamma_p \vdash^{(b_p,r_p)} p : M$

such that

- Typing context*:* $\Gamma = \Gamma_t \uplus \Gamma_p$*;*

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

- Indices: $(b,r) = (b_t + b_p, r_t + r_p)$;
- Size: $|\Phi| = |\Phi_t| + |\Phi_p| - |M|$.

Proof By induction on t. Cases:

- *Variable*, i.e. $t = y$. Two subcases, depending on the identity of y:

  1. $x = y$. Then $t\{x{\leftarrow}p\} = x\{x{\leftarrow}p\} = p$, so that $\Phi \triangleright_{hd} \Gamma \vdash^{(b,r)} p : A$. There is only one possibility: $|M| = 1$, Φ_p is

  $$\frac{\Phi \triangleright_{hd} \ \Gamma \vdash^{(b,r)} p : A}{\Gamma \vdash^{(b,r)} p : [A]}\ \mathsf{many}$$

  and Φ_t is

  $$\frac{}{x : [A] \vdash^{(0,0)} x : A}\ \mathsf{ax}$$

  2. $x \neq y$. Then $t\{x{\leftarrow}p\} = y\{x{\leftarrow}p\} = y$. There is only one possibility: $|M| = 0$, Φ_t is exactly Φ, that is,

  $$\frac{}{y : [A] \vdash^{(0,0)} y : A}\ \mathsf{ax}$$

  and Φ_p is

  $$\frac{}{\vdash^{(0,0)} p : [\,]}\ \mathsf{many}$$

- *Abstraction*, i.e. $t = \lambda y.u$. Then $t\{x{\leftarrow}p\} = \lambda y.u\{x{\leftarrow}p\}$. Two sub-cases, depending on the last rule of Φ:

  1. *Rule* fun_b. Then Φ has the following form:

  $$\frac{\Phi_{u\{x\leftarrow p\}} \triangleright_{hd} \ \Gamma; y : N \vdash^{(b_{u\{x\leftarrow p\}}, r)} u\{x{\leftarrow}p\} : D}{\Gamma \vdash^{(b_{u\{x\leftarrow p\}}+1, r)} \lambda y.u\{x{\leftarrow}p\} : N \to D}\ \mathsf{fun}_b$$

  with $b = b_{u\{x\leftarrow p\}} + 1$. By *i.h.* there exist M and typing derivations

  $$\Phi_u \triangleright_{hd} \ \Delta_u; y : N; x : M \vdash^{(b_u, r_u)} u : A \qquad \Phi_p \triangleright_{hd} \ \Delta_p \vdash^{(b_p, r_p)} p : M$$

  such that

  - *Typing context*: $(\Gamma; y : N) = (\Delta_u; y : N \uplus \Delta_p)$;
  - *Indices*: $(b_{u\{x\leftarrow p\}}, r) = (b_u + b_p, r_u + r_p)$.

  Then the derivation Φ_t defined as

  $$\frac{\Phi_u \triangleright_{hd} \ \Gamma; y : N; x : M \vdash^{(b_u, r_u)} u : D}{\Gamma; x : M \vdash^{(b_u+1, r_u)} \lambda y.u : N \to D}\ \mathsf{fun}_b$$

  satisfies the statement with respect to $b_t := b_u + 1$ and $r_t := r_u$ because

  - *Typing context*: the *i.h.* implies $\Gamma = (\Delta_u \uplus \Delta_p)$;
  - *Indices*:

    a. $b_t + b_p = b_u + 1 + b_p =_{i.h.} b_{u\{x\leftarrow p\}} + 1 = b$,

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

b. $r_t + r_p = r_u + r_p =_{i.h.} r_{u\{x\leftarrow p\}} = r$.

2. *Rule* fun_r. Then Φ has the following form:

$$\frac{\Phi_{u\{x\leftarrow p\}} \rhd_{hd}\ \Gamma; y : \mathtt{Tight} \vdash^{(b, r_{u\{x\leftarrow p\}})} u\{x\leftarrow p\} : \mathtt{tight}}{\Gamma \vdash^{(b, r_{u\{x\leftarrow p\}}+1)} \lambda y.u\{x\leftarrow p\} : \mathtt{abs}}\ \mathsf{fun}_r$$

with $r = r_{u\{x\leftarrow p\}} + 1$. By *i.h.* there exist M and typing derivations

$$\Phi_u \rhd \Delta_u; y : \mathtt{Tight}; x : M \vdash^{(b_u, r_u)} u : \mathtt{tight} \qquad \Phi_p \rhd \Delta_p \vdash^{(b_p, r_p)} p : M$$

such that

- *Typing context*: $(\Gamma; y : \mathtt{Tight}) = (\Delta_u; y : \mathtt{Tight} \uplus \Delta_p)$;
- *Indices*: $(b, r_{u\{x\leftarrow p\}}) = (b_u + b_p, r_u + r_p)$.

Then the derivation Φ_t defined as

$$\frac{\Phi_u \rhd_{hd}\ \Gamma; y : \mathtt{Tight}; x : M \vdash^{(b_u, r_u)} u : \mathtt{tight}}{\Gamma; x : M \vdash^{(b_u, r_u+1)} \lambda y.u : \mathtt{abs}}\ \mathsf{fun}_b$$

satisfies the statement with respect to $r_t := r_u + 1$ because

- *Typing context*: the *i.h.* implies $\Gamma = (\Delta_u \uplus \Delta_p)$
- *Indices*:

a. $b_u + b_p =_{i.h.} b$,

b. $r_t + r_p = r_u + 1 + r_p =_{i.h.} r_{u\{x\leftarrow p\}} + 1 = r$.

- *Application*, i.e. $t = uq$. Then $t\{x\leftarrow p\} = u\{x\leftarrow p\}q\{x\leftarrow p\}$. Two sub-cases, depending on the last rule of Φ:

1. *Rule* app_b. Then Φ has the following form:

$$\frac{\Phi_{u\{x\leftarrow p\}} \rhd_{hd}\ \Gamma_1 \vdash^{(b_1, r_1)} u\{x\leftarrow p\} : M \to A \qquad \Phi_{q\{x\leftarrow p\}} \rhd_{hd}\ \Gamma_2 \vdash^{(b_2, r_2)} q\{x\leftarrow p\} : M}{\Gamma_1 \uplus \Gamma_2 \vdash^{(b_1+b_2, r_1+r_2)} u\{x\leftarrow p\}q\{x\leftarrow p\} : A}\ \mathsf{app}_b$$

with $\Gamma = \Gamma_1 \uplus \Gamma_2$, $b = b_1 + b_2$, and $r = r_1 + r_2$.

By *i.h.* applied to $u\{x\leftarrow p\}$ and $q\{x\leftarrow p\}$, there exist (disjoint) finite sets M_u and M_q and typing derivations:

$$\Phi_u \rhd_{hd}\ \Delta_u; x : M_u \vdash^{(b_u, r_u)} u : M \to A$$

$$\Phi_q \rhd_{hd}\ \Delta_q; x : M_q \vdash^{(b_q, r_q)} q : M$$

$$\Phi_p^u \rhd_{hd}\ \Pi_u \vdash^{(b_p^u, r_p^u)} p : M_u$$

$$\Phi_p^q \rhd_{hd}\ \Pi_q \vdash^{(b_p^q, r_p^q)} p : M_q$$

such that

- *Typing context*: $\Gamma_1 = \Delta_u \uplus \Pi_u$ and $\Gamma_2 = \Delta_q \uplus \Pi_q$.
- *Indices*: $(b_1, r_1) = (b_u + b_p^u, r_u + r_p^u)$ and $(b_2, r_2) = (b_q + b_p^q, r_q + r_p^q)$.

The derivations Φ_p^u and Φ_p^q can be summed (by inverting their many final rule and reapplying a many rule to the union of the premisses) obtaining a derivation

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$\Phi_p \rhd_{hd} \ \Pi \vdash^{(b_p, r_p)} p : M$, where $\Pi = \Pi_u \uplus \Pi_q$ and $b_p = b_p^u + b_p^q$ and $r_p = r_p^u + r_p^q$ and $M = M_u + M_q$. We then apply app_b to obtain the following derivation Φ_t:

$$\dfrac{\Phi_u \rhd_{hd} \ \Delta_u; x : M_u \vdash^{(b_u, r_u)} u : M \to A \qquad \Phi_q \rhd_{hd} \ \Delta_q; x : M_q \vdash^{(b_q, r_q)} q : M}{\Delta_u \uplus \Delta_q; x : M_u + M_q \vdash^{(b_u + b_q, r_u + r_q)} uq : \mathtt{tight}} \, \mathsf{app}_b$$

We let $\Delta := \Delta_u \uplus \Delta_q$, $b_t := b_u + b_q$ and $r_t := r_u + r_q$ and then observe that we obtained the statement, because of the following equalities:

a. *Typing context*: $\Gamma = \Gamma_1 \uplus \Gamma_2 = \Delta_u \uplus \Pi_u \uplus \Delta_q \uplus \Pi_q = \Delta \uplus \Pi$.
b. *Indices*: $(b, r) = (b_1 + b_2, r_1 + r_2) = (b_t + b_p, r_t + r_p)$.

2. *Rule* app_r^{hd}. Let $t = uq$ so that $t\{x \leftarrow p\} = u\{x \leftarrow p\} q\{x \leftarrow p\}$. Then Φ has the following form:

$$\dfrac{\Phi_{u\{x \leftarrow p\}} \rhd_{hd} \ \Gamma \vdash^{(b, r')} u\{x \leftarrow p\} : \mathtt{neutral}}{\Gamma \vdash^{(b, r'+1)} u\{x \leftarrow p\} q\{x \leftarrow p\} : \mathtt{neutral}} \, \mathsf{app}_r^{hd}$$

with $r = r' + 1$.

By *i.h.* applied to $u\{x \leftarrow p\}$ there exists M and typing derivations:

$$\Phi_u \rhd_{hd} \ \Gamma_u; x : M \vdash^{(b_u, r_u)} u : \mathtt{neutral}$$

$$\Phi_p \rhd_{hd} \ \Gamma_p \vdash^{(b_p, r_p)} p : M$$

such that

– *Typing context*: $\Gamma = \Gamma_u \uplus \Gamma_p$.
– *Indices*: $(b, r') = (b_u + b_p, r_u + r_p)$.

We then apply app_r^{hd} to obtain the following derivation Φ_t:

$$\dfrac{\Phi_u \rhd_{lo} \ \Gamma_u; x : M \vdash^{(b_u, r_u)} u : \mathtt{neutral}}{\Gamma_u; x : M \vdash^{(b_u, r_u + 1)} uq : \mathtt{neutral}} \, \mathsf{app}_r^{hd}$$

We let $\Gamma_t := \Gamma_u$, $b_t := b_u$ and $r_t := r_u + 1$ and then observe that we obtained the statement, because of the following equalities:

a. *Typing context*: $\Gamma = \Gamma_t \uplus \Gamma_p$.
b. *Indices*: $(b, r) = (b, r' + 1) = (b_u + b_p, r_u + r_p) = (b_t + b_p, r_t + r_p)$. □

Appendix B Leftmost evaluation

B.1 Tight leftmost correctness

Lemma B.1 (Multi-set decomposition for lo)**.** *Let* $M = \uplus_{k \in K} M_k$. *Then* $\Phi \rhd_{lo} \ \Gamma \vdash^{(b,r)} t : M$ *if and only if there exist* $(\Phi_k)_{k \in K}$, $(\Gamma_k)_{k \in K}$, $(b_k)_{k \in K}$ *and* $(r_k)_{k \in K}$ *and such that* $\Phi_k \rhd_{lo} \ \Gamma_k \vdash^{(b_k, r_k)} t : M_k$ *for all* $k \in K$, *where* $\Gamma = \uplus_{k \in K} \Gamma_k$, $b = +_{k \in K} b_k$ *and* $r = +_{k \in K} r_k$. *Moreover,* $|\Phi|_{lo} = +_{k \in K} |\Phi_k|_{lo}$.

Proof By induction on the size of K. □

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proposition 4.4 (Properties of lo typings for normal forms)**.** *Let t be such that $\mathtt{normal}_{lo}(t)$, and $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$ be a tight type derivation. Then*

1. Tight indices*: $b = 0$ and $r = |t|_{lo}$. As a consequence $|t|_{lo} \leq |\Phi|$.*
2. Neutrality*: If $A = \mathtt{neutral}$ then $\mathtt{neutral}_{lo}(t)$.*

Proof By induction on t. Note that $\mathtt{neutral}_{lo}$ implies $\mathtt{normal}_{lo}$ and so we can apply the *i.h.* when $\mathtt{neutral}_{lo}$ holds on some subterm of t. If $\mathtt{normal}_{lo}(t)$ because $\mathtt{neutral}_{lo}(t)$ there are two cases:

- *Variable*, i.e. $t = x$. Then Φ has the following form and evidently verifies all the points of the statement:

$$\frac{}{x : [A] \vdash^{(0,0)} x : A}\ \mathsf{ax}$$

- *Application*, i.e. $t = pu$, $\mathtt{neutral}_{lo}(p)$ and $\mathtt{normal}_{lo}(u)$. Cases of the last rule of Φ:

  – app_b *rule*: this case is excluded by Lemma 4.3.
  – app_r^{lo} *rule*:

$$\frac{\Phi_p \rhd_{lo} \Gamma_p \vdash^{(b_p,r_p)} p : \mathtt{neutral} \quad \Phi_u \rhd_{lo} \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{tight}}{\Gamma_p \uplus \Gamma_u \vdash^{(b_p+b_u,r_p+r_u+1)} pu : \mathtt{neutral}}\ \mathsf{app}_r^{lo}$$

  with $b = b_p + b_u$, $r = r_p + r_u + 1$, and $\Gamma = \Gamma_p \uplus \Gamma_u$.

  1. *Tight counters*: if Φ is tight, then Φ_p and Φ_u are tight and by *i.h.* $r_p = |p|_{lo}$ and $b_p = 0$, and $r_u = |u|_{lo}$ and $b_u = 0$. Then, $r = r_p + r_u + 1 =_{i.h.} |p|_{lo} + |u|_{lo} + 1 = |pu|_{lo} = |t|_{lo}$ and $b = b_p + b_u = 0 + 0 = 0$.
  2. *Neutrality*: $\mathtt{neutral}_{lo}(t)$ holds by hypothesis.

Now, there is only one case left for $\mathtt{normal}_{lo}(t)$:

- *Abstraction*, i.e. $t = \lambda x.p$ and $\mathtt{normal}_{lo}(t)$ because $\mathtt{normal}_{lo}(p)$. Cases of the last rule of Φ:

  – fun_b *rule*:

$$\frac{\Phi_p \rhd_{lo} \Gamma; x : M \vdash^{(b_p,r)} p : A}{\Gamma \vdash^{(b_p+1,r)} \lambda x.p : M \to A}\ \mathsf{fun}_b$$

  with $b = b_p + 1$. Φ is not tight, so the statement trivially holds.
  – fun_r *rule*:

$$\frac{\Phi_p \rhd \Gamma; x : \mathtt{Tight} \vdash^{(b,r_p)} p : \mathtt{tight}}{\Gamma \vdash^{(b,r_p+1)} \lambda x.p : \mathsf{abs}}\ \mathsf{fun}_r$$

  with $r = r_p + 1$.

  1. *Tight counters*: if Φ is tight, then Φ_p is tight and by *i.h.* and $r_p = |p|_{lo}$ and $b = 0$. Then, $r = r_p + 1 =_{i.h.} |p|_{lo} + 1 = |t|_{lo}$.
  2. *Neutrality*: $A \neq \mathtt{neutral}$, so the statement trivially holds. □

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Lemma 4.5 (Substitution and typings for lo). *Let* $\Phi_t \rhd_{lo} \ \Delta; x : M \vdash^{(b,r)} t : A$ *and* $\Phi_p \rhd_{lo} \ \Gamma \vdash^{(b',r')} p : M$. *Then there exists a derivation* $\Phi_{t\{x \leftarrow p\}} \rhd_{lo} \ \Gamma \uplus \Delta \vdash^{(b+b',r+r')} t\{x \leftarrow p\} : A$ *where* $|\Phi_{t\{x \leftarrow p\}}| = |\Phi_t| + |\Phi_p| - |M|$. *Moreover, if* Φ_t *and* Φ_p *are traditional, then* $\Phi_{t\{x \leftarrow p\}}$ *is traditional too.*

Proof Let $\Phi_t \rhd_{lo} \ \Delta; x : M \vdash^{(b,r)} t : A$ and $\Phi_p \rhd_{lo} \ \Gamma \vdash^{(b',r')} p : M$. We prove that there exists a derivation $\Phi_{t\{x \leftarrow p\}} \rhd_S \ \Gamma \uplus \Delta \vdash^{(b+b',r+r')} t\{x \leftarrow p\} : A$ by induction on Φ_t. System lo differs from hd only for because it replaces rule app_r^{hd} with app_r^{lo}. Then the proof for all cases but app_r^{lo} is like the one for system hd (Lemma 3.5). We only treat here the case of app_r^{lo}.

- *Rule* app_r^{lo}. Now, $t = uq$ and M splits into two multi-sets M_u and M_q so that Φ has the following form:

$$\frac{\Phi_u \rhd_{lo} \ \Delta_u; x : M_u \vdash^{(b_u,r_u)} u : \mathtt{neutral} \qquad \Phi_q \rhd_{lo} \ \Delta_q; x : M_q \vdash^{(b_q,r_q)} q : \mathtt{tight}}{\Delta_u \uplus \Delta_q; x : M \vdash^{(b_u+b_q,r_u+r_q+1)} uq : \mathtt{neutral}} \ \mathsf{app}_r^{lo}$$

with

$$b = b_u + b_q,$$
$$r = r_u + r_q + 1, \text{ and}$$
$$\Delta = \Delta_u \uplus \Delta_q.$$

Since $M = M_u \uplus M_q$, Lemma B.1 gives two derivations $\Psi_u \rhd_{lo} \ \Gamma_u \vdash^{(b'_u,r'_u)} p : M_u$ and $\Psi_q \rhd_{lo} \ \Gamma_q \vdash^{(b'_q,r'_q)} p : M_q$ such that $\Gamma = \Gamma_u \uplus \Gamma_q$, $b' = b'_u + b'_q$, and $r' = r'_u + r'_q$.
By *i.h.* there exist Θ_u and Θ_q such that

$$\Theta_u \rhd_{lo} \ \Delta_u \uplus \Gamma_u \vdash^{(b_u+b'_u,r_u+r'_u)} u\{x \leftarrow p\} : \mathtt{neutral}$$

$$\Theta_q \rhd_{lo} \ \Delta_q \uplus \Gamma_q \vdash^{(b_q+b'_q,r_q+r'_q)} q\{x \leftarrow p\} : \mathtt{tight}$$

with $|\Theta_u| = |\Phi_u| + |\Psi_u| - |M_u|$ and $|\Theta_q| = |\Phi_q| + |\Psi_q| - |M_q|$. Then by applying app_r^{lo} we obtain

$$\Phi_{t\{x \leftarrow p\}} \rhd_{lo} \ \Gamma \uplus \Delta \vdash^{(b^*,r^*)} u\{x \leftarrow p\}q\{x \leftarrow p\} = (uq)\{x \leftarrow p\} : \mathtt{neutral}$$

where

- $b^* = b_u + b'_u + b_q + b'_q = b + b'$,
- $r^* = r_u + r'_u + r_q + r'_q + 1 = r + r'$, and
- $|\Phi_{t\{x \leftarrow p\}}| = |\Theta_u| + |\Theta_q| + 1 = |\Phi_u| + |\Psi_u| - |M_u| + |\Phi_q| + |\Psi_q| - |M_q| + 1 = |\Phi_t| - |\Phi_p| - |M|$. □

Theorem 4.7 (Tight correctness for lo). *Let* $\Phi \rhd_{lo} \ \Gamma \vdash^{(b,r)} t : A$ *be a tight derivation. Then there exists* p *such that* $t \rightarrow_{lo}^{b} p$, $\mathtt{normal}_{lo}(p)$, *and* $|p|_{lo} = r$. *Moreover, if* $A = \mathtt{neutral}$ *then* $\mathtt{neutral}_{lo}(p)$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proof By induction on $|\Phi|$. If t is a $\to_{lo}$ normal form – that covers the base case $|\Phi| = 1$, for which t is necessarily a variable – then by taking $p := t$ and $k := 0$ the statement follows from the *tightness* property of tight typings of normal forms (Proposition 4.4.1) – the *moreover* part follows from the *neutrality* property (Proposition 4.4.2). Otherwise, $t \to_{lo} u$ and by quantitative subject reduction (Proposition 4.6) there is a derivation $\Psi \rhd_{lo} \Gamma \vdash^{(b-1,r)} u : A$ such that $|\Psi| < |\Phi|$. By *i.h.*, there exists p such that $\mathtt{normal}_{lo}(p)$ and $u \to_{lo}^{b-1} p$ and $|p|_{lo} = r$. Just note that $t \to_{lo} u \to_{lo}^{b-1} p$, that is, $t \to_{lo}^{b} p$. $\square$

B.2 Tight leftmost completeness

Proposition 4.8 (Normal forms are tightly typable for lo)**.** *Let t be such that* $\mathtt{normal}_{lo}(t)$*. Then*

1. Existence*: there exists a tight derivation* $\Phi \rhd_{lo} \Gamma \vdash^{(0,|t|_{lo})} t : A$*.*
2. Structure*: moreover, if* $\mathtt{neutral}_{lo}(t)$ *then* $A = \mathtt{neutral}$*, and if* $\mathtt{abs}_{lo}(t)$ *then* $A = \mathtt{abs}$.
3. Unique size*: if Ψ is another tight derivation for t then* $|\Phi| = |\Psi|$.

Proof

By induction on $\mathtt{normal}_{lo}(t)$. Cases:

1. *Variable*, i.e. $t = x$. Then the following derivation evidently satisfies the first two points of the statement:

$$\frac{}{x : [\mathtt{neutral}] \vdash^{(0,0)} x : \mathtt{neutral}}\ \mathsf{ax}$$

   The only other possible tight derivation for x is

$$\frac{}{x : [\mathtt{abs}] \vdash^{(0,0)} x : \mathtt{abs}}\ \mathsf{ax}$$

   that has the same size.

2. *Abstraction*, i.e. $t = \lambda y.p$ with $\mathtt{normal}_{lo}(p)$. By *i.h.* there is a tight derivation $\Phi_p \rhd_{lo} \Delta \vdash^{(0,|p|_{lo})} p : \mathtt{tight}$.

   a. *Existence*: since the derivation Φ_p is tight, the typing context Δ has the shape $\Gamma; y : \mathtt{Tight}$ (potentially, $y : [\,]$). Then the following is a tight derivation for $\lambda y.p$:

$$\frac{\Phi_p \rhd_{lo} \Gamma; y : \mathtt{Tight} \vdash^{(0,|p|_{lo})} p : \mathtt{tight}}{\Gamma \vdash^{(0,|p|_{lo}+1)} \lambda y.p : \mathtt{abs}}\ \mathsf{fun}_r$$

   b. *Structure*: moreover, t is not neutral so the part about neutral terms is trivially true, while it is an abstraction and it is indeed typed with $\mathtt{abs}$.

   c. *Unique size*: by *i.h.* all tight derivations for p have the same size. The statement follows by the evident fact that all tight derivations for $\lambda x.p$ are obtained by applying a fun_r rule to a tight derivation for p.

3. *Application*, i.e. $t = pu$. Then $\mathtt{normal}_{lo}(t)$ implies $\mathtt{neutral}_{lo}(t)$, that implies $\mathtt{neutral}_{lo}(p)$ and $\mathtt{normal}_{lo}(u)$, and the first implies $\mathtt{normal}_{lo}(p)$.

   a. *Existence*: by *i.h.*, there are tight derivations

- $\Phi_p \triangleright_{lo} \Gamma_p \vdash^{(0,|p|_{lo})} p$: `neutral` typing p with `neutral` (because $\texttt{neutral}_{lo}(p)$), and
- $\Phi_u \triangleright_{lo} \Gamma_u \vdash^{(0,|u|_{lo})} u : \texttt{tight}$.

Then the following is a tight derivation Φ for $t = pu$ whose second index satisfies $|p|_{lo} + |u|_{lo} + 1 = |t|_{lo}$, as required:

$$\frac{\Phi_p \triangleright_{lo} \Gamma_p \vdash^{(0,|p|_{lo})} p : \texttt{neutral} \quad \Phi_u \triangleright_{lo} \Gamma_u \vdash^{(0,|u|_{lo})} u : \texttt{tight}}{\Gamma_p \uplus \Gamma_u \vdash^{(0,|p|_{lo}+|u|_{lo}+1)} pu : \texttt{neutral}} \, \mathsf{app}_r^{lo}$$

b. *Structure*: moreover, $\texttt{neutral}_{lo}(t)$ and Φ does indeed type t with `neutral`. Dually, t is not an abstraction and so that point trivially holds.

c. *Unique size*: from $\texttt{neutral}_{lo}(t)$ we obtain $\texttt{neutral}_{hd}(t)$. Now consider a tight derivation Ψ for t. By Lemma 4.3 the last rule of Ψ is app_r^{lo} and so – exactly as Φ in the first point – the two premisses Ψ_p and Ψ_u of the last rule are both tight. Then by *i.h.* $|\Psi_p| = |\Phi_p|$ and $|\Psi_u| = |\Phi_u|$, from which it follows $|\Psi| = |\Phi|$. □

Lemma 4.9 (Anti-substitution and typings for lo). *Let $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} t\{x \leftarrow p\} : A$. Then there exist*

- *a multi-set M;*
- *a typing derivation $\Phi_t \triangleright_{lo} \Gamma_t; x : M \vdash^{(b_t,r_t)} t : A$; and*
- *a typing derivation $\Phi_p \triangleright_{lo} \Gamma_p \vdash^{(b_p,r_p)} p : M$*

such that

- Typing context*: $\Gamma = \Gamma_t \uplus \Gamma_p$;*
- Indices*: $(b, r) = (b_t + b_p, r_t + r_p)$.*
- Sizes*: $|\Phi| = |\Phi_t| + |\Phi_p| - |M|$.*
- *If Φ is traditional, then Φ_t and Φ_p are traditional too.*

Proof By induction on t. Cases:

- *Variable*, i.e. $t = y$. Two subcases, depending on the identity of y:

  1. $x = y$. Then $t\{x \leftarrow p\} = x\{x \leftarrow p\} = p$, so that $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} p : A$. There is only one possibility: $|M| = 1$, Φ_p is

  $$\frac{\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} p : A}{\Gamma \vdash^{(b,r)} p : [A]} \, \mathsf{many}$$

  and Φ_t is

  $$\frac{}{x : [A] \vdash^{(0,0)} x : A} \, \mathsf{ax}$$

  that satisfies the all the equalities in the statement, in particular $|\Phi_p| = |\Phi|$ and $|M| = 1 = |\Phi_t|$.

  2. $x \neq y$. Then $t\{x \leftarrow p\} = y\{x \leftarrow p\} = y$. There is only one possibility: $|M| = 0$, Φ_t is exactly Φ, that is,

  $$\frac{}{y : [A] \vdash^{(0,0)} y : A} \, \mathsf{ax}$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

and Φ_p is

$$\frac{}{\vdash^{(0,0)} p : [\,]}\ \mathsf{many}$$

that satisfies the all the equalities in the statement, in particular $|\Phi_t| = |\Phi|$ and $|M| = 0 = |\Phi_p|$.

- *Abstraction*, i.e. $t = \lambda y.u$. Then $t\{x{\leftarrow}p\} = \lambda y.u\{x{\leftarrow}p\}$. Two sub-cases, depending on the last rule of Φ:

  1. *Rule* fun_b. Then Φ has the following form:

  $$\frac{\Phi_{u\{x\leftarrow p\}} \rhd_{lo}\ \Gamma; y : N \vdash^{(b_{u\{x\leftarrow p\}},r)} u\{x{\leftarrow}p\} : D}{\Gamma \vdash^{(b_{u\{x\leftarrow p\}}+1,r)} \lambda y.u\{x{\leftarrow}p\} : N \rightarrow D}\ \mathsf{fun}_b$$

  with $b = b_{u\{x\leftarrow p\}} + 1$. By *i.h.* there exist a M and type derivations

  $$\Phi_u \rhd_{lo}\ \Delta_u; y : N; x : M \vdash^{(b_u,r_u)} u : A \qquad \Phi_p \rhd_{lo}\ \Delta_p \vdash^{(b_p,r_p)} p : M$$

  such that

  - *Typing context*: $(\Gamma; y : N) = (\Delta_u; y : N \uplus \Delta_p)$;
  - *Indices*: $(b_{u\{x\leftarrow p\}}, r) = (b_u + b_p, r_u + r_p)$.
  - *Sizes*: $|\Phi_{u\{x\leftarrow p\}}| = |\Phi_u| + |\Phi_p| - |M|$.

  Then the derivation Φ_t defined as

  $$\frac{\Phi_u \rhd_{lo}\ \Gamma; y : N; x : M \vdash^{(b_u,r_u)} u : D}{\Gamma; x : M \vdash^{(b_u+1,r_u)} \lambda y.u : N \rightarrow D}\ \mathsf{fun}_b$$

  satisfies the statement with respect to $b_t := b_u + 1$ and $r_t := r_u$ because

  - *Typing context*: the *i.h.* implies $\Gamma = (\Delta_u \uplus \Delta_p)$;
  - *Indices*:
    a. $b_t + b_p = b_u + 1 + b_p =_{i.h.} b_{u\{x\leftarrow p\}} + 1 = b$,
    b. $r_t + r_p = r_u + r_p =_{i.h.} r_{u\{x\leftarrow p\}} = r$.
  - *Sizes*: $|\Phi| = |\Phi_{u\{x\leftarrow p\}}| + 1 =_{i.h.} |\Phi_u| + |\Phi_p| - |M| + 1 = |\Phi_t| + |\Phi_p| - |M|$.

  2. *Rule* fun_r. Then Φ has the following form:

  $$\frac{\Phi_{u\{x\leftarrow p\}} \rhd_{lo}\ \Gamma; y : \mathtt{Tight} \vdash^{(b,r_{u\{x\leftarrow p\}})} u\{x{\leftarrow}p\} : \mathtt{tight}}{\Gamma \vdash^{(b,r_{u\{x\leftarrow p\}}+1)} \lambda y.u\{x{\leftarrow}p\} : \mathtt{abs}}\ \mathsf{fun}_r$$

  with $r = r_{u\{x\leftarrow p\}} + 1$. By *i.h.* there exist M and type derivations

  $$\Phi_u \rhd \Delta_u; y : \mathtt{Tight}; x : M \vdash^{(b_u,r_u)} u : \mathtt{tight} \qquad \Phi_p \rhd \Delta_p \vdash^{(b_p,r_p)} p : M$$

  such that

  - *Typing context*: $(\Gamma; y : \mathtt{Tight}) = (\Delta_u; y : \mathtt{Tight} \uplus \Delta_p)$;
  - *Indices*: $(b, r_{u\{x\leftarrow p\}}) = (b_u + b_p, r_u + r_p)$.
  - *Sizes*: $|\Phi_{u\{x\leftarrow p\}}| = |\Phi_u| + |\Phi_p| - |M|$.

Then the derivation Φ_t defined as

$$\frac{\Phi_u \rhd_{lo} \ \Gamma; y : \mathtt{Tight}; x : M \vdash^{(b_u, r_u)} u : \mathtt{tight}}{\Gamma; x : M \vdash^{(b_u, r_u+1)} \lambda y.u : \mathtt{abs}} \ \mathsf{fun}_b$$

satisfies the statement with respect to $r_t := r_u + 1$ because

- *Typing context*: the *i.h.* implies $\Gamma = (\Delta_u \uplus \Delta_p)$
- *Indices*:

  a. $b_u + b_p =_{i.h.} b$,
  b. $r_t + r_p = r_u + 1 + r_p =_{i.h.} r_{u\{x \leftarrow p\}} + 1 = r$.
- *Sizes*: $|\Phi| = |\Phi_{u\{x \leftarrow p\}}| + 1 =_{i.h.} |\Phi_u| + |\Phi_p| - |M| + 1 = |\Phi_t| + |\Phi_p| - |M|$.

- *Application*, i.e. $t = uq$. Then $t\{x \leftarrow p\} = u\{x \leftarrow p\} q\{x \leftarrow p\}$. Two sub-cases, depending on the last rule of Φ:

1. *Rule* app_b. Then Φ has the following form:

$$\frac{\Phi_{u\{x \leftarrow p\}} \rhd_{lo} \ \Gamma_1 \vdash^{(b_1, r_1)} u\{x \leftarrow p\} : M \to A \qquad \Phi_{q\{x \leftarrow p\}} \rhd_{lo} \ \Gamma_2 \vdash^{(b_2, r_2)} q\{x \leftarrow p\} : M}{\Gamma_1 \uplus \Gamma_2 \vdash^{(b_1+b_2, r_1+r_2)} u\{x \leftarrow p\} q\{x \leftarrow p\} : A} \ \mathsf{app}_b$$

with $\Gamma = \Gamma_1 \uplus \Gamma_2$, $b = b_1 + b_2$, and $r = r_1 + r_2$.
By *i.h.* applied to $u\{x \leftarrow p\}$ and $q\{x \leftarrow p\}$, there exist (disjoint) finite sets M_u and M_q and type derivations:

$$\Phi_u \rhd_{lo} \ \Delta_u; x : M_u \vdash^{(b_u, r_u)} u : M \to A$$

$$\Phi_q \rhd_{lo} \ \Delta_q; x : M_q \vdash^{(b_q, r_q)} q : M$$

$$\Phi_p^u \rhd_{lo} \ \Pi_u \vdash^{(b_p^u, r_p^u)} p : M_u$$

$$\Phi_p^q \rhd_{lo} \ \Pi_q \vdash^{(b_p^q, r_p^q)} p : M_q$$

such that

- *Type context*: $\Gamma_1 = \Delta_u \uplus \Pi_u$ and $\Gamma_2 = \Delta_q \uplus \Pi_q$.
- *Indices*: $(b_1, r_1) = (b_u + b_p^u, r_u + r_p^u)$ and $(b_2, r_2) = (b_q + b_p^q, r_q + r_p^q)$.
- *Sizes*: $|\Phi_{u\{x \leftarrow p\}}| = |\Phi_u| + |\Phi_p^u| - |M_u|$ and $|\Phi_{q\{x \leftarrow p\}}| = |\Phi_q| + |\Phi_p^q| - |M_q|$.

The derivations Φ_p^u and Φ_p^q can be summed (by inverting their many final rule and reapplying a many rule to the union of the premisses) obtaining a derivation $\Phi_p \rhd_{lo} \ \Pi \vdash^{(b_p, r_p)} p : M$, where $\Pi = \Pi_u \uplus \Pi_q$ and $b_p = b_p^u + b_p^q$ and $r_p = r_p^u + r_p^q$ and $M = M_u + M_q$ and $|\Phi_p| = |\Phi_p^u| + |\Phi_p^q|$. We then apply app_b to obtain the following derivation Φ_t:

$$\frac{\Phi_u \rhd_{lo} \ \Delta_u; x : M_u \vdash^{(b_u, r_u)} u : M \to A \qquad \Phi_q \rhd_{lo} \ \Delta_q; x : M_q \vdash^{(b_q, r_q)} q : M}{\Delta_u \uplus \Delta_q; x : M_u + M_q \vdash^{(b_u+b_q, r_u+r_q)} uq : A} \ \mathsf{app}_b$$

We let $\Delta := \Delta_u \uplus \Delta_q$, $b_t := b_u + b_q$ and $r_t := r_u + r_q$ and then observe that we obtained the statement, because of the following equalities:

a. *Typing context*: $\Gamma = \Gamma_1 \uplus \Gamma_2 = \Delta_u \uplus \Pi_u \uplus \Delta_q \uplus \Pi_q = \Delta \uplus \Pi$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

b. *Indices*: $(b,r) = (b_1 + b_2, r_1 + r_2) = (b_t + b_p, r_t + r_p)$.
c. *Sizes*: $|\Phi| = |\Phi_{u\{x\leftarrow p\}}| + |\Phi_{q\{x\leftarrow p\}}| + 1 =_{i.h.} |\Phi_u| + |\Phi_p^u| - |M_u| + |\Phi_q| + |\Phi_p^q| - |M_q| + 1 = |\Phi_u| + |\Phi_q| + |\Phi_p| - |M| + 1 = |\Phi_t| + |\Phi_p| - |M|$.

2. *Rule* app_r^{lo}. Let $t = uq$ so that $t\{x\leftarrow p\} = u\{x\leftarrow p\}q\{x\leftarrow p\}$. Then Φ has the following form:

$$\frac{\Phi_{u\{x\leftarrow p\}} \rhd_S \ \Gamma_1 \vdash^{(b_1,r_1)} u\{x\leftarrow p\} : \mathtt{neutral} \qquad \Phi_{q\{x\leftarrow p\}} \rhd_S \ \Gamma_2 \vdash^{(b_2,r_2)} q\{x\leftarrow p\} : \mathtt{tight}}{\Gamma_1 \uplus \Gamma_2 \vdash^{(b_1+b_2,r_1+r_2+1)} u\{x\leftarrow p\}q\{x\leftarrow p\} : \mathtt{tight}}\ \mathsf{app}_r^{lo}$$

with $\Gamma = \Gamma_1 \uplus \Gamma_2$, $b = b_1 + b_2$, $r = r_1 + r_2 + 1$.
By *i.h.* applied to $u\{x\leftarrow p\}$ and $q\{x\leftarrow p\}$, there exist (disjoint) finite sets M_u and M_q and type derivations:

$$\Phi_u \rhd_{lo} \ \Delta_u; x : M_u \vdash^{(b_u,r_u)} u : \mathtt{neutral}$$

$$\Phi_q \rhd_{lo} \ \Delta_q; x : M_q \vdash^{(b_q,r_q)} q : \mathtt{tight}$$

$$\Phi_p^u \rhd_{lo} \ \Pi_u \vdash^{(b_p^u,r_p^u)} p : M_u$$

$$\Phi_p^q \rhd_{lo} \ \Pi_q \vdash^{(b_p^q,r_p^q)} p : M_q$$

such that

- *Typing context*: $\Gamma_1 = \Delta_u \uplus \Pi_u$ and $\Gamma_2 = \Delta_q \uplus \Pi_q$.
- *Indices*: $(b_1, r_1) = (b_u + b_p^u, r_u + r_p^u)$ and $(b_2, r_2) = (b_q + b_p^q, r_q + r_p^q)$.
- *Sizes*: $|\Phi_{u\{x\leftarrow p\}}| = |\Phi_u| + |\Phi_p^u| - |M_u|$ and $|\Phi_{q\{x\leftarrow p\}}| = |\Phi_q| + |\Phi_p^q| - |M_q|$.

The derivations Φ_p^u and Φ_p^q can be summed (by inverting their many final rule and reapplying a many rule to the union of the premisses) obtaining a derivation $\Phi_p \rhd_{lo} \ \Pi \vdash^{(b_p,r_p)} p : M$, where $\Pi = \Pi_u \uplus \Pi_q$ and $b_p = b_p^u + b_p^q$ and $r_p = r_p^u + r_p^q$ and $M = M_u + M_q$. We then apply app_r^{lo} to obtain the following derivation Φ_t:

$$\frac{\Phi_u \rhd_{lo} \ \Delta_u; x : M_u \vdash^{(b_u,r_u)} u : \mathtt{neutral} \qquad \Phi_q \rhd_{lo} \ \Delta_q; x : M_q \vdash^{(b_q,r_q)} q : \mathtt{tight}}{\Delta_u \uplus \Delta_q; x : M_u + M_q \vdash^{(b_u+b_q,r_u+r_q+1)} uq : \mathtt{neutral}}\ \mathsf{app}_r^{lo}$$

We let $\Delta := \Delta_u \uplus \Delta_q$, $b_t := b_u + b_q$ and $r_t := r_u + r_q + 1$ and then observe that we obtained the statement, because of the following equalities:

a. *Typing context*: $\Gamma = \Gamma_1 \uplus \Gamma_2 = \Delta_u \uplus \Pi_u \uplus \Delta_q \uplus \Pi_q = \Delta \uplus \Pi$.
b. *Indices*: $(b,r) = (b_1 + b_2, r_1 + r_2 + 1) = (b_t + b_p, r_t + r_p)$.
c. *Sizes*: $|\Phi| = |\Phi_{u\{x\leftarrow p\}}| + |\Phi_{q\{x\leftarrow p\}}| + 1 =_{i.h.} |\Phi_u| + |\Phi_p^u| - |M_u| + |\Phi_q| + |\Phi_p^q| - |M_q| + 1 = |\Phi_u| + |\Phi_q| + |\Phi_p| - |M| + 1 = |\Phi_t| + |\Phi_p| - |M|$. □

Proposition 4.10 (Quantitative tight subject expansion for *lo*). *Let* $\Phi \rhd_{lo} \ \Gamma \vdash^{(b,r)} p : A$ *be a tight derivation. If* $t \to_{lo} p$ *then there exists a (tight) typing* Ψ *such that* $\Psi \rhd_{lo} \ \Gamma \vdash^{(b+1,r)} t : A$ *and* $|\Psi| > |\Phi|$.

Proof We prove the following stronger statement by induction on $t \to_{lo} p$ (tightness is decomposed in two predicates $\mathtt{tight}(\Gamma)$ and $\mathtt{tight}(A)$, and the second is paired together with a further assumption).

Let $t \to_{lo} p$, $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} p : A$, $\mathtt{tight}(\Gamma)$, and either $\mathtt{tight}(A)$ or $\neg\mathtt{abs}_{lo}(t)$. Then there exists a typing $\Psi \triangleright_{lo} \Gamma \vdash^{(b+1,r)} t : A$ with $|\Psi| > |\Phi|$.

- Rule

$$\frac{}{t = (\lambda x.u)q \to_{lo} u\{x{\leftarrow}q\} = p}$$

Assume $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} u\{x{\leftarrow}q\} : A$ and $\mathtt{tight}(\Gamma)$. By applying the anti-substitution lemma (Lemma 4.9) we obtain the two premisses of the following derivation Φ' that satisfies the statement:

$$\frac{\dfrac{\Phi_u \triangleright_{lo} \Gamma_u, x : M \vdash^{(b_u,r_u)} u : A}{\Gamma_u \vdash^{(b_u+1,r_u)} \lambda x.u : M \to A} \qquad \Phi_q \triangleright_{lo} \Gamma_q \vdash^{(b_q,r_q)} q : M}{\Gamma_u \uplus \Gamma_q \vdash^{(b_u+b_q+2,r_u+r_q)} (\lambda x.u)q : A}$$

with $(b,r) = (b_u + b_q, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_q$. We conclude since $|\Psi| = |\Phi_u| + |\Phi_q| + 2 > |\Phi_u| + |\Phi_q| - |M| = |\Phi|$.

- Rule

$$\frac{u \to_{lo} q}{t = \lambda x.u \to_{lo} \lambda x.q = p}$$

Assume $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} \lambda x.q : A$ and $\mathtt{tight}(\Gamma)$. Since $\mathtt{abs}_{lo}(\lambda x.u)$ we must have hypothesis $\mathtt{tight}(A)$, and as Φ must then finish with rule fun_r we must have a sub-derivation $\Phi_q \triangleright_{lo} \Gamma, x : \mathtt{Tight} \vdash^{(b,r-1)} q : \mathtt{tight}$. As $\mathtt{tight}(\Gamma, x : \mathtt{Tight})$, we can apply the *i.h.* and get the premiss of the derivation Φ' as follows:

$$\frac{\Phi_u \triangleright_{lo} \Gamma, x : \mathtt{Tight} \vdash^{(b+1,r-1)} u : \mathtt{tight}}{\Gamma \vdash^{(b+1,r)} \lambda x.u : A}$$

The decrement of the size follows from the *i.h.*

- Rule

$$\frac{\neg\mathtt{abs}_{lo}(u) \quad u \to_{lo} q}{t = um \to_{lo} qm = p}$$

Assume $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} qm : A$ and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule app_b or app_r^{lo}. Then, there are derivations $\Phi_q \triangleright_{lo} \Gamma_q \vdash^{(b_q,r_q)} q : A_q$ and $\Phi_m \triangleright_{lo} \Gamma_m \vdash^{(b_m,r_m)} m : A_m$, with $\Gamma = \Gamma_q \uplus \Gamma_m$. Since $\mathtt{tight}(\Gamma)$ we have $\mathtt{tight}(\Gamma_q)$, and since $\neg\mathtt{abs}_{lo}(u)$ we can apply the *i.h.* to q (independently of whether A_q is tight) obtaining the derivation $\Phi_u \triangleright_{lo} \Gamma_q \vdash^{(b_q+1,r_q)} u : A_q$ and build, using the same rule app_b or app_r^{lo}, the derivation Φ' next:

$$\frac{\Phi_u \triangleright_{lo} \Gamma_q \vdash^{(b_q+1,r_q)} u : A_q \qquad \Phi_m \triangleright_{lo} \Gamma_m \vdash^{(b_m,r_m)} m : A_m}{\Gamma \vdash^{(b+1,r)} um : A}$$

The decrement of the size follows from the *i.h.*

- Rule

$$\frac{\mathtt{neutral}_{lo}(m) \quad u \to_{lo} q}{t = mu \to_{lo} mq = p}$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Assume $\Phi \rhd_{lo} \; \Gamma \vdash^{(b,r)} mq : A$ and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule $\mathtt{app}_b$ or $\mathtt{app}_r^{lo}$, and therefore there are two derivations $\Phi_m \rhd_{lo} \; \Gamma_m \vdash^{(b_m,r_m)} m : A_m$ and $\Phi_q \rhd_{lo} \; \Gamma_q \vdash^{(b_q,r_q)} q : A_q$, for some types A_m and A_q, with $\Gamma = \Gamma_m \uplus \Gamma_q$. Since $\mathtt{tight}(\Gamma)$ we have $\mathtt{tight}(\Gamma_m)$ and $\mathtt{tight}(\Gamma_q)$. By the *tight spreading on neutral terms* (Lemma 4.3), from $\mathtt{tight}(\Gamma_m)$ and $\mathtt{neutral}_{lo}(m)$ it follows $\mathtt{tight}(A_m)$. Therefore, the last rule of Φ must be $\mathtt{app}_r^{lo}$, whence $A_m = A = \mathtt{neutral}$ and $A_q = \mathtt{tight}$. Now, the sub-derivation Φ_q is tight ($\mathtt{tight}(\Gamma_q)$ and $A_q = \mathtt{tight}$) and we can apply the *i.h.* obtaining the derivation $\Phi_u \rhd_{lo} \; \Gamma_q \vdash^{(b_q+1,r_q)} u : A_q$ and build, using the same rule $\mathtt{app}_r^{lo}$, the derivation Φ' next:

$$\frac{\Phi_m \rhd_{lo} \; \Gamma_m \vdash^{(b_m,r_m)} m : A_m \qquad \Phi_u \rhd_{lo} \; \Gamma_q \vdash^{(b_q+1,r_q)} u : A_q}{\Gamma \vdash^{(b+1,r)} mu : A}$$

The decrement of the size follows from the *i.h.* □

Theorem 4.11 (Tight completeness for *lo*). *Let $t \to_{lo}^k p$ with $\mathtt{normal}_{lo}(p)$. Then*

1. Existence*: there exists a tight typing $\Phi \rhd_{lo} \; \Gamma \vdash^{(k,|p|_{lo})} t : A$.*
2. Structure*: moreover, if $\mathtt{neutral}_{lo}(p)$ then $A = \mathtt{neutral}$, and if $\mathtt{abs}_{lo}(p)$ then $A = \mathtt{abs}$.*

Proof By induction on $t \to_{lo}^k p$. If $k = 0$ the statement is given by the existence of tight typings for $\mathtt{normal}_{lo}$ terms (Proposition 4.8), that also provides the *moreover* part. Let $k > 0$ and $t \to_{lo} u \to_{lo}^{k-1} p$. By *i.h.*, there exists a tight typing derivation $\Psi \rhd \vdash_{\mathtt{tight}}^{(k-1,|p|_{lo})} u$. By subject expansion (Proposition 4.10) there exists a typing derivation Φ of u with the same types in the ending judgement of Ψ – then Φ is tight – and with indices $(k, |p|_{lo})$. □

Appendix C Leftmost evaluation and minimal typings

Lemma 5.3 (Transitivity of polarities). *Let T, U, V be (multi)-types and $a, b \in \{+, -\}$. If $U \in \mathtt{Occ}_a(T)$ and $V \in \mathtt{Occ}_b(U)$ then $V \in \mathtt{Occ}_{\delta(a,b)}(T)$, where*

$$\delta(+,+) := + \qquad \delta(-,+) := - \qquad \delta(-,-) := + \qquad \delta(+,-) := -$$

Proof Let $\neg + := -$ and $\neg - := +$. The proof can be presented in a way that is completely parametric in the polarities, but for readability reasons we spell out the positive and negative cases separately. Cases in which $a = +$:

- *Axioms*, i.e. $U = T$. Note that $\delta(+, b) = b$. Then $V \in \mathtt{Occ}_b(U)$ becomes $V \in \mathtt{Occ}_b(T) = \mathtt{Occ}_{\delta(+,b)}(T)$ as required.
- *Positive occurrence in an element A of a multi-set M*, i.e. $T = M$ and $U \in \mathtt{Occ}_+(M)$ because $U \in \mathtt{Occ}_+(A)$. By the *i.h.* $V \in \mathtt{Occ}_{\delta(+,b)}(A)$ and so $V \in \mathtt{Occ}_{\delta(+,b)}(M)$ by one of the two rules about multi-sets.
- *Positive occurrence on the right of $M \to A$*, i.e. $T = M \to A$ and $U \in \mathtt{Occ}_+(M \to A)$ because $U \in \mathtt{Occ}_+(A)$. By the *i.h.* $V \in \mathtt{Occ}_{\delta(+,b)}(A)$ and so $V \in \mathtt{Occ}_{\delta(+,b)}(M)$ by one of the two rules about arrow types.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

- *Negative occurrence on the left of* $M \to A$, i.e. $T = M \to A$ and $U \in \mathtt{Occ}_+(M \to A)$ because $U \in \mathtt{Occ}_-(M)$. By the *i.h.* $V \in \mathtt{Occ}_{\delta(-,b)}(A)$ and so $V \in \mathtt{Occ}_{\neg\delta(-,b)}(M) = \mathtt{Occ}_{\delta(+,b)}(M)$ by one of the two rules about arrow types.

Cases in which $a = -$:

- *Negative occurrence in an element* A *of a multi-set* M, i.e. $T = M$ and $U \in \mathtt{Occ}_-(M)$ because $U \in \mathtt{Occ}_-(A)$. By the *i.h.* $V \in \mathtt{Occ}_{\delta(-,b)}(A)$ and so $V \in \mathtt{Occ}_{\delta(-,b)}(M)$ by one of the two rules about multi-sets.
- *Negative occurrence on the right of* $M \to A$, i.e. $T = M \to A$ and $U \in \mathtt{Occ}_-(M \to A)$ because $U \in \mathtt{Occ}_-(A)$. By the *i.h.* $V \in \mathtt{Occ}_{\delta(-,b)}(A)$ and so $V \in \mathtt{Occ}_{\delta(-,b)}(M)$ by one of the two rules about arrow types.
- *Positive occurrence on the left of* $M \to A$, i.e. $T = M \to A$ and $U \in \mathtt{Occ}_-(M \to A)$ because $U \in \mathtt{Occ}_+(M)$. By the *i.h.* $V \in \mathtt{Occ}_{\delta(+,b)}(A)$ and so $V \in \mathtt{Occ}_{\neg\delta(+,b)}(M) = \mathtt{Occ}_{\delta(-,b)}(M)$ by one of the two rules about arrow types. □

C.1 Shrinking correctness

Lemma 5.6 (Occurrences spreading on neutral terms). *Let* t *be such that* $\mathtt{neutral}_{hd}(t)$ *and* $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$ *be a typing derivation. Then* A *is a positive occurrence of* Γ. *Moreover, if* Γ *is co-shrinking (resp unitary co-shrinking) then* A *is co-shrinking (resp unitary co-shrinking).*

Proof By induction on $\mathtt{neutral}_{lo}(t)$:

- *Variable*, i.e. $t = x$. Then $\Gamma = x : [A]$ and $A \in \mathtt{Occ}_+(\Gamma)$. If Γ is co-shrinking (resp. unitary co-shrinking) then A is co-shrinking (resp. unitary co-shrinking) by definition of shrinking (resp. unitary shrinking) type context.
- *Application*, i.e. $t = pu$, the last rule of Φ can only be $\mathtt{app}_b$ or $\mathtt{app}_r^{lo}$. In both cases the left subterm p is typed by a sub-derivation $\Phi_p \rhd_{lo} \Gamma_p \vdash^{(b',r')} p : B$ such that all types in Γ_p appear in Γ. Since $\mathtt{neutral}_{hd}(t)$ implies $\mathtt{neutral}_{hd}(p)$, we can apply the *i.h.* and obtain that B has a positive occurrence in Γ_p, and thus in Γ, that is, that there is a declaration $x : M$ in Γ such that $B \in \mathtt{Occ}_+(M)$. There are two cases, either $B = A = \mathtt{neutral}$ or $B = M' \to A$. In both cases, A is a positive occurrence of B. By transitivity of polarised occurrences (Lemma 5.3), A is a positive occurrence of M, and thus of Γ. Let $M \in \mathtt{Occ}_-(A)$. Since $A \in \mathtt{Occ}_+(\Gamma)$ then $M \in \mathtt{Occ}_-(\Gamma)$ by transitivity of polarised occurrences. Suppose Γ is co-shrinking (resp. unitary co-shrinking), then A turns out to be co-shrinking (resp. unitary co-shrinking). □

Proposition 5.7 (Shrinking derivations bound the size of normal forms). *Let* $\mathtt{normal}_{lo}(t)$ *and* $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$ *be a derivation, and let* $|\Phi|_{\mathsf{ax}}$ *denote the number of axiom rules in* Φ.

1. *If* Γ *is co-shrinking and (*A *is shrinking or* t *is not an abstraction) then* $|t|_{lo} \le |\Phi| - |\Phi|_{\mathsf{ax}}$. *Moreover, if* Φ *is traditional then* $|t|_\lambda \le b$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

2. *If Γ is unitary co-shrinking and (A is unitary shrinking or t is not an abstraction) then $|t|_{lo} = |\Phi| - |\Phi|_{\mathsf{ax}}$. Moreover, if Φ is traditional then $|t|_\lambda = b$.*

Proof By induction on t. Note that $\mathtt{neutral}_{lo}$ implies $\mathtt{normal}_{lo}$ and so we can apply the *i.h.* when $\mathtt{neutral}_{lo}$ holds on some subterm of t. If $\mathtt{normal}_{lo}(t)$ because $\mathtt{neutral}_{lo}(t)$ there are three cases:

- *Variable*, i.e. $t = x$. Then Φ has the following form and evidently verifies both statements because $|x|_{lo} = 0 = 1 - 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$ and $|x|_\lambda = 0 = b$:

$$\frac{}{x : [A] \vdash^{(0,0)} x : A}\ \mathsf{ax}$$

- *Application*, i.e. $t = pu$, $\mathtt{neutral}_{lo}(p)$ and $\mathtt{normal}_{lo}(u)$. Cases of the last rule of Φ:
  - app_b *rule*:

$$\frac{\Phi_p \rhd_{lo}\ \Gamma_p \vdash^{(b_p,r_p)} p : [B_i]_{i\in I} \to A \qquad \dfrac{(\Phi_u^i \rhd_{lo}\ \Delta_i \vdash^{(b_i,r_i)} u : B_i)_{i\in I}}{\uplus_{i\in I}\Delta_i \vdash^{(+_{i\in I}b_i, +_{i\in I}r_i)} u : [B_i]_{i\in I}}\ \mathsf{many}}{\Gamma_p \uplus (\uplus_{i\in I}\Delta_i) \vdash^{(b_p +_{i\in I} b_i, r_p +_{i\in I} r_i)} pu : A}\ \mathsf{app}_b$$

with $b = b_p +_{i\in I} b_i$, $r = r_p +_{i\in I} r_i$, and $\Gamma = \Gamma_p \uplus_{i\in I} \Delta_i$. Let $M = [B_i]_{i\in I}$.

1. Since $\mathtt{neutral}_{lo}(p)$ and Γ is co-shrinking then $M \to A$ is co-shrinking by Lemma 5.6, and so M is shrinking. Therefore, M is not empty, i.e. $|I| \neq 0$, and every B_i is shrinking. Moreover, Γ_p and every Δ_i are also shrinking so that every Φ_u^i is shrinking.

   Since p is neutral and thus not an abstraction, we can apply the *i.h.* on $\mathtt{normal}_{lo}(p)$ and obtain $|p|_{lo} \leq |\Phi_p| - |\Phi_p|_{\mathsf{ax}}$. Since every Φ_u^i is shrinking we can apply the *i.h.* on $\mathtt{normal}_{lo}(u)$ obtaining $|u|_{lo} \leq |\Phi_u^i| \leq +_{i\in I}|\Phi_u^i| - |\Phi_u^i|_{\mathsf{ax}}$, thus $|t|_{lo} = |p|_{lo} + |u|_{lo} + 1 \leq |\Phi_p| - |\Phi_p|_{\mathsf{ax}} +_{i\in I} (|\Phi_u^i| - |\Phi_u^i|_{\mathsf{ax}}) + 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$.

   Moreover, if Φ is traditional so are its sub-derivations and so the *i.h.* on $\mathtt{normal}_{lo}(p)$ gives $|p|_\lambda \leq b_p$ and the *i.h.* on $\mathtt{normal}_{lo}(u)$ gives $|u|_\lambda \leq b_i$ for all $i \in I$. Then $|t|_\lambda = |p|_\lambda + |u|_\lambda \leq b_p +_{i\in I} b_i = b$.
2. Since $\mathtt{neutral}_{lo}(p)$ and Γ is unitary co-shrinking then $M \to A$ is unitary co-shrinking by Lemma 5.6, and so M is unitary shrinking. Therefore, M is a singleton, i.e. $M = [B_1]$, and B_1 is unitary shrinking. Moreover, Γ_p and Δ_1 are also unitary shrinking so that Φ_u^1 is unitary shrinking.

   Since p is neutral and thus not an abstraction, we can apply the *i.h.* on $\mathtt{normal}_{lo}(p)$ and obtain $|p|_{lo} = |\Phi_p| - |\Phi_p|_{\mathsf{ax}}$. Since Φ_u^1 is unitary shrinking we can apply the *i.h.* on $\mathtt{normal}_{lo}(u)$ obtaining $|u|_{lo} = |\Phi_u^1| - |\Phi_u^1|_{\mathsf{ax}}$, thus $|t|_{lo} = |p|_{lo} + |u|_{lo} + 1 = |\Phi_p| - |\Phi_p|_{\mathsf{ax}} + |\Phi_u^1| - |\Phi_u^1|_{\mathsf{ax}} + 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$.

   Moreover, if Φ is traditional so are its sub-derivations and so the *i.h.* on $\mathtt{normal}_{lo}(p)$ gives $|p|_\lambda = b_p$ and the *i.h.* on $\mathtt{normal}_{lo}(u)$ gives $|u|_\lambda = b_1$. Then $|t|_\lambda = |p|_\lambda + |u|_\lambda = b_p + b_1 = b$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

– app_r^{lo} *rule*:

$$\frac{\Phi_p \triangleright_{lo} \;\; \Gamma_p \vdash^{(b_p,r_p)} p : \mathtt{neutral} \quad \Phi_u \triangleright_{lo} \;\; \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{tight}}{\Gamma_p \uplus \Gamma_u \vdash^{(b_p+b_u,r_p+r_u+1)} pu : \mathtt{neutral}} \; \mathsf{app}_r^{lo}$$

with $b = b_p + b_u$, $r = r_p + r_u + 1$, and $\Gamma = \Gamma_p \uplus \Gamma_u$.

1. Since Γ is co-shrinking, then Γ_p and Γ_u are co-shrinking. Since $\mathtt{neutral}$ and $\mathtt{tight}$ are shrinking types, by *i.h.* $|p|_{lo} \leq |\Phi_p| - |\Phi_p|_{\mathsf{ax}}$ and $|u|_{lo} \leq |\Phi_u| - |\Phi_u|_{\mathsf{ax}}$. Then $|t|_{lo} = |p|_{lo} + |u|_{lo} + 1 \leq_{i.h.} |\Phi_p| - |\Phi_p|_{\mathsf{ax}} + |\Phi_u| - |\Phi_u|_{\mathsf{ax}} + 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$. The moreover statement does not apply in this case.
2. Since Γ is unitary co-shrinking, then Γ_p and Γ_u are unitary co-shrinking. Since $\mathtt{neutral}$ and $\mathtt{tight}$ are unitary shrinking types, by *i.h.* $|p|_{lo} = |\Phi_p| - |\Phi_p|_{\mathsf{ax}}$ and $|u|_{lo} = |\Phi_u| - |\Phi_u|_{\mathsf{ax}}$. Then $|t|_{lo} = |p|_{lo} + |u|_{lo} + 1 =_{i.h.} |\Phi_p| - |\Phi_p|_{\mathsf{ax}} + |\Phi_u| - |\Phi_u|_{\mathsf{ax}} + 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$. The moreover statement does not apply in this case.

Now, there is only one case left for $\mathtt{normal}_{lo}(t)$:

- *Abstraction*, i.e. $t = \lambda x.p$ and $\mathtt{normal}_{lo}(t)$ because $\mathtt{normal}_{lo}(p)$. Cases of the last rule of Φ:

  – fun_b *rule*:

$$\frac{\Phi_p \triangleright_S \;\; \Gamma; x : M \vdash^{(b_p,r)} p : A}{\Gamma \vdash^{(b_p+1,r)} \lambda x.p : M \to A} \; \mathsf{fun}_b$$

  with $b = b_p + 1$.

  1. Since t is an abstraction, it must hold that $M \to A$ is shrinking, that is, A is shrinking, and M is co-shrinking. This last fact, together with the hypothesis that Γ is co-shrinking gives $\Gamma; x : M$ co-shrinking. Then we can apply the *i.h.* obtaining $|p|_{lo} \leq |\Phi_p| - |\Phi_p|_{\mathsf{ax}}$, and $|t|_{lo} = |p|_{lo} + 1 \leq_{i.h.} |\Phi_p| - |\Phi_p|_{\mathsf{ax}} + 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$.
     Moreover, if Φ is traditional so is Φ_p and the *i.h.* gives $|p|_\lambda \leq b_p$. Then $|t|_\lambda = |p|_\lambda + 1 \leq b_p + 1 = b$.
  2. Since t is an abstraction, it must hold that $M \to A$ is unitary shrinking, that is, A is unitary shrinking and M is unitary co-shrinking. This last fact, together with the hypothesis that Γ is unitary co-shrinking gives $\Gamma; x : M$ unitary shrinking. Then we can apply the *i.h.*, obtaining $|p|_{lo} = |\Phi_p| - |\Phi_p|_{\mathsf{ax}}$. Then, $|t|_{lo} = |p|_{lo} + 1 =_{i.h.} |\Phi_p| - |\Phi_p|_{\mathsf{ax}} + 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$.
     Moreover, if Φ is traditional so is Φ_p and the *i.h.* gives $|p|_\lambda = b_p$. Then $|t|_\lambda = |p|_\lambda + 1 = b_p + 1 = b$.

  – fun_r *rule*:

$$\frac{\Phi_p \triangleright \Gamma; x : \mathtt{Tight} \vdash^{(b,r_p)} p : \mathtt{tight}}{\Gamma \vdash^{(b,r_p+1)} \lambda x.p : \mathtt{abs}} \; \mathsf{fun}_r$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

with $r = r_p + 1$.

1. If Γ is co-shrinking, then $\Gamma; x : \texttt{Tight}$ is co-shrinking. Since $\texttt{tight}$ is a shrinking type, by *i.h.* $|p|_{lo} \leq |\Phi_p| - |\Phi_p|_{\mathsf{ax}}$. Then, $|t|_{lo} = |p|_{lo} + 1 \leq_{i.h.} |\Phi_p| - |\Phi_p|_{\mathsf{ax}} + 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$. The moreover statement does not apply in this case.
2. If Γ is unitary co-shrinking, then $\Gamma; x : \texttt{Tight}$ is unitary co-shrinking. Since $\texttt{tight}$ is a unitary shrinking type, by *i.h.* $|p|_{lo} = |\Phi_p| - |\Phi_p|_{\mathsf{ax}}$. Then, $|t|_{lo} = |p|_{lo} + 1 =_{i.h.} |\Phi_p| - |\Phi_p|_{\mathsf{ax}} + 1 = |\Phi| - |\Phi|_{\mathsf{ax}}$. The moreover statement does not apply in this case. $\square$

Proposition 5.8 (Traditional types bounds the size of neutral and normal terms). *Let $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} t : A$ be a traditional derivation such that Γ is co-shrinking. Then*

1. *if $\texttt{neutral}_{lo}(t)$ then $\#(A) + |t|_{lo} \leq \#(\Gamma)$ and $\#^N(A) + b \leq \#^N(\Gamma)$;*
2. *if $\texttt{normal}_{lo}(t)$ and A is shrinking then $|t|_{lo} \leq \#(\Gamma) + \#(A)$ and $b \leq \#^N(\Gamma) + \#^P(A)$.*

Proof By mutual induction on $\texttt{neutral}_{lo}(t)$ and $\texttt{normal}_{lo}(t)$.

1. Cases of $\texttt{neutral}_{lo}(t)$:

   – *Variable*, i.e. $t = x$. Then
   $$\frac{}{x : [A] \vdash^{(0,0)} x : A}\ \mathsf{ax}$$
   Moreover, $\#(A) + |x|_{lo} = \#(A) + 0 = \#(A) = \#([A]) = \#(x : [A])$ and $\#^N(A) + b = \#^N(A) + 0 = \#^N(A) = \#^N([A]) = \#^N(x : [A])$.

   – *Application*, i.e. $t = pu$ with $\texttt{neutral}_{lo}(p)$ and $\texttt{normal}_{lo}(u)$. The hypothesis that Φ is traditional forces the last rule of Φ to be app_b and Φ to have the following form:
   $$\frac{\Phi_p \triangleright_{lo}\ \Gamma_p \vdash^{(b_p,0)} p : [B_i]_{i\in I} \to A \qquad \dfrac{(\Phi_u \triangleright_{lo}\ \Delta_i \vdash^{(b'_i,0)} u : B_i)_{i\in I}}{\uplus_{i\in I}\Delta_i \vdash^{(+_{i\in I}b'_i,0)} u : [B_i]_{i\in I}}\ \mathsf{many}}{\Gamma_p \uplus (\uplus_{i\in I}\Delta_i) \vdash^{(b_p+_{i\in I}b'_i,0)} pu : A}\ \mathsf{app}_b$$
   Let $M := [B_i]_{i\in I}$ and $\Gamma_u = \uplus_{i\in I}\Delta_i$. Since Φ is shrinking, Γ_p is co-shrinking. The hypothesis $\texttt{neutral}_{lo}(p)$ gives $\texttt{neutral}_{hd}(p)$ (which is a weaker predicate), and by the occurrences spreading on neutral terms (Lemma 5.6) we obtain that $M \to A$ is co-shrinking and so M is shrinking. Therefore, M is not empty, that is, $I \neq \emptyset$ and each B_i is shrinking.

   By *i.h.* (Point 2) (repeatedly) applied to u, we obtain $|u|_{lo} \leq \#(\Delta_i) + \#(B_i)$ and $b'_i \leq \#^N(\Delta_i) + \#^P(B_i)$ for every $i \in I$, and so $|u|_{lo} \leq \#(\Gamma_u) + \#(M)$ and $+_{i\in I}b'_i \leq +_{i\in I}(\#^N(\Delta_i) + \#^P(B_i)) = +_{i\in I}\#^N(\Delta_i) + \#^P(M)$.

   By *i.h.* (Point 1) applied to p, we obtain $\#(M \to A) + |p|_{lo} \leq \#(\Gamma_p)$ and $\#^N(M \to A) + b_p \leq \#^N(\Gamma_p)$.

Then:

$$\begin{array}{rcl} \#(A)+|t|_{lo} & = & \#(A)+|p|_{lo}+|u|_{lo}+1 \\ & \leq_{i.h.\text{ on }u} & \#(A)+|p|_{lo}+\#(\Gamma_u)+\#(M)+1 \\ & = & \#(\Gamma_u)+|p|_{lo}+\#(M \to A) \\ & \leq_{i.h.\text{ on }p} & \#(\Gamma_u)+\#(\Gamma_p) \\ & = & \#(\Gamma_p \uplus \Gamma_u) = \#(\Gamma) \end{array}$$

and

$$\begin{array}{rcl} \#^N(A)+b = \#^N(A)+b_p+_{i\in I} b'_i & \leq_{i.h.\text{ on }u} & \#^N(A)+b_p+_{i\in I}\#^N(\Delta_i)+\#^P(M) \\ & = & \#^N(\Gamma_u)+\#^N(M\to A)+b_p \\ & \leq_{i.h.\text{ on }p} & \#^N(\Gamma_u)+\#^N(\Gamma_p) \\ & = & \#^N(\Gamma_p \uplus \Gamma_u) = \#^N(\Gamma) \end{array}$$

2. Cases of $\mathtt{normal}_{lo}(t)$:

   a. $\mathtt{neutral}_{lo}(t)$. By *i.h.*, $\#(A)+|t|_{lo} \leq \#(\Gamma)$ and $\#^N(A)+b \leq \#^N(\Gamma)$, from which it trivially follows $|t|_{lo} \leq \#(\Gamma)+\#(A)$ and $b \leq \#^N(\Gamma) \leq \#^N(\Gamma)+\#^P(A)$.

   b. *Abstraction*, i.e. $t = \lambda y.p$ and $\mathtt{normal}_{lo}(p)$. Since Φ is traditional, its last rule is necessarily fun_b. Then let $y : M$ the declaration of y in the premiss of fun_b (remark that M is possibly $[\,]$). Then Φ has the following form:

$$\frac{\Phi_p \rhd_{lo}\ y : M; \Gamma \vdash^{(b_p,0)} p : B}{\Gamma \vdash^{(b_p+1,0)} \lambda y.p : M \to B}\ \mathsf{fun}_b$$

   with $b = b_p + 1$ and $A = M \to B$ shrinking, that implies B shrinking and M co-shrinking, that is, $\Gamma; y : M$ is co-shrinking (because Γ is co-shrinking by hypothesis). We can then apply the *i.h.* and obtain:

$$\begin{array}{rcl} |\lambda y.p|_{lo} & = & |p|_{lo}+1 \\ & \leq_{i.h.} & \#(y : M; \Gamma)+\#(B)+1 \\ & = & \#(\Gamma)+\#(M)+\#(B)+1 \\ & = & \#(\Gamma)+\#(M \to B) \end{array}$$

   and

$$\begin{array}{rcl} b_p+1 & \leq_{i.h.} & \#^N(y : M; \Gamma)+\#^P(B)+1 \\ & = & \#^N(\Gamma)+\#^N(M)+\#^P(B)+1 \\ & = & \#^N(\Gamma)+\#^P(M \to B) \end{array}$$

□

Proposition 5.9 (Shrinking subject reduction). *Let $\Phi \rhd_{lo}\ \Gamma \vdash^{(b,r)} t : A$. If $t \to_{lo} p$ then $b \geq 1$ and there exists Ψ such that $\Psi \rhd_{lo}\ \Gamma \vdash^{(b',r)} p : A$ with $b' \leq b$ and $|\Psi| \leq |\Phi|$. Moreover, Φ traditional implies Ψ traditional, and if Φ is shrinking (resp. unitary shrinking) then $b' < b$ (resp. $b' = b - 1$) and $|\Psi| < |\Phi|$.*

Proof The first part (without the shrinking/unitary shrinking hypothesis) is an easy induction on $t \rightarrow_{lo} p$. The *moreover* part is also by induction on $t \rightarrow_{lo} p$, but it requires a strengthened statement, along the same lines of the proof for the tight case:

1. If $t \rightarrow_{lo} p$, $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} t : A$, Γ is co-shrinking, and (A is shrinking or $\neg\mathtt{abs}_{lo}(t)$), then there exists a typing $\Psi \triangleright_{lo} \Gamma \vdash^{(b',r)} p : A$ with $b' < b$ and $|\Psi| < |\Phi|$.
2. If $t \rightarrow_{lo} p$, $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} t : A$, Γ is unitary co-shrinking, and A is unitary shrinking or $\neg\mathtt{abs}_{lo}(t)$, then there exists a typing $\Psi \triangleright_{lo} \Gamma \vdash^{(b',r)} p : A$ with $b' = b - 1$ and $|\Psi| < |\Phi|$.

The cases of evaluation at top level, under abstraction, and in the left subterm of an application follows exactly the schema of the tight case: at top level the tight/shrinking hypothesis does not play any role, the abstraction case immediately follows from the *i.h.*, and the left application case follows from the reinforced hypothesis that the left subterm is not an abstraction. We treat the case of evaluation in the right subterm of an application, that is the delicate one, where the shrinking predicate plays a crucial role.

The rule is

$$\frac{\mathtt{neutral}_{lo}(u) \quad q \rightarrow_{lo} m}{t = uq \rightarrow_{lo} um = p}$$

There are two cases for the last rule of the derivation Φ:

- app_b *rule*:

$$\dfrac{\Phi_u \triangleright_{lo} \Gamma_u \vdash^{(b_u,r_u)} u : [B_i]_{i\in I} \rightarrow A \qquad \dfrac{(\Phi_{q_i} \triangleright_{lo} \Gamma_q^i \vdash^{(b_i,r_i)} q : B_i)_{i\in I}}{\uplus_{i\in I} \Gamma_q^i \vdash^{(+_{i\in I} b_i, +_{i\in I} r_i)} q : [B_i]_{i\in I}}\ \mathsf{many}}{\Gamma = \Gamma_u \uplus_{i\in I} \Gamma_q^i \vdash^{(b_u +_{i\in I} b_i + 1, r_u +_{i\in I} r_i)} uq : A}\ \mathsf{app}_b$$

The *i.h.* applied to each Φ_{q_i} and $q \rightarrow_{lo} m$ gives Φ_{m_i} such that $\Phi_{m_i} \triangleright_{lo} \Gamma_q^i \vdash^{(b'_i,r_i)} m : B_i$ with $b'_i \leq b_i$ and $|\Phi_{m_i}| \leq |\Phi_{q_i}|$. Then the derivation Ψ given by

$$\dfrac{\Phi_u \triangleright_{lo} \Gamma_u \vdash^{(b_u,r_u)} u : [B_i]_{i\in I} \rightarrow A \qquad \dfrac{(\Phi_{m_i} \triangleright_{lo} \Gamma_q^i \vdash^{(b'_i,r_i)} m : B_i)_{i\in I}}{\uplus_{i\in I} \Gamma_q^i \vdash^{(+_{i\in I} b'_i, +_{i\in I} r_i)} m : [B_i]_{i\in I}}}{\Gamma = \Gamma_u \uplus_{i\in I} \Gamma_q^i \vdash^{(b_u +_{i\in I} b'_i + 1, r_u +_{i\in I} r_i)} um : A}\ \mathsf{app}_b$$

verifies the statement. Let $M := [B_i]_{i\in I}$.

Shrinking: we have to show two things, that the multi-set M is non-empty and that, in order to apply the *i.h.*, the derivations Φ_{m_i} in the right premiss of the rule are all shrinking. Since Φ is shrinking, Γ is co-shrinking, and so are Γ_u and all the Γ_q^i. The hypothesis $\mathtt{neutral}_{lo}(u)$ gives $\mathtt{neutral}_{hd}(u)$ (which is a weaker predicate). Then, $\mathtt{neutral}_{hd}(u)$ and Γ_u co-shrinking allow to apply the occurrences spreading on neutral terms (Lemma 5.6), obtaining $M \rightarrow A$ is co-shrinking, and so M is shrinking. Then $I \neq \emptyset$ and every B_i is shrinking and so every premiss Φ_{m_i} is shrinking.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Then by *i.h.* $b'_i < b_i$ and $|\Phi_{m_i}| < |\Phi_{q_i}|$ for every $i \in I$, and so $b' = b_u +_{i \in I} b'_i + 1 < b_u +_{i \in I} b_i + 1 = b$, and $|\Psi| = |\Phi_u| +_{i \in I} |\Phi_{m_i}| + 1 <_{i.h.} |\Phi_u| +_{i \in I} |\Phi_{q_i}| + 1 = |\Phi|$, as required.

Unitary shrinking: Since Γ is unitary co-shrinking, then Γ_u is unitary co-shrinking. This, together with $\mathtt{neutral}_{lo}(u)$ allows to apply Lemma 5.6, then $M \to A$ is unitary co-shrinking and so M is unitary shrinking. Therefore, M is a singleton and B_1 is unitary shrinking. Moreover, Γ_q^1 is also unitary co-shrinking so that Φ_{q_1} is unitary shrinking.

Then by *i.h.* $b'_1 = b_1 - 1$ and $|\Phi_{m_1}| < |\Phi_{q_1}|$, and so $b' = b_u + b'_1 + 1 = b_u + b_1 - 1 + 1 = b - 1$, and $|\Psi| = |\Phi_u| + |\Phi_{m_1}| + 1 <_{i.h.} |\Phi_u| + |\Phi_{q_1}| + 1 = |\Phi|$, as required.

- app_r^{lo} *rule*:

$$\dfrac{\Phi_u \rhd_{lo} \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{neutral} \quad \Phi_q \rhd_{lo} \Gamma_q \vdash^{(b_q,r_q)} q : \mathtt{tight}}{\Gamma = \Gamma_u \uplus \Gamma_q \vdash^{(b_u+b_q,r_u+r_q+1)} uq : \mathtt{neutral}}\ \mathsf{app}_r^{lo}$$

with $b = b_u + b_q$ and $r = r_u + r_q + 1$. The *i.h.* applied to Φ_q and $q \to_{lo} m$ gives Φ_m such that $\Phi_m \rhd_{lo} \Gamma_q \vdash^{(b_m,r_q)} m : \mathtt{tight}$ with $b_m \leq b_q$ and so $|\Phi_m| \leq |\Phi_q|$. Then the derivation Ψ given by:

$$\dfrac{\Phi_u \rhd_{lo} \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{neutral} \quad \Phi_m \rhd_{lo} \Gamma_q \vdash^{(b_m,r_q)} m : \mathtt{tight}}{\Gamma = \Gamma_u \uplus \Gamma_q \vdash^{(b_u+b_m,r_u+r_q+1)} uq : \mathtt{neutral}}\ \mathsf{app}_r^{lo}$$

verifies the statement.

Shrinking: if Φ is shrinking then Γ_q is co-shrinking, and so is Φ_q (because tight types are shrinking). By *i.h.* then $b_m < b_q$ and $|\Phi_m| < |\Phi_q|$, and so $b' = b_u + b_m < b_u + b_q = b$, and $|\Psi| = |\Phi_u| + |\Phi_m| + 1 < |\Phi_u| + |\Phi_q| + 1 = |\Phi|$, as required.

Unitary shrinking: if Φ is unitary shrinking then Γ_q is unitary co-shrinking, and so is Φ_q (because tight types are unitary shrinking). By *i.h.* then $b_m = b_q - 1$ and $|\Phi_m| < |\Phi_q|$, and so $b' = b_u + b_m = b_u + b_q - 1 = b - 1$, and $|\Psi| = |\Phi_u| + |\Phi_m| + 1 < |\Phi_u| + |\Phi_q| + 1 = |\Phi|$, as required. □

Theorem 5.10 (Shrinking correctness). *Let $\Phi \rhd_{lo} \Gamma \vdash^{(b,r)} t : A$ be a shrinking derivation. Then there exists p such that $\mathtt{normal}_{lo}(p)$ and $k \leq b$ such that*

1. Steps*: $t \to_{lo}$ -evaluates to p in k steps, i.e. $t \to_{lo}^{k} p$;*
2. Size bound*: $|p|_{lo} + k \leq |\Phi|$;*

Moreover, if Φ is traditional then $|p|_{lo} \leq \#(\Gamma) + \#(A)$ and $|p|_\lambda \leq \#^N(\Gamma) + \#^P(A)$, and if Φ is also unitary shrinking then $|p|_\lambda = b - k$.

Proof By induction on $|\Phi|$. If t is a $\to_{lo}$ normal form – that covers the base case $|\Phi| = 1$, for which t is necessarily a variable – then we take $p := t$ and $k := 0$. The first statement trivially holds. The second statement holds by Proposition 5.7. The *moreover* part: if Φ is traditional $|p|_{lo} \leq \#(\Gamma) + \#(A)$ holds by Proposition 5.8.2 and $|p|_\lambda \leq \#^N(\Gamma) + \#^P(A)$ is obtained by composing $|p|_\lambda \leq b$, given by Proposition 5.7.1, and $b \leq \#^N(\Gamma) + \#^P(A)$,

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

given by Proposition 5.8.2, and if Φ is unitary shrinking then $|p|_\lambda = b$ is given by Proposition 5.7.2.

If instead $t \to_{lo} u$ then by shrinking subject reduction (Proposition 5.9) there is a shrinking derivation $\Psi \rhd_{lo} \Gamma \vdash^{(b',r)} u : A$ such that $b' < b$ and $|\Psi| < |\Phi|$. By *i.h.*, there exists a $\to_{lo}$ normal form p and a natural number $k' \leq b'$ satisfying the statement with respect to u, so in particular $|p|_{lo} + k' \leq |\Psi|_{lo}$. Let $k := k' + 1$. Then:

1. *Steps*: $t \to_{lo}^{k} p$ because $t \to_{lo} u \to_{lo}^{k'} p$. Moreover, $k = k' + 1 \leq_{i.h.} b' + 1 \leq b$.
2. *Size bound*: $|p|_{lo} + k = |p|_{lo} + k' + 1 \leq_{i.h.} |\Psi| + 1 \leq |\Phi|$.

The *moreover* part mainly follows from the *i.h.*: only the relationship $|p|_\lambda = b - k$ is not immediate, but if Φ is unitary shrinking then $|p|_\lambda =_{i.h.} b' - k'$ and $b = b' + 1$ by shrinking subject reduction and, since $k = k' + 1$, then $|p|_\lambda = b - k$ holds. □

C.2 Shrinking completeness

Proposition 5.11 (Neutral and normal terms have minimal traditional shrinking typings)**.**

1. *If* $\mathtt{neutral}_{lo}(t)$ *then for every unitary co-shrinking type* A *there exists a traditional derivation* $\Phi \rhd_{lo} \Gamma \vdash^{(b,0)} t : A$ *such that* Γ *is unitary co-shrinking,* $\#(A) + |t|_{lo} = \#(\Gamma)$*, and* $\#^N(A) + b = \#^N(\Gamma)$.
2. *If* $\mathtt{normal}_{lo}(t)$ *then there exists a traditional unitary shrinking derivation* $\Phi \rhd_{lo} \Gamma \vdash^{(b,0)} t : A$ *such that* $|t|_{lo} = \#(\Gamma) + \#(A)$ *and* $b = \#^N(\Gamma) + \#^P(A)$.

Proof By mutual induction on $\mathtt{neutral}_{lo}(t)$ and $\mathtt{normal}_{lo}(t)$.

1. Cases of $\mathtt{neutral}_{lo}(t)$:
   - *Variable*, i.e. $t = x$. Then
   $$\frac{}{x : [A] \vdash^{(0,0)} x : A}\ \mathsf{ax}$$
   whose type context $x : [A]$ is unitary shrinking because A is unitary co-shrinking by hypothesis. We have $\#(A) + |x|_{lo} = \#([A]) + 0 = \#(x : [A])$ and $\#^N(A) + b = \#^N([A]) + 0 = \#^N(x : [A])$.
   - *Application*, i.e. $t = pu$ with $\mathtt{neutral}_{lo}(p)$ and $\mathtt{normal}_{lo}(u)$. By *i.h.* (point 2) applied to u, there exists a traditional unitary shrinking typing $\Phi_u \rhd_{lo} \Gamma_u \vdash^{(b_u,0)} u : B$ with $|u|_{lo} = \#(\Gamma_u) + \#(B)$ and $b_u = \#^N(\Gamma_u) + \#^P(B)$.

     Now, consider the type $[B] \to A$, that is unitary co-shrinking, because A is unitary co-shrinking and B is unitary shrinking. By *i.h.* (point 1) applied to p and $[B] \to A$ there exists a traditional typing $\Phi_p \rhd_{lo} \Gamma_p \vdash^{(b_p,0)} p : [B] \to A$ such that Γ_p is unitary co-shrinking and satisfying $\#([B] \to A) + |p|_{lo} = \#(\Gamma_p)$ and $\#^N([B] \to A) + b_p = \#^N(\Gamma_p)$.

     Then the derivation Φ built as follows:
   $$\frac{\Phi_p \rhd_{lo} \Gamma_p \vdash^{(b_p,0)} p : [B] \to A \qquad \dfrac{\Phi_u \rhd_{lo} \Gamma_u \vdash^{(b_u,0)} u : B}{\Gamma_u \vdash^{(b_u,0)} u : [B]}\ \mathsf{many}}{\Gamma_p \uplus \Gamma_u \vdash^{(b_p+b_u,0)} pu : A}\ \mathsf{app}_b$$

It is traditional and such that its type context is unitary co-shrinking. Moreover,

$$\begin{array}{rcl} \#(A) + |t|_{lo} & = & \#(A) + |p|_{lo} + |u|_{lo} + 1 \\ & =_{i.h.\text{ on } u} & \#(A) + |p|_{lo} + \#(\Gamma_u) + \#(B) + 1 \\ & = & \#(\Gamma_u) + \#([B] \to A) + |p|_{lo} \\ & =_{i.h.\text{ on } p} & \#(\Gamma_u) + \#(\Gamma_p) \\ & = & \#(\Gamma_p \uplus \Gamma_u) \end{array}$$

and

$$\begin{array}{rcl} \#^N(A) + b_p + b_u & =_{i.h.\text{ on } u} & \#^N(A) + b_p + \#^N(\Gamma_u) + \#^P(B) \\ & = & \#^N(\Gamma_u) + \#^N([B] \to A) + b_p \\ & =_{i.h.\text{ on } p} & \#^N(\Gamma_u) + \#^N(\Gamma_p) \\ & = & \#^N(\Gamma_p \uplus \Gamma_u) \end{array}$$

2. Cases of $\texttt{normal}_{lo}(t)$:

   a. $\texttt{neutral}_{lo}(t)$. By *i.h.* (point 1), *for every* unitary co-shrinking type A there exists a traditional typing $\Phi \triangleright_{lo} \Gamma \vdash^{(b,0)} t : A$ such that Γ is unitary co-shrinking. It is then enough to pick $A := X$, that is both unitary shrinking and unitary co-shrinking, so that Φ is unitary shrinking, $\#(A) = 0$, and the statement trivially holds, because then $|t|_{lo} = \#(A) + |t|_{lo} =_{i.h.} \#(\Gamma) = \#(\Gamma) + \#(A)$. Moreover, $\#^P(A) = \#^N(A) = 0$, so that by *i.h.* $\#^N(A) + b = \#^N(\Gamma)$, which is equivalent to $b = \#^N(\Gamma) + \#^P(A)$, as required.

   b. *Abstraction*, i.e. $t = \lambda y.p$ and $\texttt{normal}_{lo}(p)$. By *i.h.* (point 2), there exists a unitary shrinking traditional typing $\Phi_p \triangleright_{lo} \Gamma_p \vdash^{(b_p,0)} p : B$ with $|p|_{lo} = \#(\Gamma_p) + \#(B)$.

   Then let $y : M$ (M possibly $[\,]$) the declaration of y in Γ_p and set Γ be Γ_p without $y : M$. Then let Φ be the derivation

$$\dfrac{\Phi_p \triangleright_{lo} y : M; \Gamma \vdash^{(b_p,0)} p : B}{\Gamma \vdash^{(b_p+1,0)} \lambda y.p : M \to B}\ \mathsf{fun}_b$$

   which is traditional and unitary shrinking because Φ_p is. We have

$$\begin{array}{rcl} |\lambda y.p|_{lo} & = & |p|_{lo} + 1 \\ & =_{i.h.} & \#(y : M; \Gamma) + \#(B) + 1 \\ & = & \#(\Gamma) + \#(M) + \#(B) + 1 \\ & = & \#(\Gamma) + \#(M \to B) \end{array}$$

   and

$$\begin{array}{rcl} b_p + 1 & =_{i.h.} & \#^N(y : M; \Gamma) + \#^P(B) + 1 \\ & = & \#^N(\Gamma) + \#^N(M) + \#^P(B) + 1 \\ & = & \#^N(\Gamma) + \#^P(M \to B) \end{array}$$

□

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proposition 5.12 (Shrinking subject expansion)**.** *If $t \to_{lo} p$ and $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} p : A$ then there exists Ψ such that $\Psi \triangleright_{lo} \Gamma \vdash^{(b',r)} t : A$ with $b' \geq b$. Moreover, if Φ is shrinking (resp. unitary shrinking) then $b' \geq b+1$ (resp. $b' = b+1$) and $|\Psi| > |\Phi|$, and if Φ is traditional then Ψ is traditional.*

Proof The first part (without the shrinking hypothesis) is an easy induction on $t \to_{lo} p$. The part about shrinking/unitary shrinking typings is also by induction on $t \to_{lo} p$, but it requires a strengthened statement, along the same lines of the proof for the tight case and of subject reduction:

1. If $t \to_{lo} p$, $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} p : A$, Γ is co-shrinking and either A is shrinking or $\neg\mathtt{abs}_{lo}(t)$, then there exists a typing $\Psi \triangleright_{lo} \Gamma \vdash^{(b+1,r)} t : A$ such that $b' \geq b+1$ and $|\Psi| > |\Phi|$.
2. If $t \to_{lo} p$, $\Phi \triangleright_{lo} \Gamma \vdash^{(b,r)} p : A$, Γ is unitary co-shrinking, and either A is unitary shrinking or $\neg\mathtt{abs}_{lo}(t)$, then there exists a typing $\Psi \triangleright_{lo} \Gamma \vdash^{(b+1,r)} t : A$ such that $b' = b+1$ and $|\Psi| > |\Phi|$.

The cases of evaluation at top level, under abstraction, and in the left subterm of an application follows exactly the schema of the tight case: at top level the tight/shrinking hypothesis does not play any role, the abstraction case immediately follows from the *i.h.*, and the left application case follows from the reinforced hypothesis that the left subterm is not an abstraction. We treat the case of evaluation in the right subterm of an application, that is the delicate one, where shrinkness plays a crucial role.

The rule is:

$$\frac{\mathtt{neutral}_{lo}(u) \quad q \to_{lo} m}{t = uq \to_{lo} um = p}$$

There are two cases for the last rule of the derivation Φ:

- app_b *rule*:

$$\frac{\Phi_u \triangleright_{lo} \Pi \vdash^{(b_u,r_u)} u : [B_i]_{i\in I} \to A \qquad \dfrac{(\Phi_{m_i} \triangleright_{lo} \Delta_i \vdash^{(b_i,r_i)} m : B_i)_{i\in I}}{\uplus_{i\in I}\Delta_i \vdash^{(+_{i\in I}b_i, +_{i\in I}r_i)} m : [B_i]_{i\in I}}\,\mathsf{many}}{\Gamma = \Pi \uplus_{i\in I} \Delta_i \vdash^{(b_u +_{i\in I} b_i, r_u +_{i\in I} r_i)} um : A}\,\mathsf{app}_b$$

  The *i.h.* applied to each Φ_{m_i} and $q \to_{lo} m$ gives Φ_{q_i} such that $\Phi_{q_i} \triangleright_{lo} \Delta_i \vdash^{(b'_i,r_i)} q : B_i$ with $b'_i \geq b_i$ and $|\Phi_{q_i}| \geq |\Phi_{m_i}|$. Then the derivation Ψ given by:

$$\frac{\Phi_u \triangleright_{lo} \Pi \vdash^{(b_u,r_u)} u : [B_i]_{i\in I} \to A \qquad \dfrac{(\Psi_{q_i} \triangleright_{lo} \Delta_i \vdash^{(b'_i,r_i)} q : B_i)_{i\in I}}{\uplus_{i\in I}\Delta_i \vdash^{(+_{i\in I}b'_i, +_{i\in I}r_i)} q : [B_i]_{i\in I}}}{\Gamma = \Pi +_{i\in I} \Delta_i \vdash^{(b_u +_{i\in I} b'_i, r_u +_{i\in I} r_i)} uq : A}\,\mathsf{app}_b$$

  verifies the statement. The statement $|\Psi| > |\Phi|$ is a straightforward consequence of the *i.h.*

  Shrinking/Unitary shrinking: exactly the same reasoning used for shrinking/unitary shrinking subject reduction proves that Φ_u is shrinking/unitary shrinking, I is non-empty, and the B_i are all shrinking/unitary shrinking. The *i.h.* then provides $b'_i \geq b_i + 1$ (resp. $b'_i = b_i + 1$) for every $i \in I$, from which the property follows:

- app_r^{lo} *rule*:

$$\dfrac{\Phi_u \triangleright_{lo} \; \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{neutral} \quad \Phi_m \triangleright_{lo} \; \Gamma_m \vdash^{(b_m,r_m)} m : \mathtt{tight}}{\Gamma = \Gamma_u \uplus \Gamma_m \vdash^{(b_u+b_m,r_u+r_m+1)} um : \mathtt{neutral}} \; \mathsf{app}_r^{lo}$$

with $b = b_u + b_m$ and $r = r_u + r_m + 1$. The *i.h.* applied to Φ_m and $q \rightarrow_{lo} m$ gives Ψ_q such that $\Psi_q \triangleright_{lo} \; \Gamma_q \vdash^{(b_m,r_q)} m : \mathtt{tight}$ with $b_q \geq b_m$ and so $|\Psi_q| \geq |\Phi_m|$. Then the derivation Ψ given by:

$$\dfrac{\Phi_u \triangleright_{lo} \; \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{neutral} \quad \Psi_q \triangleright_{lo} \; \Gamma_q \vdash^{(b_q,r_m)} q : \mathtt{tight}}{\Gamma = \Gamma_u \uplus \Gamma_q \vdash^{(b_u+b_q,r_u+r_m+1)} uq : \mathtt{neutral}} \; \mathsf{app}_r^{lo}$$

verifies the statement. The statement $|\Psi| > |\Phi|$ is a straightforward consequence of the *i.h.*

Shrinking/Unitary shrinking: if Φ is shrinking/unitary shrinking then Γ_m is co-shrinking/unitary co-shrinking, and Φ_m is shrinking/unitary shrinking (because tight types are shrinking and unitary shrinking). By *i.h.* $b_q \geq b_m + 1$ (resp. $b_q = b_m + 1$), and so $b' = b_u + b_q \geq b_u + b_m + 1 = b + 1$ (resp. $b' = b_u + b_q = b_u + b_m + 1 = b + 1$), as required. $\square$

Theorem 5.13 (Shrinking completeness). *Let $t \rightarrow_{lo}^{k} p$ with p such that $\mathtt{normal}_{lo}(p)$. Then there exists a traditional unitary shrinking typing $\Phi \triangleright_{lo} \; \Gamma \vdash^{(b,0)} t : A$ such that $k = b - \#^N(\Gamma) - \#^P(A)$ and $|p|_{lo} = \#(\Gamma) + \#(A)$.*

Proof By induction on k. If $k = 0$ the statement is given by the existence of traditional unitary shrinking typings for $\rightarrow_{lo}$ -normal terms (Proposition 5.11), for which $k = 0$ and $b = \#^N(\Gamma) + \#^P(A)$. Let $k > 0$ and $t \rightarrow_{lo} u \rightarrow_{lo}^{k-1} p$. By *i.h.*, there exists a traditional unitary shrinking typing derivation $\Psi \triangleright_{lo} \; \Gamma \vdash^{(b',0)} u : A$ with $k - 1 = b' - \#^N(\Gamma) - \#^P(A)$, and $|p|_{lo} = \#(\Gamma) + \#(A)$. By shrinking subject expansion (Proposition 5.12) there exists a traditional typing derivation Φ of t with the same types in the ending judgement of Ψ – then Φ is unitary shrinking and $|p|_{lo} = \#(\Gamma) + \#(A)$ still holds – and with indices $(b' + 1, 0)$. Then $k = k - 1 + 1 =_{i.h.} b' - \#^N(\Gamma) - \#^P(A) + 1 = b - \#^N(\Gamma) - \#^P(A)$. $\square$

Appendix D Maximal evaluation

D.1 Tight correctness

Proposition 7.4 (Properties of typings for normal forms). *Given $\Phi \triangleright_{max} \; \Gamma \vdash^{(b,r)} t : A$ with $\mathtt{normal}_{max}(t)$,*

1. Size bound*: $|t| \leq |\Phi|$.*
2. Tight indices*: if Φ is tight then $b = 0$ and $r = |t|$.*
3. Neutrality*: if $A = \mathtt{neutral}$ then $\mathtt{neutral}_{max}(t)$.*

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proof By induction on t. Note that $\mathtt{neutral}_{max}$ implies $\mathtt{normal}_{max}$ and so we can apply the *i.h.* when $\mathtt{neutral}_{max}$ holds on some subterm of t. If $\mathtt{normal}_{max}(t)$ because $\mathtt{neutral}_{max}(t)$ there are two cases:

- *Variable*, i.e. $t = x$. Then Φ has the following form and evidently verifies all the points of the statement:

$$\frac{}{x : [A] \vdash^{(0,0)} x : A}\ \mathsf{ax}$$

- *Application*, i.e. $t = pu$, $\mathtt{neutral}_{max}(p)$ and $\mathtt{normal}_{max}(u)$. Cases of the last rule of Φ:

  – app_b *rule*:

$$\frac{\Phi_p \rhd_{max}\ \Gamma_p \vdash^{(b_p,r_p)} p : M \to A \qquad \Phi_u \rhd_{max}\ \Gamma_u \vdash^{(b_u,r_u)} u : M}{\Gamma_p \uplus \Gamma_u \vdash^{(b_p+b_u+1,r_p+r_u)} pu : A}\ \mathsf{app}_b$$

  with $b = b_p + b_u + 1$, $r = r_p + r_u$, and $\Gamma = \Gamma_p \uplus \Gamma_u$.

  1. *Size bound*: Φ_u end with rule $(\mathsf{many}_{>0})$ or (none), so in both cases there are $n \geq 1$ subderivations $(\Phi_u^i)_{1 \leq i \leq n}$ typing u such that $|u|_{max} \leq_{i.h.} |\Phi_u^i|_{max} \leq |\Phi_u|_{max}$. The *i.h.* gives $|p|_{max} \leq |\Phi_p|_{max}$. We conclude $|t|_{max} = |p|_{max} + |u|_{max} + 1 \leq |\Phi_p|_{max} + |\Phi_u|_{max} + 1 = |\Phi|_{max}$.
  2. *Tight indices*: Lemma 7.3 shows this case to be impossible, as $\mathtt{normal}_{max}(pu)$ means $\mathtt{neutral}_{max}(pu)$, which implies $\mathtt{neutral}_{hd}(pu)$.
  3. *Neutrality*: $\mathtt{neutral}_{max}(t)$ holds by hypothesis.

  – app_r^{lo} *rule*:

$$\frac{\Phi_p \rhd_{lo}\ \Gamma_p \vdash^{(b_p,r_p)} p : \mathtt{neutral} \qquad \Phi_u \rhd_{lo}\ \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{tight}}{\Gamma_p \uplus \Gamma_u \vdash^{(b_p+b_u,r_p+r_u+1)} pu : \mathtt{neutral}}\ \mathsf{app}_r^{lo}$$

  with $b = b_p + b_u$, $r = r_p + r_u + 1$, and $\Gamma = \Gamma_p \uplus \Gamma_u$.

  1. *Size bound*: by *i.h.* $|p|_{lo} \leq |\Phi_p|_{lo}$ and $|u|_{max} \leq |\Phi_u|_{max}$. Then $|t|_{max} = |p|_{max} + |u|_{max} + 1 \leq_{i.h.} |\Phi_p|_{max} + |\Phi_u|_{max} + 1 = |\Phi|_{max}$.
  2. *Tight indices*: if Φ is tight, then Φ_p and Φ_u are tight and $r_p = |p|_{max}$ and $b_p = 0$, and $r_u = |u|_{max}$ and $b_u = 0$. Then, $r = r_p + r_u + 1 =_{i.h.} |p|_{max} + |u|_{max} + 1 = |pu|_{max} = |t|_{max}$ and $b = b_p + b_u = 0 + 0 = 0$.
  3. *Neutrality*: $\mathtt{neutral}_{lo}(t)$ holds by hypothesis.

Now, there is only one case left for $\mathtt{normal}_{max}(t)$:

- *Abstraction*, i.e. $t = \lambda x.p$ and $\mathtt{normal}_{max}(t)$ because $\mathtt{normal}_{max}(p)$. Cases of the last rule of Φ:

  – fun_b *rule*:

$$\frac{\Phi_p \rhd_S\ \Gamma; x : M \vdash^{(b_p,r)} p : A}{\Gamma \vdash^{(b_p+1,r)} \lambda x.p : M \to A}\ \mathsf{fun}_b$$

with $b = b_p + 1$.

1. *Size bound*: Then, $|t|_{max} = |p|_{max} + 1 \leq_{i.h.} |\Phi_p|_{max} + 1 = |\Phi|_{max}$.
2. *Tight indices*: Φ is not tight, so the statement trivially holds.
3. *Neutrality*: $A \neq \mathtt{neutral}$, so the statement trivially holds.

– fun_r *rule*:

$$\frac{\Phi_p \rhd \Gamma; x : \mathtt{Tight} \vdash^{(b,r_p)} p : \mathtt{tight}}{\Gamma \vdash^{(b,r_p+1)} \lambda x.p : \mathtt{abs}}\ \mathsf{fun}_r$$

with $r = r_p + 1$.

1. *Size bound*: Then, $|t|_{max} = |p|_{max} + 1 \leq_{i.h.} |\Phi_p|_{max} + 1 = |\Phi|_{max}$.
2. *Tight indices*: if Φ is tight, then Φ_p is tight and by *i.h.* $r_p = |p|_{max}$ and $b = 0$. Then, $r = r_p + 1 =_{i.h.} |p|_{max} + 1 = |t|_{max}$.
3. *Neutrality*: $A \neq \mathtt{neutral}$, so the statement trivially holds. □

Proposition 7.7 (Quantitative tight subject reduction for *max*). *If $\Phi \rhd_{max}\ \Gamma \vdash^{(b,r)} t : A$ is max-tight and $t \xrightarrow{e}_{max} p$, then there exist $\Gamma' \sqsubseteq \Gamma$ and an max-tight typing Ψ such that $\Psi \rhd_{max}\ \Gamma' \vdash^{(b-1,r-e)} p : A$ and $|\Phi| > |\Psi|$.*

Proof We prove, by induction on $t \xrightarrow{e}_{max} p$, the stronger statement:

Assume $t \xrightarrow{e}_{max} p$, $\Phi \rhd_{max}\ \Gamma \vdash^{(b,r)} t : A$ is garbage-tight, $\mathtt{tight}(\Gamma)$, and either $\mathtt{tight}(A)$ or $\neg\mathtt{abs}_{max}(t)$. Then there exist Γ' and a garbage-tight typing $\Psi \rhd_{max}\ \Gamma' \vdash^{(b-1,r-e)} p : A$ such that $\mathtt{tight}(\Gamma')$.

- Non-erasing top-level step:

$$\frac{x \in \mathtt{fv}(u)}{(\lambda x.u)q \xrightarrow{0}_{max} u\{x{\leftarrow}q\}}$$

Assume $\Phi \rhd_{max}\ \Gamma \vdash^{(b,r)} (\lambda x.u)q : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule app_b, the derivation of its premiss for $\lambda x.u$ must end with fun_b. Hence, there are two garbage-tight derivations $\Phi_u \rhd_{max}\ \Gamma_u; x : M \vdash^{(b_u,r_u)} u : A$ and $\Phi_p \rhd_{max}\ \Gamma_p \vdash^{(b_p,r_p)} p : M$, with $(b,r) = (b_u + b_q + 1, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_p$. Moreover, by hypothesis $x \in \mathtt{fv}(u)$, and so $M \neq [\,]$ by relevance (Lemma 7.2). Then, the substitution lemma (Lemma 7.6) gives a garbage-tight derivation $\Psi \rhd_{max}\ \Gamma \vdash^{(b_u+b_q,r_u+r_q)} u\{x{\leftarrow}q\} : A$ such that $|\Psi| = |\Phi_u| + |\Phi_q| - |M| < |\Phi_u| + |\Phi_q| + 2 = |\Phi|$.
- Erasing top-level step:

$$\frac{x \notin \mathtt{fv}(u) \qquad \mathtt{normal}_{max}(q)}{(\lambda x.u)q \xrightarrow{|q|_{max}}_{max} u}$$

Assume $\Phi \rhd_{max}\ \Gamma \vdash^{(b,r)} (\lambda x.u)q : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule app_b, and the derivation of its premiss for $(\lambda x.u)$ must end with fun_b. Moreover, since $x \notin \mathtt{fv}(u)$, then by relevance (Lemma 7.2) the derivation of

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

its premiss q must end with rule none:

$$\dfrac{\dfrac{\Phi_u \rhd_{max} \; \Gamma_u \vdash^{(b_u,r_u)} u : A}{\Gamma_u \vdash^{(b_u+1,r_u)} \lambda x.u : [\,] \to A}\,\mathsf{fun}_b \qquad \dfrac{\Phi_q \rhd_{max} \; \Gamma_q \vdash^{(b_q,r_q)} q : A_q}{\Gamma_q \vdash^{(b_q,r_q)} q : [\,]}\,\mathsf{none}}{\Gamma_u \uplus \Gamma_q \vdash^{(b_u+b_q+1,r_u+r_q)} (\lambda x.u)q : A}\,\mathsf{app}_b$$

with $(b, r) = (b_u + b_q + 1, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_p$. Since Φ is garbage-tight, then Γ_q is tight and A_q must be $\mathtt{tight}$, and since $\mathtt{normal}_{max}(q)$, we can apply the tight indices property of normal forms (Proposition 7.4) and obtain $(b_q, r_q) = (0, |q|_{max})$, so that $(b_u, r_u) = (b - 1, r - |q|_{max})$. Since $\mathtt{tight}(\Gamma_u \uplus \Gamma_q)$ we have $\mathtt{tight}(\Gamma_u)$, so Φ_u is the desired garbage-tight derivation. Moreover, $|\Phi_u| < |\Phi_u| + |\Phi_q| + 2 = |\Phi|$.

- Rule

$$\dfrac{t \xrightarrow{e}_{max} p}{\lambda x.t \xrightarrow{e}_{max} \lambda x.p}$$

Assume $\Phi \rhd_{max} \; \Gamma \vdash^{(b,r)} \lambda x.t : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. Since $\mathtt{abs}_{max}(\lambda x.t)$ we must have hypothesis $\mathtt{tight}(A)$, then Φ must necessarily finish with rule fun_r and there is a subderivation $\Phi_t \rhd_{max} \; \Gamma; x : \mathtt{Tight} \vdash^{(b,r-1)} t : \mathtt{tight}$. As Φ_t is garbage-tight and $\mathtt{tight}(\Gamma; x : \mathtt{Tight})$ we can apply the *i.h.* and get $\Phi_p \rhd_{max} \; \Gamma'; x : \mathtt{Tight?} \vdash^{(b-1,r-1-e)} p : \mathtt{tight}$ where $x : \mathtt{Tight?}$ means $x : \mathtt{Tight}$ or $x : [\,]$ and $\Gamma' \sqsubseteq \Gamma$, Φ_p is garbage-tight and $\mathtt{tight}(\Gamma'; x : \mathtt{Tight?})$. Then we construct the derivation Ψ:

$$\dfrac{\Phi_p \rhd_{max} \; \Gamma'; x : \mathtt{Tight?} \vdash^{(b-1,r-1-e)} p : \mathtt{tight}}{\Gamma' \vdash^{(b-1,r-e)} \lambda x.p : A}\,\mathsf{fun}_r$$

Then $\mathtt{tight}(\Gamma')$ and thus Ψ is garbage-tight. We conclude $|\Phi| = |\Phi_t| + 1 > |\Phi_p| + 1 = |\Psi|$ thanks to the *i.h.* $|\Phi_t| > |\Phi_p|$.

- Rule

$$\dfrac{\neg\mathtt{abs}_{max}(t) \quad t \xrightarrow{e}_{max} p}{tu \xrightarrow{e}_{max} pu}$$

Assume $\Phi \rhd_{max} \; \Gamma \vdash^{(b,r)} tu : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule app_b or app_r^{lo}, and therefore there are two garbage-tight derivations $\Phi_t \rhd_{max} \; \Gamma_t \vdash^{(b_t,r_t)} t : A_t$ and $\Phi_u \rhd_{max} \; \Gamma_u \vdash^{(b_u,r_u)} u : A_u$, for some types A_t and A_u, with $\Gamma = \Gamma_t \uplus \Gamma_u$. Since $\mathtt{tight}(\Gamma)$ we have $\mathtt{tight}(\Gamma_t)$ and $\mathtt{tight}(\Gamma_u)$. Since $\neg\mathtt{abs}_{max}(t)$, we can apply the *i.h.* and get the garbage-tight derivation $\Phi_p \rhd_{max} \; \Gamma_p \vdash^{(b_t-1,r_t-e)} p : A_t$, with $\mathtt{tight}(\Gamma_p)$. Then the same rule app_b or app_r^{lo} can be applied to get the garbage-tight derivation $\Psi \rhd_{max} \; \Gamma_p \uplus \Gamma_u \vdash^{(b-1,r-e)} pu : A$, with $\mathtt{tight}(\Gamma_p \uplus \Gamma_u)$. We conclude $|\Phi| = |\Phi_t| + |\Phi_u| + 1 > |\Phi_p| + |\Phi_u| + 1 = |\Psi|$ thanks to the *i.h.* $|\Phi_t| > |\Phi_p|$.

- Rule

$$\dfrac{\mathtt{neutral}_{max}(u) \quad t \xrightarrow{e}_{max} p}{ut \xrightarrow{e}_{max} up}$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} ut : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule $\mathtt{app}_r^{lo}$, since Lemma 7.3 (which applies as $\mathtt{neutral}_{max}(u)$ implies $\mathtt{neutral}_{hd}(ut)$) rules out rule $\mathtt{app}_b$. Hence, there are two garbage-tight derivations $\Phi_u \triangleright_{max} \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{neutral}$ and $\Phi_t \triangleright_{max} \Gamma_t \vdash^{(b_t,r_t)} t : \mathtt{tight}$, with $\Gamma = \Gamma_u \uplus \Gamma_t$, $\mathtt{tight}(\Gamma_u)$ and $\mathtt{tight}(\Gamma_t)$. Furthermore, $A = \mathtt{neutral}$. Therefore we can apply the *i.h.* to get the tight derivation $\Phi_p \triangleright_{max} \Gamma_p \vdash^{(b_t-1,r_t-e)} p : \mathtt{tight}$.
Then $\mathtt{app}_r^{lo}$ can be applied to get the tight derivation $\Psi \triangleright_{max} \Gamma_u \uplus \Gamma_p \vdash^{(b-1,r-e)} up : \mathtt{neutral}$. We conclude $|\Phi| = |\Phi_u| + |\Phi_t| + 1 > |\Phi_u| + |\Phi_p| + 1 = |\Psi|$ thanks to the *i.h.* $|\Phi_t| > |\Phi_p|$.

- Rule

$$\frac{x \notin \mathtt{fv}(u) \quad t \xrightarrow{e}_{max} p}{(\lambda x.u)t \xrightarrow{e}_{max} (\lambda x.u)p}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} (\lambda x.u)t : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule $\mathtt{app}_b$, therefore there are garbage-tight derivations $\Phi_u \triangleright_{max} \Gamma_u \vdash^{(b_u,r_u)} \lambda x.u : [\,] \to A$ and $\Phi_t \triangleright_{max} \Gamma_t \vdash^{(b_t,r_t)} t : \mathtt{tight}$, with $\Gamma = \Gamma_u \uplus \Gamma_t$ and $(b,r) = (b_t + b_u + 1, r_t + r_u)$. We can apply the *i.h.* to get the tight derivation $\Phi_p \triangleright_{max} \Gamma_p \vdash^{(b_t-1,r_t-e)} p : \mathtt{tight}$. Then $\mathtt{app}_b$ can be applied to get the garbage-tight derivation $\Psi \triangleright_{max} \Gamma_u \uplus \Gamma_p \vdash^{(b-1,r-e)} (\lambda x.u)p : A$, with $\mathtt{tight}(\Gamma_u \uplus \Gamma_p)$. We conclude $|\Phi| = |\Phi_u| + |\Phi_t| + 1 > |\Phi_u| + |\Phi_p| + 1 = |\Psi|$ thanks to the *i.h.* $|\Phi_t| > |\Phi_p|$. □

Theorem 7.8 (Tight correctness for *max*-evaluation)**.** *Let $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} t : A$ be a max-tight derivation. Then there is an integer e and a term p such that $\mathtt{normal}_{max}(p)$, $t \xrightarrow{e}{}^{b}_{max} p$ and $|p|_{max} + e = r$. Moreover, if $A = \mathtt{neutral}$ then $\mathtt{neutral}_{max}(p)$.*

Proof By induction on $|\Phi|$. If t is a *max* normal form – that covers the base case $|\Phi| = 1$, for which t is necessarily a variable – then by taking $p := t$, $k := 0$ and $e := 0$ the statement follows from the *tightness* property of tight typings of normal forms (Proposition 7.4.2) – the *moreover* part follows from the *neutrality* property (Proposition 7.4.3). Otherwise, $t \xrightarrow{e'}_{max} u$ and by quantitative subject reduction (Proposition 7.7) there is a derivation $\Psi \triangleright_{max} \Gamma' \vdash^{(b-1,r-e')} u : A$ such that $\Gamma' \sqsubseteq \Gamma$ and $|\Psi| < |\Phi|$. By *i.h.*, there exists p such that $\mathtt{normal}_{max}(p)$ and $u \xrightarrow{e''}{}^{b-1}_{max} p$ and $|p|_{max} + e'' = r - e'$. Just note that $t \xrightarrow{e'+e''}{}^{b}_{max} p$. We conclude by taking $e = e' + e''$ because $|p|_{max} + e' + e'' = r$ as required. □

D.2 Tight completeness

Proposition 7.11 (Quantitative tight subject expansion for *max*)**.** *If $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} p : A$ is max-tight and $t \xrightarrow{e}_{max} p$, then there exist $\Gamma' \sqsupseteq \Gamma$ and a max-tight typing Ψ such that $\Psi \triangleright_{max} \Gamma' \vdash^{(b+1,r+e)} t : A$ and $|\Phi| < |\Psi|$. $\Psi \triangleright_{max} \Gamma' \vdash^{(b+1,r+e)} t : A$ and $|\Phi| < |\Psi|$.*

Proof We prove, by induction on $t \xrightarrow{e}_{max} p$, the stronger statement:

Assume $t \xrightarrow{e}_{max} p$, $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} p : A$ is garbage-tight, $\mathtt{tight}(\Gamma)$, and either $\mathtt{tight}(A)$ or $\neg\mathtt{abs}_{max}(t)$. Then there exist $\Gamma' \sqsupseteq \Gamma$ and a garbage-tight typing $\Psi \triangleright_{max} \Gamma' \vdash^{(b+1,r+e)} t : A$ such that $\mathtt{tight}(\Gamma')$ and $|\Phi| < |\Psi|$. In what follows we treat all the cases, by omitting

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

the details about the decreasigness of size derivations, which are the same appearing in previous subject expansion properties of this paper.

- Rule

$$\frac{x \in \mathtt{fv}(u)}{(\lambda x.u)q \xrightarrow{0}_{max} u\{x \leftarrow q\}}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} u\{x\leftarrow q\} : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. By applying the anti substitution (Lemma 7.10) we obtain the premisses of the following derivation Φ':

$$\dfrac{\dfrac{\Phi_u \triangleright_{max} \Gamma_u, x : M \vdash^{(b_u,r_u)} u : A}{\Gamma_u \vdash^{(b_u+1,r_u)} \lambda x.u : M \to A} \qquad \Phi_q \triangleright_{max} \Gamma_q \vdash^{(b_q,r_q)} q : M}{\Gamma_u \uplus \Gamma_q \vdash^{(b_u+b_q+1,r_u+r_q)} (\lambda x.u)q : A}$$

with $(b,r) = (b_u + b_q, r_u + r_q)$ and $\Gamma = \Gamma_u \uplus \Gamma_q$. Moreover, Φ_u and Φ_q are all garbage-tight, so Φ' is garbage-tight.

- Rule

$$\frac{x \notin \mathtt{fv}(u) \quad \mathtt{normal}_{max}(q)}{(\lambda x.u)q \xrightarrow{|q|_{max}}_{max} u}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} u : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. By applying the existence of tight derivations for normal forms (Proposition 7.9), we obtain the max-tight derivation Φ_q used in the construction of derivation Φ' next:

$$\dfrac{\dfrac{\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} u : A}{\Gamma \vdash^{(b+1,r)} \lambda x.u : [\,] \to A} \qquad \dfrac{\Phi_q \triangleright_{max} \Gamma_q \vdash^{(0,|q|_{max})} q : \mathtt{tight}}{\Gamma_q \vdash^{(0,|q|_{max})} q : [\,]}}{\Gamma \uplus \Gamma_q \vdash^{(b+1,r+|q|_{max})} (\lambda x.u)q : A}$$

Moreover, $\mathtt{tight}(\Gamma \uplus \Gamma_q)$ and Φ' is garbage-tight.

- Rule

$$\frac{t \xrightarrow{e}_{max} p}{\lambda x.t \xrightarrow{e}_{max} \lambda x.p}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} \lambda x.p : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. Since $\mathtt{abs}_{max}(\lambda x.t)$ holds, then we must have $\mathtt{tight}(A)$, and then Φ must finish with rule fun_r which must have a subderivation of the form $\Phi_p \triangleright_{max} \Gamma, x : \mathtt{Tight} \vdash^{(b,r-1)} p : \mathtt{tight}$. The derivation Φ_p is garbage-tight and $\mathtt{tight}(\Gamma, x : \mathtt{Tight})$ holds, then we can apply the *i.h.* and get $\Phi_t \triangleright_{max} \Gamma', x : \mathtt{Tight} \vdash^{(b+1,r-1+e)} t : \mathtt{tight}$, where $\Gamma' \sqsupseteq \Gamma$, Φ_t is garbage-tight and $\mathtt{tight}(\Gamma', x : \mathtt{Tight})$. We construct the following derivation Φ':

$$\frac{\Phi_t \triangleright_{max} \Gamma', x : \mathtt{Tight} \vdash^{(b+1,r-1+e)} t : \mathtt{tight}}{\Gamma' \vdash^{(b+1,r+e)} \lambda x.t : A}\ \mathsf{fun}_r$$

Then Φ' is garbage-tight and $\mathtt{tight}(\Gamma')$.

- Rule

$$\frac{\neg\mathtt{abs}_{max}(t) \quad t \xrightarrow{e}_{max} p}{tu \xrightarrow{e}_{max} pu}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} pu : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule $\mathtt{app}_b$ or $\mathtt{app}_r^{lo}$, and therefore there are two garbage-tight derivations $\Phi_p \triangleright_{max} \Gamma_p \vdash^{(b_p,r_p)} p : A_p$ and $\Phi_u \triangleright_{max} \Gamma_u \vdash^{(b_u,r_u)} u : A_u$, for some types A_p and A_u, with $\Gamma = \Gamma_p \uplus \Gamma_u$. Since $\mathtt{tight}(\Gamma)$ we have $\mathtt{tight}(\Gamma_p)$ and $\mathtt{tight}(\Gamma_u)$. Since $\neg\mathtt{abs}_{max}(t)$, we can apply the *i.h.* and get the garbage-tight derivation $\Phi_t \triangleright_{max} \Gamma_t \vdash^{(b_p+1,r_p+e)} t : A_p$, with $\mathtt{tight}(\Gamma_t)$. Then the same rule $\mathtt{app}_b$ or $\mathtt{app}_r^{lo}$ can be applied to get the garbage-tight derivation $\Phi' \triangleright_{max} \Gamma_t \uplus \Gamma_u \vdash^{(b+1,r+e)} tu : A$, with $\mathtt{tight}(\Gamma_t \uplus \Gamma_u)$.
- Rule

$$\frac{\mathtt{neutral}_{max}(u) \quad t \xrightarrow{e}_{max} p}{ut \xrightarrow{e}_{max} up}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} up : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule $\mathtt{app}_r^{lo}$, since Lemma 7.3 (which applies as $\mathtt{neutral}_{max}(u)$ implies $\mathtt{neutral}_{hd}(up)$) rules out rule $\mathtt{app}_b$. Hence, there are two garbage-tight derivations $\Phi_u \triangleright_{max} \Gamma_u \vdash^{(b_u,r_u)} u : \mathtt{neutral}$ and $\Phi_p \triangleright_{max} \Gamma_p \vdash^{(b_p,r_p)} p : \mathtt{tight}$, with $\Gamma = \Gamma_u \uplus \Gamma_p$, $\mathtt{tight}(\Gamma_u)$ and $\mathtt{tight}(\Gamma_p)$. Furthermore, $A = \mathtt{neutral}$. Therefore we can apply the *i.h.* to get the tight derivation $\Phi_t \triangleright_{max} \Gamma_t \vdash^{(b_p+1,r_p+e)} t : \mathtt{tight}$. Then $\mathtt{app}_r^{lo}$ can be applied to get the tight derivation $\Phi' \triangleright_{max} \Gamma_t \uplus \Gamma_u \vdash^{(b+1,r+e)} ut : \mathtt{neutral}$.
- Rule

$$\frac{x \notin \mathtt{fv}(u) \quad t \xrightarrow{e}_{max} p}{(\lambda x.u)t \xrightarrow{e}_{max} (\lambda x.u)p}$$

Assume $\Phi \triangleright_{max} \Gamma \vdash^{(b,r)} (\lambda x.u)p : A$ is garbage-tight and $\mathtt{tight}(\Gamma)$. The derivation Φ must end with rule $\mathtt{app}_b$, therefore there are garbage-tight derivations $\Phi_u \triangleright_{max} \Gamma_u \vdash^{(b_u,r_u)} \lambda x.u : [\,] \to A$ and $\Phi_p \triangleright_{max} \Gamma_p \vdash^{(b_p,r_p)} p : \mathtt{tight}$, with $\Gamma = \Gamma_u \uplus \Gamma_p$ and $(b,r) = (b_p + b_u + 1, r_p + r_u)$. We can apply the *i.h.* to get the tight derivation $\Phi_t \triangleright_{max} \Gamma_t \vdash^{(b_p+1,r_p+e)} t : \mathtt{tight}$. Then $\mathtt{app}_b$ can be applied to get the garbage-tight derivation $\Phi' \triangleright_{max} \Gamma_t \uplus \Gamma_u \vdash^{(b+1,r+e)} (\lambda x.u)t : A$, with $\mathtt{tight}(\Gamma_t \uplus \Gamma_u)$. □

Theorem 7.12 (Tight completeness for *max*). *If $t \xrightarrow{e}{}^{k}_{max} p$ with $\mathtt{normal}_{max}(p)$, then there exists an max-tight typing $\Phi \triangleright_{max} \Gamma \vdash^{(k,|p|_{max}+e)} t : A$. Moreover, if $\mathtt{neutral}_{max}(p)$ then $A = \mathtt{neutral}$, and if $\mathtt{abs}_{max}(p)$ then $A = \mathtt{abs}$.*

Proof By induction on $t \to^{k}_{max} p$. If $k = 0$ (and thus $e = 0$) the statement is given by the existence of tight typings for $\mathtt{normal}_{max}$ terms (Proposition 7.9), that also provides the *moreover* part. Let $k > 0$ and $t \xrightarrow{e'}_{max} u \xrightarrow{e''}{}^{k-1}_{max} p$, where $e = e' + e''$. By *i.h.*, there exists a max-tight derivation $\Psi \triangleright_{max} \Gamma_u \vdash^{(k-1,|p|_{max}+e'')} u : A$ which also provides the *moreover*

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

part. By subject expansion (Proposition 7.11) there exists a max-tight typing derivation $\Phi \triangleright_{max} \Gamma \vdash^{(k-1+1,|p|_{max}+e''+e')} t : A$ such that $\Gamma \sqsupseteq \Gamma_u$. Then we conclude since the indices $(k, |p|_{max} + e)$ are as expected. □

Appendix E Linear head evaluation

Proposition 8.1 (linear head evaluation system).
$(\Lambda_{\mathtt{lsc}}, \to_{lhd}, \mathtt{neutral}_{lhd}, \mathtt{normal}_{lhd}, \mathtt{abs}_{lhd})$ is an evaluation system.

Proof The determinism of $\to_{lhd}$ is straightforward. We prove here the characterisation of *lhd*-normal terms and *lhd*-neutral terms.

⇒) Let t be $\to_{lhd}$ -normal. Then t has either a free head variable x or a bound head variable. We then refine the general statement as follows:

1. If t is $\to_{lhd}$ -normal and has a free head variable x and is not a (potentially) substituted abstraction, then $\mathtt{neutral}^x_{lhd}(t)$.
2. If t is $\to_{lhd}$ -normal and has a free head variable x and is a (potentially) substituted abstraction, then $\mathtt{normal}^x_{lhd}(t)$.
3. If t is $\to_{lhd}$ -normal has a bound head variable, then $\mathtt{normal}^\#_{lhd}(t)$.

We show simultaneously the three statement by induction on terms.

- If t is a variable, then it corresponds to case (1) and we conclude by rule `lhnvar`.
- If $t = \lambda y.p$, then p is also $\to_{lhd}$ -normal. There are two cases: case (2) or (3).
  If $\lambda y.p$ corresponds to case (2), then $y \neq x$ and p corresponds to case (1) or (2). In the first case the *i.h.* (1) gives that $\mathtt{neutral}^x_{lhd}(p)$ and thus we conclude by rules `lhnno` and then `lhnolamx`. In the second case the *i.h.* (2) gives that $\mathtt{normal}^x_{lhd}(p)$ and we conclude with rule `lhnolamx`.
  If $\lambda y.p$ corresponds to case (3), then either p corresponds to case (3), or p corresponds to cases (1) or (2) with $y = x$. In the first case we get that $\mathtt{normal}^\#_{lhd}(p)$ by the *i.h.* (3) and thus $\mathtt{normal}^\#_{lhd}(\lambda x.p)$ by rule `lhnolam`. In the second case we get that $\mathtt{neutral}^x_{lhd}(p)$ by the *i.h.* (1) (resp. $\mathtt{normal}^x_{lhd}(p)$ by the *i.h.* (2)). We conclude with rules `lhnno` and then `lhnolamx` (resp. `lhnolamx`).
- If $t = pu$, then p is also $\to_{lhd}$ -normal, otherwise rule lhd_λ would apply, and p is not a (potentially) substituted abstraction, otherwise rule $lhd_{\mathtt{m}}$ would apply. The term pu necessarily corresponds to case (1) for some variable x and the same for p. We thus obtain that $\mathtt{neutral}^x_{lhd}(p)$ by the *i.h.* (1) and we conclude by rule `lhnapp`.
- If $t = p[y\backslash u]$, p is also $\to_{lhd}$ -normal, otherwise rule lhd_s would apply, and p has no free head variable y, otherwise rule $lhd_{\mathtt{e}}$ would apply. Then $p[y\backslash u]$ corresponds to one of cases (1)-(2)-(3). If p corresponds to (1), then $\mathtt{neutral}^x_{lhd}(p)$ by the *i.h.* (1) and we conclude with rule `lhnsubx`. If p corresponds to (2), then $\mathtt{normal}^x_{lhd}(p)$ by the *i.h.* (2) and we conclude with rule `lhnosubx`. If p corresponds to (3), then $\mathtt{normal}^\#_{lhd}(p)$ by the *i.h.* (3) and we conclude with rule `lhnosub`.

Now, given t in $\to_{lhd}$ -normal: if case (1) holds we conclude $\mathtt{neutral}^x_{lhd}(t)$ with the previous statement (1), then rules `lhnno` and `lhnox`; if case (2) holds we conclude

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

$\mathtt{normal}^{x}_{lhd}(t)$ with the previous statement (2), then rule $\mathtt{lhnox}$; if case (3) holds we conclude $\mathtt{normal}^{x}_{lhd}(t)$ with the previous statement (3), then rule $\mathtt{lhno}$;

$\Leftarrow$) By induction on $\mathtt{normal}_{lhd}(t)$. We remark that two cases are possible: either $\mathtt{normal}^{x}_{lhd}(t)$ for some variable x or $\mathtt{normal}^{\#}_{lhd}(t)$. We then refine the statement as follows:

1. If $\mathtt{neutral}^{x}_{lhd}(t)$, then t is $\rightarrow_{lhd}$ -normal and t has a head free variable x and t is not a (potentially) substituted abstraction.
2. If $\mathtt{normal}^{x}_{lhd}(t)$, then t is $\rightarrow_{lhd}$ -normal and t has a head free variable x.
3. If $\mathtt{normal}^{\#}_{lhd}(t)$, then t is $\rightarrow_{lhd}$ -normal and t has a head bound variable.

We reason by induction on the definition.

- If $\mathtt{neutral}^{x}_{lhd}(t)$ by rule $\mathtt{lhnvar}$, then property (1) trivially holds.
- If $\mathtt{neutral}^{x}_{lhd}(pu)$ because $\mathtt{neutral}^{x}_{lhd}(p)$ by rule $\mathtt{lhnapp}$, then by the *i.h.* (1) p is $\rightarrow_{lhd}$ -normal – so rule $lhd_{\mathtt{m}}$ does not apply – and p has a head free variable x and is not a (potentially) substituted abstraction – so rule $lhd_{@}$ does not apply. Then pu is $\rightarrow_{lhd}$ -normal, it has a head free variable x and is not a (potentially) substituted abstraction.
- If $\mathtt{neutral}^{x}_{lhd}(p[y\backslash u])$ because $\mathtt{neutral}^{x}_{lhd}(p)$ and $y \neq x$ by rule $\mathtt{lhnsubx}$, then by the *i.h.* (1) p is $\rightarrow_{lhd}$ -normal – so rule lhd_{s} does not apply – and p has a head free variable x and is not a (potentially) substituted abstraction – so rule $lhd_{\mathtt{e}}$ does not apply. Then $p[y\backslash u]$ is $\rightarrow_{lhd}$ -normal, it has a head free variable x and is not a (potentially) substituted abstraction.
- If $\mathtt{normal}^{x}_{lhd}(t)$ because $\mathtt{neutral}^{x}_{lhd}(t)$ by rule $\mathtt{lhnno}$, then by the *i.h.* (1) t is $\rightarrow_{lhd}$ -normal and has a head free variable x. We are then done for this case.
- If $\mathtt{normal}^{x}_{lhd}(\lambda y.p)$ because $\mathtt{normal}^{x}_{lhd}(t)$ and $y \neq x$ by rule $\mathtt{lhnolamx}$, then by the *i.h.* (2) p is $\rightarrow_{lhd}$ -normal – so that rule lhd_{λ} does not apply – and p has a head free variable x. We conclude $\lambda y.p$ is $\rightarrow_{lhd}$ -normal and has a head free variable x.
- If $\mathtt{normal}^{x}_{lhd}(p[y\backslash u])$ because $\mathtt{normal}^{x}_{lhd}(p)$ and $y \neq x$ by rule $\mathtt{lhnosubx}$, then by the *i.h.* (2) p is $\rightarrow_{lhd}$ -normal – so that rule lhd_{s} does not apply – and p has a head free variable x – so that rule $lhd_{\mathtt{e}}$ does not apply–. We conclude $p[y\backslash u]$ is $\rightarrow_{lhd}$ -normal and has a head free variable x.
- If $\mathtt{normal}^{\#}_{lhd}(\lambda x.p)$ because $\mathtt{normal}^{x}_{lhd}(p)$ by rule $\mathtt{lhnolamx}$, then by the *i.h.* (2) p is $\rightarrow_{lhd}$ -normal – so that rule lhd_{λ} does not apply–. We conclude $\lambda x.p$ is $\rightarrow_{lhd}$ -normal and has a bound head variable.
- If $\mathtt{normal}^{\#}_{lhd}(\lambda y.p)$ because $\mathtt{normal}^{\#}_{lhd}(p)$ by rule $\mathtt{lhnolam}$, then by the *i.h.* (3) p is $\rightarrow_{lhd}$ -normal – so that rule lhd_{λ} does not apply – and p has a bound head variable. We conclude $\lambda x.p$ is $\rightarrow_{lhd}$ -normal and has a bound head variable.
- If $\mathtt{normal}^{\#}_{lhd}(p[y\backslash u])$ because $\mathtt{normal}^{\#}_{lhd}(p)$ by rule $\mathtt{lhnosub}$, then by the *i.h.* (3) p is $\rightarrow_{lhd}$ -normal – so that rule lhd_{λ} does not apply – and p has a bound head variable. We conclude $p[y\backslash u]$ is $\rightarrow_{lhd}$ -normal and has a bound head variable. □

E.1 Tight correctness

Lemma E.1 (Multi-set decomposition for lhd). *Let* $M = \uplus_{k \in K} M_k$. *Then* $\Phi \rhd_{lhd} \Gamma \vdash^{(b,e,r)} t : M$ *if and only if there exist* $(\Phi_k)_{k \in K}$, $(\Gamma_k)_{k \in K}$, $(b_k)_{k \in K}$, $(e_k)_{k \in K}$ *and* $(r_k)_{k \in K}$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

and such that $\Phi_k \triangleright_{lhd} \Gamma_k \vdash^{(b_k,e_k,r_k)} t : M_k$*, where* $\Gamma = \uplus_{k\in K}\Gamma_k$, $b = +_{k\in K} b_k$, $e = +_{k\in K} e_k$ *and* $r = +_{k\in K} r_k$*. Moreover,* $|\Phi|_{lhd} = +_{k\in K}|\Phi_k|_{hd}$.

Proof By induction on the size of K. □

Lemma 8.3 (Tight spreading on neutral terms, plus typing contexts)**.** *Let* $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} t : A$ *be a derivation.*

1. *If* $\mathtt{neutral}^{x}_{lhd}(t)$ *then* $x \in \mathtt{dom}(\Gamma)$*. Moreover, if* $\mathtt{tight}(\Gamma(x))$ *then* $\mathtt{tight}(A)$ *and* $\mathtt{dom}(\Gamma) = \{x\}$.
2. *If* $\mathtt{normal}^{x}_{lhd}(t)$ *then* $x \in \mathtt{dom}(\Gamma)$*. Moreover, if* $\mathtt{tight}(\Gamma(x))$ *then* $\mathtt{dom}(\Gamma) = \{x\}$.
3. *If* $\mathtt{normal}^{\#}_{lhd}(t)$ *and* $\mathtt{tight}(A)$ *then* $A = \mathtt{abs}$ *and* Γ *is empty.*

In all the cases, if $\mathtt{tight}(\Gamma)$*, then the last rule of* Φ *is not* app_b.

Proof

1. By induction on $\mathtt{neutral}^{x}_{lhd}(t)$. Cases:
   - *Variable*, i.e. $t = x$. Then Φ is
     $$\frac{}{x : [A] \vdash^{(0,0,1)} x : A}\,\mathsf{ax}$$
     and so $\mathtt{dom}(\Gamma) = \{x\}$. If $\Gamma(x) = \mathtt{Tight}$ then it must be $A = \mathtt{tight}$.
   - *Application*, i.e. $t = pu$. The last rule of Φ can only be app_b or app^{hd}_r. In both cases the left subterm p is typed by a sub-derivation $\Phi' \triangleright_{lhd} \Gamma_p \vdash^{(b',e',r')} p : B$ such that all assignments in Γ_p appear in Γ. Since $\mathtt{neutral}^{x}_{lhd}(t)$ implies $\mathtt{neutral}^{x}_{lhd}(p)$, we can apply the *i.h.* and obtain that $x \in \mathtt{dom}(\Gamma_p) \subseteq \mathtt{dom}(\Gamma)$. If moreover, $\Gamma(x) = \mathtt{Tight}$ then $\Gamma_p(x) = \mathtt{Tight}$ and by *i.h.* $B = \mathtt{tight}$ and $\mathtt{dom}(\Gamma_p) = \{x\}$. This forces $B = \mathtt{neutral}$ and the last rule of Φ to be app^{hd}_r. Then $A = \mathtt{neutral}$ and $\Gamma = \Gamma_p$, that implies $\mathtt{dom}(\Gamma) = \{x\}$.
   - *Explicit substitution*, i.e. $t = p[y\backslash u]$ and $y \neq x$. The last rule of Φ is ES and the left subterm p is typed by a sub-derivation $\Phi' \triangleright_{lhd} \Gamma_p; y : M \vdash^{(b',e',r')} p : A$ such that all types in Γ_p appear in Γ. Since $\mathtt{neutral}^{x}_{lhd}(t)$ implies $\mathtt{neutral}^{x}_{lhd}(p)$, we can apply the *i.h.* and obtain that $x \in \mathtt{dom}(\Gamma_p) \subseteq \mathtt{dom}(\Gamma)$. If moreover, $\Gamma(x) = \mathtt{Tight}$ then $(\Gamma_p; y : M)(x) = \mathtt{Tight}$ and by the *i.h.* $A = \mathtt{tight}$ and $\mathtt{dom}(\Gamma_p; y : M) = \{x\}$. This forces $M = [\,]$ and the ES rule to have no right premiss. Then $\Gamma = \Gamma_p$, that implies $\mathtt{dom}(\Gamma) = \{x\}$.
2. By induction on $\mathtt{normal}^{x}_{lhd}(t)$. If $\mathtt{normal}^{x}_{lhd}(t)$ because $\mathtt{neutral}^{x}_{lhd}(t)$ then it follows from the previous point. The two other cases are:
   - *Abstraction*, i.e. $t = \lambda y.p$ with $\mathtt{normal}^{x}_{lhd}(p)$ and $y \neq x$. The last rule of Φ can only be fun_b or fun_r. In both cases the subterm p is typed by a sub-derivation $\Phi' \triangleright_{lhd} \Gamma; y : M \vdash^{(b',e',r')} p : B$. By *i.h.*, $x \in \mathtt{dom}(\Gamma; y : M)$ and so $x \in \mathtt{dom}(\Gamma)$, because $y \neq x$. If moreover, $\Gamma(x) = \mathtt{Tight}$ then by *i.h.* $\mathtt{dom}(\Gamma; y : M) = \{x\}$, that is, $M = [\,]$. Then $\mathtt{dom}(\Gamma) = \{x\}$.
   - *Explicit substitution*, i.e. $t = p[y\backslash u]$ with $\mathtt{normal}^{x}_{lhd}(p)$ and $y \neq x$. The last rule of Φ is ES and the left subterm p is typed by a sub-derivation $\Phi' \triangleright_{lhd}$

$\Gamma_p; y : M \vdash^{(b',e',r')} p : A$ such that all types in Γ_p appear in Γ. By *i.h.*, $x \in \mathtt{dom}(\Gamma_p) \subseteq \mathtt{dom}(\Gamma)$. If moreover, $\Gamma(x) = \mathtt{Tight}$ then by *i.h.* $\mathtt{dom}(\Gamma_p; y : M) = \{x\}$, that is, $M = [\,]$. Therefore, the ES rule has no right premiss. Then $\Gamma = \Gamma_p$, that implies $\mathtt{dom}(\Gamma) = \{x\}$.

3. By induction on $\mathtt{normal}^{\#}_{lhd}(t)$. Cases:
   - *Abstraction on the head variable*, i.e. $t = \lambda x.p$ with $\mathtt{normal}^{x}_{lhd}(p)$. If $A = \mathtt{tight}$ then the last rule of Φ can only be fun_r and $A = \mathtt{abs}$:

   $$\frac{\Gamma; x : \mathtt{Tight} \vdash^{(b,e,r)} p : \mathtt{tight}}{\Gamma \vdash^{(b,e,r+1)} \lambda x.p : \mathtt{abs}}\ \mathsf{fun}_r$$

   By the previous point, $\mathtt{dom}(\Gamma; x : \mathtt{Tight}) = \{x\}$, that is, Γ is empty.
   - *Abstraction on a non-head variable*, i.e. $t = \lambda x.p$ with $\mathtt{normal}^{\#}_{lhd}(p)$. If $A = \mathtt{tight}$ then the last rule of Φ can only be fun_r and $A = \mathtt{abs}$:

   $$\frac{\Gamma; x : \mathtt{Tight} \vdash^{(b,e,r)} p : \mathtt{tight}}{\Gamma \vdash^{(b,e,r+1)} \lambda x.p : \mathtt{abs}}\ \mathsf{fun}_r$$

   By *i.h.*, Γ is empty.
   - *Explicit substitution*, i.e. $t = p[y \backslash u]$ with $\mathtt{normal}^{\#}_{lhd}(p)$. The last rule of Φ is ES and the left subterm p is typed by a sub-derivation $\Phi' \triangleright_{lhd} \Gamma_p; y : M \vdash^{(b',e',r')} p : \mathtt{tight}$ such that all types in Γ_p appear in Γ. By *i.h.*, the typing context $\Gamma_p; y : M$ is empty, that forces $M = [\,]$. Therefore, the ES rule has no right premiss. Then $\Gamma = \Gamma_p$, i.e. Γ is empty. □

Proposition 8.4 (Properties of *lhd* tight typings for normal forms). *Let t be such that $\mathtt{normal}_{lhd}(t)$, and $\Phi \triangleright_{lhd} \Gamma \vdash^{(b,e,r)} t : A$ be a typing derivation.*

1. Size bound*: $|t|_{lhd} \leq |\Phi|$.*
2. Tightness*: if Φ is tight then $b = e = 0$ and $r = |t|_{lhd}$.*
3. Neutrality*: if $A = \mathtt{neutral}$ then $\mathtt{neutral}_{lhd}(t)$.*

Proof By induction on Φ. Cases of t:

- *Variable*, i.e. $t = x$. Then Φ has the following form and evidently verifies all the points of the statement:

$$\frac{}{x : [A] \vdash^{(0,0,1)} x : A}\ \mathsf{ax}$$

The derivation verifies $r = 1 = |x|_{lhd} = |\Phi|$, $b = e = 0$, as required.
- *Abstraction*, i.e. $t = \lambda x.p$ with $\mathtt{normal}_{lhd}(p)$. Cases of the last rule of Φ:

  – fun_b *rule*:

  $$\frac{\Psi \triangleright_{lhd} \Gamma; x : M \vdash^{(b',e,r)} p : A}{\Gamma \vdash^{(b'+1,e,r)} \lambda x.p : M \to A}\ \mathsf{fun}_b$$

  with $b = b' + 1$.

  1. *Size bound*: by *i.h.*, $|p|_{lhd} \leq |\Psi|$. Then, $|t|_{lhd} = |p|_{lhd} + 1 \leq_{i.h.} |\Psi| + 1 = |\Phi|$.
  2. *Tight bound*: Φ is not tight, so the statement trivially holds.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

– fun_r *rule*:

$$\frac{\Psi \rhd_{lhd}\ \Gamma; x : \mathtt{Tight} \vdash^{(b,e,r')} p : \mathtt{tight}}{\Gamma \vdash^{(b,e,r'+1)} \lambda x.p : \mathtt{abs}}\ \mathsf{fun}_r$$

with $r = r' + 1$.

1. *Size bound*: by *i.h.*, $|p|_{lhd} \leq |\Psi|$. Then, $|t|_{lhd} = |p|_{lhd} + 1 \leq_{i.h.} |\Psi| + 1 = |\Phi|$.
2. *Tight bound*: if Φ is tight, then Ψ is tight and by *i.h.* $r' = |p|_{lhd}$ and $b = e = 0$. Then, $r = r' + 1 =_{i.h.} |p|_{lhd} + 1 = |t|_{lhd}$.

- *Application*, i.e. $t = pu$ with $\mathtt{neutral}^x_{lhd}(p)$ for some x. Cases of the last rule of Φ:

– app_b *rule*:

$$\frac{\Psi \rhd_{lhd}\ \Delta \vdash^{(b',e',r')} p : M \to A \qquad \Theta \rhd \Pi \vdash^{(b'',e'',r'')} u : M}{\Delta \uplus \Pi \vdash^{(b'+b''+1,e'+e'',r'+r'')} pu : A}\ \mathsf{app}_b$$

with $b = b' + b'' + 1$, $e = e' + e''$, $r = r' + r''$, and $\Gamma = \Delta \uplus \Pi$.

1. *Size bound*: by *i.h.*, $|p|_{lhd} \leq |\Psi|$, from which it follows $|t|_{lhd} = |p|_{lhd} + 1 \leq_{i.h.} |\Psi| + 1 = |\Phi|$.
2. *Tight bound*: since Φ is tight and t is normal, this case is impossible by Lemma 8.3.

– app^{hd}_r *rule*:

$$\frac{\Psi \rhd_{lhd}\ \Gamma \vdash^{(b,e,r')} p : \mathtt{neutral}}{\Gamma \vdash^{(b,e,r'+1)} pu : \mathtt{neutral}}\ \mathsf{app}^{hd}_r$$

with $r = r' + 1$.

1. *Size bound*: by *i.h.*, $|p|_{lhd} \leq |\Psi|$. Then $|t|_{lhd} = |p|_{lhd} + 1 \leq_{i.h.} |\Psi| + 1 = |\Phi|$.
2. *Tight bound*: if Φ is tight, then Ψ is tight and by *i.h.* $r' = |p|_{lhd}$ and $b = e = 0$. Then, $r = r' + 1 =_{i.h.} |p|_{lhd} + 1 = |pu|_{lhd} = |t|_{lhd}$.

- *Explicit substitution*, i.e. $t = p[x\backslash u]$ and the last rule of Φ is:

$$\frac{\Psi \rhd_{lhd}\ \Delta; x : M \vdash^{(b',e',r')} p : A \qquad \Pi \vdash^{(b'',e'',r'')} u : M}{\Delta \uplus \Pi \vdash^{(b'+b'',e'+e''+|M|,r'+r''-|M|)} p[x\backslash u] : A}\ \mathtt{ES}$$

with $b = b' + b''$, $e = e' + e''$, $r = r' + r''$, and $\Gamma = \Delta \uplus \Pi$.

1. *Size bound*: by *i.h.* $|p|_{lhd} \leq |\Psi|$. Then $|t|_{lhd} = |p|_{lhd} \leq_{i.h.} |\Psi| < |\Phi|$.
2. *Tight bound*: There are two cases:

– $\mathtt{normal}^y_{lhd}(p)$ *for some* $y \neq x$. By Lemma 8.3.2 $y \in \mathtt{dom}(\Delta)$. All assignments in Δ are $\mathtt{Tight}$ because Φ is tight, and so applying Lemma 8.3.2 again we obtain that $\mathtt{dom}(\Delta) = \{y\}$, that is, that $M = [\,]$. Two consequences: first, the $\mathtt{ES}$ has no right premiss, that is, it rather has the following shape:

$$\frac{\Psi \rhd_{lhd}\ \Gamma \vdash^{(b,e,r)} p : A}{\Gamma \vdash^{(b,e,r)} p[x\backslash u] : A}\ \mathtt{ES}$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

second, Ψ is tight, and so by *i.h.* $b = e = 0$ and $r = |p|_{lhd}$. The statement follows from the fact that $|p|_{lhd} = |p[x\backslash u]|_{lhd}$.

– $\mathtt{normal}^{\#}_{lhd}(p)$. If Φ is tight then $A = \mathtt{tight}$ and by Lemma 8.3.3 the context $\Delta; x : M$ is empty, that is, $M = [\,]$. Two consequences: first, the ES has no right premiss, that is, it rather has the following shape:

$$\frac{\Psi \rhd_{lhd} \ \vdash^{(b,e,r)} p : A}{\vdash^{(b,e,r)} p[x\backslash u] : A}\ \mathsf{ES}$$

second, Ψ is tight, and so by *i.h.* $b = e = 0$ and $r = |p|_{lhd}$. The statement follows from the fact that $|p|_{lhd} = |p[x\backslash u]|_{lhd}$. □

Lemma 8.5 (Linear substitution and typings for *lhd*). *Let* $\Phi \rhd_{lhd} \ x : M; \Gamma \vdash^{(b,e,r)} H\langle\!\langle x\rangle\!\rangle : A$. *Then there exists* $B \in M$ *such that for all* $\Phi_t \rhd_{lhd} \ \Gamma_t \vdash^{(b_t,e_t,r_t)} t : B$ *there exists a derivation* $\Psi \rhd_{lhd} \ x : M \setminus [B]; \Gamma \uplus \Gamma_t \vdash^{(b+b_t,e+e_t,r+r_t-1)} H\langle\!\langle t\rangle\!\rangle : A$. *Moreover,* $|\Psi| = |\Phi| + |\Phi_t| - 1$.

Proof By induction on H. Cases:

- *Empty context*, i.e. $H = \langle\cdot\rangle$. The typing derivation Φ is simply

$$\frac{}{x : [A] \vdash^{(0,0,1)} x : A}\ \mathsf{ax}$$

and Γ is empty. Then $M = [A]$. The statement then holds with respect to $\Psi := \Phi_t$, because $b = 0$, $e = 0$, and $r = 1$. The moreover statement is straightforward since $|\Phi| = 1$.

- *Abstraction*, i.e. $H = \lambda y.H'$. Two sub-cases, depending on the last rule of Φ:

1. The last rule is fun_b, and so Φ has the form:

$$\frac{x\colon M; y\colon N; \Gamma \vdash^{(b_\Gamma,e,r)} H'\langle\!\langle x\rangle\!\rangle : A}{x\colon M; \Gamma \vdash^{(b_\Gamma+1,e,r)} \lambda y.H'\langle\!\langle x\rangle\!\rangle : N \to A}\ \mathsf{fun}_b$$

where $b = b_\Gamma + 1$. By *i.h.*, there exists a splitting $M = [B] \uplus O$ such that for every derivation $\Psi \rhd_{lhd} \ \Delta \vdash^{(b',e',r')} t : B$ there exists a derivation

$$\Phi_{H'\langle\!\langle t\rangle\!\rangle} \rhd_{lhd} \ x : O; y\colon N; \Gamma \uplus \Delta \vdash^{(b_\Gamma+b',e+e',r+r'-1)} H'\langle\!\langle t\rangle\!\rangle : A$$

Note that $y \notin \mathtt{dom}(\Delta)$: we are working up to α-equivalence, and so $y \notin \mathtt{fv}(t)$, and the system is relevant, and so $y \notin \mathtt{fv}(t)$ implies $y \notin \mathtt{dom}(\Delta)$. By applying the fun_b rule we obtain:

$$\frac{x : O; y\colon N; \Gamma \uplus \Delta \vdash^{(b_\Gamma+b',e+e',r+r'-1)} H'\langle\!\langle t\rangle\!\rangle : A}{x\colon O; \Gamma \uplus \Delta \vdash^{(b_\Gamma+1+b',e+e',r+r'-1)} \lambda y.H'\langle\!\langle t\rangle\!\rangle : N \to A}\ \mathsf{fun}_b$$

that satisfies the statement (because $b = b_\Gamma + 1$).

2. The last rule is fun_r, and so Φ has the form:

$$\frac{x\colon M; y\colon \mathtt{Tight}; \Gamma \vdash^{(b,e,r_\Gamma)} H'\langle\!\langle x\rangle\!\rangle : \mathtt{tight}}{x\colon M; \Gamma \vdash^{(b,e,r_\Gamma+1)} \lambda y.H'\langle\!\langle x\rangle\!\rangle : \mathtt{abs}}\ \mathsf{fun}_r$$

where $r = r_\Gamma + 1$. By *i.h.*, there exists a splitting $M = [B] \uplus O$ such that for every derivation $\Psi \triangleright_{lhd} \ \Delta \vdash^{(b',e',r')} t : B$ there exists a derivation

$$\Phi_{H'\langle\langle t\rangle\rangle} \triangleright_{lhd} \ x : O; y : N; \Gamma \uplus \Delta \vdash^{(b+b',e+e',r_\Gamma+r'-1)} H'\langle\langle t\rangle\rangle : \mathtt{tight}$$

Note that $y \notin \mathtt{dom}(\Delta)$, for the same reasons as in the previous sub-case. By applying an fun_r rule we obtain:

$$\frac{x : O; y : \mathtt{Tight}; \Gamma \uplus \Delta \vdash^{(b+b',e+e',r_\Gamma+r'-1)} H'\langle\langle t\rangle\rangle : \mathtt{tight}}{x : O; \Gamma \uplus \Delta \vdash^{(b+b',e+e',r_\Gamma+r')} \lambda y.H'\langle\langle t\rangle\rangle : \mathtt{abs}}\ \mathsf{fun}_r$$

that satisfies the statement (because $r = r_\Gamma + 1$).

In both cases, the moreover statement is straightforward by the *i.h.*

- *Left on an application*, i.e. $H = H'p$. Two sub-cases, depending on the last rule of Φ:

1. The last rule is app_b, and so Φ has the form:

$$\frac{x : M_\Pi; \Pi \vdash^{(b_\Pi,e_\Pi,r_\Pi)} H'\langle\langle x\rangle\rangle : N \to A \quad x : M_\Sigma; \Sigma \vdash^{(b_\Sigma,e_\Sigma,r_\Sigma)} p : N}{x : (M_\Pi \uplus M_\Sigma); (\Pi \uplus \Sigma) \vdash^{(b_\Pi+b_\Sigma,e_\Pi+e_\Sigma,r_\Pi+r_\Sigma)} H'\langle\langle x\rangle\rangle p : A}\ \mathsf{app}_b$$

where $\Gamma = \Pi \uplus \Sigma$, $\Pi(x) = \Sigma(x) = [\,]$, $M_\Pi \uplus M_\Sigma = M$, $b = b_\Pi + b_\Sigma$, $e = e_\Pi + e_\Sigma$, and $r = r_\Pi + r_\Sigma$.

By *i.h.*, there exists a splitting $M_\Pi = [B] \uplus O$ such that for every derivation $\Psi \triangleright_{lhd} \ \Delta \vdash^{(b',e',r')} t : B$ there exists a derivation

$$\Phi_{H'\langle\langle t\rangle\rangle} \triangleright_{lhd} \ x : O; \Pi \uplus \Delta \vdash^{(b_\Pi+b',e_\Pi+e',r_\Pi+r'-1)} H'\langle\langle t\rangle\rangle : N \to A$$

By applying an app_b rule, we obtain

$$\frac{x : O; \Pi \uplus \Delta \vdash^{(b_\Pi+b',e_\Pi+e',r_\Pi+r'-1)} H'\langle\langle t\rangle\rangle : N \to A \quad x : M_\Sigma; \Sigma \vdash^{(b_\Sigma,e_\Sigma,r_\Sigma)} p : N}{x : (O \uplus M_\Sigma); (\Pi \uplus \Delta \uplus \Sigma) \vdash^{(b_\Pi+b'+b_\Sigma,e_\Pi+e'+e_\Sigma,r_\Pi+r'+r_\Sigma-1)} H'\langle\langle t\rangle\rangle p : A}\ \mathsf{app}_b$$

Now, by defining $N := O \uplus M_\Sigma$, we obtain $M = M_\Pi \uplus M_\Sigma = [B] \uplus O \uplus M_\Sigma = [B] \uplus N$. Therefore by applying the equalities on the type context the last obtained judgement is in fact:

$$x : N; (\Gamma \uplus \Delta) \vdash^{(b_\Pi+b'+b_\Sigma,e_\Pi+e'+e_\Sigma,r_\Pi+r'+r_\Sigma-1)} H'\langle\langle t\rangle\rangle p : A$$

and by applying those on the indices we obtain:

$$x : N; (\Gamma \uplus \Delta) \vdash^{(b+b',e+e',r+r'-1)} H'\langle\langle t\rangle\rangle p : A$$

as required.

2. The last rule of Φ is app_r^{hd}, and so Φ has the form

$$\frac{x : M_\Pi; \Pi \vdash^{(b_\Pi,e_\Pi,r_\Pi)} H'\langle\langle x\rangle\rangle : \mathtt{neutral} \quad x : M_\Sigma; \Sigma \vdash^{(b_\Sigma,e_\Sigma,r_\Sigma)} p : \mathtt{tight}}{x : (M_\Pi \uplus M_\Sigma); (\Pi \uplus \Sigma) \vdash^{(b_\Pi+b_\Sigma,e_\Pi+e_\Sigma,r_\Pi+r_\Sigma+1)} H'\langle\langle x\rangle\rangle p : \mathtt{neutral}}\ \mathsf{app}_r^{hd}$$

where $\Gamma = \Pi \uplus \Sigma$, $\Pi(x) = \Sigma(x) = [\,]$, $M_\Pi \uplus M_\Sigma = M$, $b = b_\Pi + b_\Sigma$, $e = e_\Pi + e_\Sigma$, and $r = r_\Pi + r_\Sigma + 1$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

By *i.h.*, there exists a splitting $M_\Pi = [B] \uplus O$ such that for every derivation $\Psi \rhd_{lhd} \Delta \vdash^{(b',e',r')} t : B$ there exists a derivation

$$\Phi_{H'\langle\langle t\rangle\rangle} \rhd_{lhd} x : O; \Pi \uplus \Delta \vdash^{(b_\Pi+b',e_\Pi+e',r_\Pi+r'-1)} H'\langle\langle t\rangle\rangle : \mathtt{neutral}$$

By applying an $\mathtt{app}_r^{hd}$ rule, we obtain

$$\frac{x : O; \Pi \uplus \Delta \vdash^{(b_\Pi+b',e_\Pi+e',r_\Pi+r'-1)} H'\langle\langle t\rangle\rangle : \mathtt{neutral} \quad x : M_\Sigma; \Sigma \vdash^{(b_\Sigma,e_\Sigma,r_\Sigma)} p : \mathtt{tight}}{x : (O \uplus M_\Sigma); (\Pi \uplus \Delta \uplus \Sigma) \vdash^{(b_\Pi+b'+b_\Sigma,e_\Pi+e'+e_\Sigma,r_\Pi+r'+r_\Sigma)} H'\langle\langle t\rangle\rangle p : \mathtt{neutral}} \,\mathtt{app}_r^{hd}$$

Now, by defining $N := O \uplus M_\Sigma$, we obtain $M = M_\Pi \uplus M_\Sigma = [B] \uplus O \uplus M_\Sigma = [B] \uplus N$. Therefore by applying the equalities on the type context the last obtained judgement is in fact:

$$x : N; (\Gamma \uplus \Delta) \vdash^{(b_\Pi+b'+b_\Sigma,e_\Pi+e'+e_\Sigma,r_\Pi+r'+r_\Sigma)} H'\langle\langle t\rangle\rangle p : \mathtt{neutral}$$

and by applying those on the indices, we obtain

$$x : N; (\Gamma \uplus \Delta) \vdash^{(b+b',e+e',r+r'-1)} H'\langle\langle t\rangle\rangle p : \mathtt{neutral}$$

as required.

In both cases the moreover statement is straightforward by the *i.h.*

- *Left of a substitution*, i.e. $H = H'[y\backslash p]$. Note that $x \neq y$, because the hypothesis $H\langle\langle x\rangle\rangle$ implies that H does not capture x.

  The last rule of Φ can only be ES, and so Φ has the form

$$\frac{x : M_\Pi; y : M'; \Pi \vdash^{(b_\Pi,e_\Pi,r_\Pi)} H'\langle\langle x\rangle\rangle : A \quad x : M_\Sigma; \Sigma \vdash^{(b_\Sigma,e_\Sigma,r_\Sigma)} p : M'}{x : (M_\Pi \uplus M_\Sigma); (\Pi \uplus \Sigma) \vdash^{(b_\Pi+b_\Sigma,e_\Pi+e_\Sigma+|M|,r_\Pi+r_\Sigma-|M|)} H'\langle\langle x\rangle\rangle[y\backslash p] : A} \,\mathtt{ES}$$

  where $\Gamma = \Pi \uplus \Sigma$, $\Pi(x) = \Sigma(x) = [\,]$, $M_\Pi \uplus M_\Sigma = M$, $b = b_\Pi + b_\Sigma$, $e = e_\Pi + e_\Sigma + |M|$, and $r = r_\Pi + r_\Sigma - |M|$.
  By *i.h.*, there exists a splitting $M_\Pi = [B] \uplus O$ such that for every derivation $\Psi \rhd_{lhd} \Delta \vdash^{(b',e')} t : B$ there exists a derivation

$$\Phi_{H'\langle\langle t\rangle\rangle} \rhd_{lhd} x : O; y : M'; \Pi \uplus \Delta \vdash^{(b_\Pi+b',e_\Pi+e',r_\Pi+r'-1)} H'\langle\langle t\rangle\rangle : A$$

  Note that $y \notin \mathtt{dom}(\Delta)$: we are working up to α-equivalence, and so $y \notin \mathtt{fv}(p)$, and the system is relevant, and so $y \notin \mathtt{fv}(p)$ implies $y \notin \mathtt{dom}(\Delta)$. By applying a ES rule we obtain

$$\frac{x : O; y : M'; \Pi \uplus \Delta \vdash^{(b_\Pi+b',e_\Pi+e',r_\Pi+r'-1)} H'\langle\langle t\rangle\rangle : A \quad x : M_\Sigma; \Sigma \vdash^{(b_\Sigma,e_\Sigma,r_\Sigma)} p : M'}{x : O \uplus M_\Sigma; \Pi \uplus \Delta \uplus \Sigma \vdash^{(b_\Pi+b'+b_\Sigma,e_\Pi+e'+e_\Sigma+|M|,r_\Pi+r'-1+r_\Sigma-|M|)} H'\langle\langle t\rangle\rangle[y\backslash p] : A} \,\mathtt{ES}$$

  Now, by defining $N := O \uplus M_\Sigma$, we obtain $M = M_\Pi \uplus M_\Sigma = [B] \uplus O \uplus M_\Sigma = [B] \uplus N$. Therefore, by applying the equalities on the type context, the last obtained judgement is in fact

$$x : N; (\Gamma \uplus \Delta) \vdash^{(b_\Pi+b'+b_\Sigma,e_\Pi+e'+e_\Sigma+|M|,r_\Pi+r'-1+r_\Sigma-|M|)} H'\langle\langle t\rangle\rangle[y\backslash p] : A$$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

and by applying those on the indices, we obtain

$$x\colon N; (\Gamma \uplus \Delta) \vdash^{(b+b',e+e',r+r'-1)} H'\langle\!\langle t\rangle\!\rangle[y\backslash p] : A$$

as required.

The moreover statement is straightforward by the *i.h.* □

Proposition 8.6 (Quantitative subject reduction for *lhd*). *If $\Phi \triangleright \Gamma \vdash^{(b,e,r)} t : A$, then*

1. *if $t \to_{\mathtt{m}} u$ then $b \geq 1$ and there is a typing Φ' such that $\Phi' \triangleright \Gamma \vdash^{(b-1,e,r)} u : A$ and $|\Phi'| = |\Phi| - 1$.;*
2. *if $t \to_{\mathtt{e}} u$ then $e \geq 1$ and there is a typing Φ' such that $\Phi' \triangleright \Gamma \vdash^{(b,e-1,r)} u : A$ and $|\Phi'| = |\Phi| - 1$.*

Proof By induction on the reduction relation $\to_{lhd}$.

- $t = L\langle\lambda x.v\rangle s \to_{\mathtt{m}} L\langle v[x\backslash s]\rangle = t'$, then we proceed by induction on L. Let $L = \langle\cdot\rangle$. By construction the derivation Φ is of the form

$$\dfrac{\dfrac{\Phi_v \triangleright\ x : M; \Pi \vdash^{(b_v,e_v,r_v)} v : \sigma}{\Pi \vdash^{(b_v+1,e_v+|M|,r_v-|M|)} \lambda x.v : M \to \sigma} \qquad \Phi_s \triangleright\ \Gamma \vdash^{(b_s,e_s,r_s)} s : M}{\Pi \uplus \Gamma \vdash^{(b_v+b_s+1,e_v+e_s+|M|,r_v+r_s-|M|)} (\lambda x.v)s : A}$$

where $b = b_v + b_s + 1$, $e = e_v + e_s + |M|$ and $r = r_v + r_s - |M|$. We notice that $b = b_v + b_s + 1 \geq 1$, as required. We construct the following derivation Φ':

$$\dfrac{x : M; \Pi \vdash^{(b_v,e_v,r_s)} v : \sigma \qquad \Gamma \vdash^{(b_s,e_s,r_s)} s : M}{\Pi \uplus_{i\in I} \Gamma_i \vdash^{(b_v+b_s,e_v+e_s+|M|,r_v+r_s-|M|)} v[x\backslash s] : \sigma}$$

We let $b' = b_v + b_s$, $e' = e_v + e_s + |M|$ and $r' = r_v + r_s - |M|$. So that we can verify $b = b' - 1$, $b = b'$, and $r = r'$ as required. We conclude since $|\Phi'| = |\Phi_v| + |\Phi_s| + 1 = |\Phi| - 1$.

For $L = L'[y\backslash s]$, the statement follows from the *i.h.*
- $t = H\langle\!\langle x\rangle\!\rangle[x\backslash v] \to_{lhd} H\langle\!\langle v\rangle\!\rangle[x\backslash v] = u$, then Φ is of the form

$$\dfrac{\Phi_{H\langle\!\langle x\rangle\!\rangle} \triangleright\ x : M; \Pi \vdash^{(b_H,e_H,r_H)} H\langle\!\langle x\rangle\!\rangle : A \qquad \Phi_v \triangleright\ \Delta \vdash^{(b_v,e_v,r_v)} v : M}{\Pi \uplus \Delta \vdash^{(b_H+b_v,e_H+e_v+|M|,r_H+r_v-|M|)} H\langle\!\langle x\rangle\!\rangle[x\backslash v] : A}$$

where $b = b_H + b_v$, $e = e_H + e_v + |M|$ and $r = r_H + r_v - |M|$.

It is not difficult to see that $|M| \neq 0$ and thus $e \geq 1$ as required.

Let $M = [B] \uplus N$ be the splitting of M given by the linear substitution lemma (Lemma 8.5) applied to $\Phi_H\langle\!\langle x\rangle\!\rangle$. By the multi-sets decomposition lemma (Lemma E.1) applied to Φ_v with respect to such a decomposition, there exist two derivations:

$$\Phi_B \triangleright\ \Delta_B \vdash^{(b_B,e_B,r_B)} v : B \qquad \Phi_N \triangleright\ \Delta_N \vdash^{(b_N,e_N,r_N)} v : N$$

such that $\Delta_v = \Delta_B \uplus \Delta_N$, $b_v = b_B + b_N$, $e_v = e_B + e_N$, $r_v = r_B + r_N$, and $|\Phi_v| = |\Phi_B| + |\Phi_N|$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

By the linear substitution Lemma 8.5, there exist a derivation

$$\Phi_{H\langle\langle v\rangle\rangle} \triangleright_{lhd} \ x : N; \Pi \uplus \Delta_B \vdash^{(B,E,R)} H\langle\langle v\rangle\rangle : A$$

where $B = b_H + b_B$, $E = e_H + e_B$, $R = r_H + r_B - 1$ and $|\Phi_{H\langle\langle v\rangle\rangle}| = |\Phi_{H\langle\langle x\rangle\rangle}| + |\Phi_B| - 1$. We construct the following derivation Φ':

$$\frac{x : N; \Pi \uplus \Delta_B \vdash^{(B,E,R)} H\langle\langle v\rangle\rangle : A \qquad \Delta_N \vdash^{(b_N,e_N,r_N)} v : N}{\Pi \uplus \Delta_B \uplus \Delta_N \vdash^{(B+b_N, E+e_N+|N|, R+r_N-|N|)} H\langle\langle v\rangle\rangle[x\backslash v] : A}$$

that verifies the statement because

- $B + b_N = b_H + b_B + b_N = b_H + b_v = b$,
- $E + e_N + |N| = e_H + e_B + e_N + |N| = e_H + e_v + |N| = e_H + e_v + |M| - 1 = e - 1$,
- $R + r_N - |N| = r_H + r_B - 1 + r_N - |N| = r_H + r_v - |M| = r$,
- $|\Phi'| = |\Phi_{H\langle\langle v\rangle\rangle}| + |\Phi_N| + 1 = |\Phi_{H\langle\langle x\rangle\rangle}| + |\Phi_B| - 1 + |\Phi_N| + 1 = |\Phi_{H\langle\langle x\rangle\rangle}| + |\Phi_v| = |\Phi| - 1$.

- All the other cases follow from the *i.h.* □

Theorem 8.7 (Tight correctness for *lhd*)**.** *Let $\Phi \triangleright_{lhd} \ \Gamma \vdash^{(b,e,r)} t : A$ be a tight derivation. Then there exists p such that $t \rightarrow^{b+e}_{lhd} p$, $\mathtt{normal}_{lhd}(p)$ and $|p|_{lhd} = r$. Moreover, if $A = \mathtt{neutral}$ then $\mathtt{neutral}_{lhd}(p)$.*

Proof By induction on $|\Phi|$. If t is a $\rightarrow_{lhd}$ normal form – that covers the base case $|\Phi| = 1$, for which t is necessarily a variable – then by taking $p := t$ and $k := 0$ the statement follows from the *tightness* property of tight typings of normal forms (Proposition 8.4.2) – the *moreover* part follows from the *neutrality* property (Proposition 8.4.3). Otherwise, two cases:

1. *Multiplicative steps*: $t \rightarrow_{\mathtt{m}} u$ and by quantitative subject reduction (Proposition 8.6) there is a derivation $\Psi \triangleright_{lhd} \ \Gamma \vdash^{(b-1,e,r)} u : A$ such that $|\Psi| = |\Phi| - 1$. By *i.h.*, there exists p such that $\mathtt{normal}_{lhd}(p)$ and $u \rightarrow^{b-1+e}_{lhd} p$ and $|p|_{lhd} = r$. Just note that $t \rightarrow_{\mathtt{m}} u \rightarrow^{b-1+e}_{lhd} p$, that is, $t \rightarrow^{b+e}_{lhd} p$.
2. *Exponential steps*: $t \rightarrow_{\mathtt{e}} u$ and by quantitative subject reduction (Proposition 8.6) there is a derivation $\Psi \triangleright_{lhd} \ \Gamma \vdash^{(b,e-1,r)} u : A$ such that $|\Psi| = |\Phi| - 1$. By *i.h.*, there exists p such that $\mathtt{normal}_{lhd}(p)$ and $u \rightarrow^{b+e-1}_{lhd} p$ and $|p|_{lhd} = r$. Just note that $t \rightarrow_{\mathtt{e}} u \rightarrow^{b+e-1}_{lhd} p$, that is, $t \rightarrow^{b+e}_{lhd} p$. □

E.2 Tight completeness

Proposition 8.8 (Linear head normal forms are tightly typable for *lhd*)**.** *Let t be such that $\mathtt{normal}_{lhd}(t)$. Then there exists a tight typing $\Phi \triangleright_{lhd} \ \Gamma \vdash^{(0,0,|t|_{lhd})} t : A$. Moreover, if $\mathtt{neutral}_{lhd}(t)$ then $A = \mathtt{neutral}$, and if $\mathtt{abs}_{lhd}(t)$ then $A = \mathtt{abs}$.*

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proof In the proof, for the sake of simplicity, we let the indices on the judgements generic, and not as precise as in the statement, because once one knows that there is a tight derivation then the indices are forced by Proposition 8.4.

1. By induction on $\mathtt{neutral}^{x}_{lhd}(t)$:

   - *Variable*, i.e. $t = x$. Then the derivation

   $$\dfrac{}{x : [\mathtt{neutral}] \vdash^{(0,0,1)} x : \mathtt{neutral}}\ \mathsf{ax}_r$$

     is tight and types x with $\mathtt{neutral}$.
   - *Application*, i.e. $t = pu$ and $\mathtt{neutral}_{lhd}(t)$ because $\mathtt{neutral}_{lhd}(p)$. By *i.h.*, there is a tight derivation $\Psi \rhd_{lhd}\ \Gamma \vdash^{(b,e,r)} p : \mathtt{neutral}$. Then the following is a tight derivation Φ typing $t = pu$ with $\mathtt{neutral}$:

   $$\dfrac{\Psi \rhd_{lhd}\ \Gamma \vdash^{(b,e,r)} p : \mathtt{neutral}}{\Gamma \vdash^{(b,e,r+1)} pu : \mathtt{neutral}}\ \mathsf{app}^{hd}_r$$

   - *Explicit substitution*, i.e. $t = p[y \backslash u]$ and $\mathtt{neutral}_{lhd}(t)$ because $\mathtt{neutral}^{x}_{lhd}(p)$ and $x \neq y$. By *i.h.*, there is a tight derivation $\Psi \rhd_{lhd}\ \Gamma \vdash^{(b,r)} p : \mathtt{neutral}$. By Lemma 8.3.1, $\mathtt{dom}(\Gamma) = \{y\}$, that is, in Γ the variable x is implicitly typed with $[\,]$. Then the following tight derivation Φ types $t = p[x \backslash u]$ with $\mathtt{neutral}$:

   $$\dfrac{\Gamma; x : [\,] \vdash^{(b,e,r)} t : \mathtt{neutral}}{\Gamma \vdash^{(b,e,r)} t[x \backslash u] : \mathtt{neutral}}\ \mathsf{ES}$$

2. First, by induction on $\mathtt{normal}^{x}_{lhd}(t)$:

   - $\mathtt{normal}^{x}_{lhd}(t)$ because $\mathtt{neutral}^{x}_{lhd}(t)$. Then it follows from the previous point.
   - *Abstraction*, i.e. $t = \lambda y.p$ and $\mathtt{normal}^{x}_{lhd}(t)$ because $\mathtt{normal}^{x}_{lhd}(p)$ and $x \neq y$. By *i.h.* there is a tight derivation $\Psi \rhd_{lhd}\ \Delta \vdash^{(b,e,r)} p : \mathtt{tight}$. Since the derivation Ψ is tight, the typing context Δ has the shape $\Gamma; y : \mathtt{Tight}$ (potentially, $y : [\,]$). Then the following is a tight derivation for $\lambda y.p$ with $\mathtt{abs}$:

   $$\dfrac{\Psi \rhd_{lhd}\ \Gamma; y : \mathtt{Tight} \vdash^{(b,e,r+1)} p : \mathtt{tight}}{\Gamma \vdash^{(b,e,r+1)} \lambda y.p : \mathtt{abs}}\ \mathsf{fun}_r$$

   - *Explicit substitution*, i.e. $t = p[y \backslash u]$ and $\mathtt{normal}^{x}_{lhd}(t)$ because $\mathtt{normal}^{x}_{lhd}(p)$ and $x \neq y$. It is essentially like in the neutral case. By *i.h.*, there is a tight derivation $\Psi \rhd_{lhd}\ \Delta \vdash^{(b,e,r)} p : \mathtt{tight}$. By Lemma 8.3.1, $\mathtt{dom}(\Delta) = \{x\}$, that is, in Δ the variable y is implicitly typed with $[\,]$. Then using the notation $\Delta = \Gamma; y : [\,]$ the following tight derivation Φ types $t = p[y \backslash u]$:

   $$\dfrac{\Gamma; y : [\,] \vdash^{(b,e,r)} p : \mathtt{tight}}{\Gamma \vdash^{(b,e,r)} p[y \backslash u] : \mathtt{tight}}\ \mathsf{ES}$$

   The part about predicates follows from the *i.h.*

   Now, by induction on $\mathtt{normal}^{\#}_{lhd}(t)$:

   - *Abstraction on the head variable*, i.e. $t = \lambda x.p$ and $\mathtt{normal}^{x}_{lhd}(t)$ because $\mathtt{normal}^{x}_{lhd}(p)$. By *i.h.* there is a tight derivation $\Psi \rhd_{lhd}\ \Delta \vdash^{(b,e,r)} p : \mathtt{tight}$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Since the derivation Ψ is tight, the typing context Δ has the shape $\Gamma; y : \mathtt{Tight}$ (potentially, $y : [\,]$). Then the following is a tight derivation for $\lambda y.p$ with $\mathtt{abs}$:

$$\frac{\Psi \rhd_{lhd} \;\; \Gamma; y : \mathtt{Tight} \vdash^{(b,e,r)} p : \mathtt{tight}}{\Gamma \vdash^{(b,e,r+1)} \lambda y.p : \mathtt{abs}}\ \mathsf{fun}_r$$

- *Abstraction on a non-head variable*, i.e. $t = \lambda x.p$ and $\mathtt{normal}^{x}_{lhd}(t)$ because $\mathtt{normal}^{\#}_{lhd}(p)$. It is exactly as in the previous sub-case. By *i.h.* there is a tight derivation $\Psi \rhd_{lhd} \;\; \Delta \vdash^{(b,e,r)} p : \mathtt{tight}$. Since the derivation Ψ is tight, the typing context Δ has the shape $\Gamma; y : \mathtt{Tight}$ (potentially, $y : [\,]$). Then the following is a tight derivation for $\lambda y.p$ with $\mathtt{abs}$:

$$\frac{\Psi \rhd_{lhd} \;\; \Gamma; y : \mathtt{Tight} \vdash^{(b,e,r)} p : \mathtt{tight}}{\Gamma \vdash^{(b,e,r+1)} \lambda y.p : \mathtt{abs}}\ \mathsf{fun}_r$$

- *Explicit substitution*, i.e. $t = p[y \backslash u]$ and $\mathtt{normal}^{\#}_{lhd}(t)$ because $\mathtt{normal}^{\#}_{lhd}(p)$. By *i.h.*, there is a tight derivation $\Psi \rhd_{lhd} \;\; \Delta \vdash^{(b,e,r)} p : \mathtt{tight}$. By Lemma 8.3.3, Δ is empty, that is, the variable y is implicitly typed with $[\,]$. Then the following tight derivation Φ types $t = p[x \backslash u]$:

$$\frac{y : [\,] \vdash^{(b,e,r)} p : \mathtt{tight}}{\vdash^{(b,e,r)} p[y \backslash u] : \mathtt{tight}}\ \mathrm{ES}$$

The part about predicates follows from the *i.h.* □

Lemma 8.9 (Linear anti-substitution and typings for *lhd*)**.** *Let* $\Phi \rhd_{lhd} \;\; \Gamma \vdash^{(b,e,r)} H\langle\!\langle u \rangle\!\rangle : A$, *where* $x \notin u$. *Then there exists*

- *a type* B
- *a typing derivation* $\Phi_u \rhd_{lhd} \;\; \Gamma_u \vdash^{(b_u,e_u,r_u)} u : B$
- *a typing derivation* $\Phi_{H\langle\!\langle x \rangle\!\rangle} \rhd_{lhd} \;\; \Gamma' \uplus x{:}[B] \vdash^{(b',e',r')} H\langle\!\langle x \rangle\!\rangle : A$

such that

- Typing contexts*:* $\Gamma = \Gamma' \uplus \Gamma_u$.
- Indices*:* $(b,e,r) = (b' + b_u, e' + e_u, r' + r_u - 1)$.
- Sizes*:* $|\Phi| = |\Phi_u| + |\Phi_{H\langle\!\langle x \rangle\!\rangle}| - 1$.

Proof By induction on H.

- If $H = \langle \cdot \rangle$, then we let $\Gamma' = \emptyset$ and $\sigma = \tau$. We have $(b', e', r') = (0, 0, 1)$ so that $(b, e, r) = (b_u, e_u, r_u)$. All the equalities are verified.
- In all the other cases the property is straightforward by the *i.h.* □

Proposition 8.10 (Quantitative subject expansion for *lhd*)**.** *If* $\Phi' \rhd_{lhd} \;\; \Gamma \vdash^{(b,e,r)} t' : A$ *then*

1. *if* $t \to_{\mathtt{m}} t'$ *then there is a derivation* $\Phi \rhd_{lhd} \;\; \Gamma \vdash^{(b+1,e,r)} t : \tau$ *and* $|\Phi'| = |\Phi| + 1$*;*
2. *if* $t \to_{\mathtt{e}} t'$ *then there is a derivation* $\Phi \rhd_{lhd} \;\; \Gamma \vdash^{(b,e+1,r)} t : A$ *and* $|\Phi'| = |\Phi| + 1$.

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press

Proof The proof is by induction on $t \to_{lhd} t'$.

- If $t = L\langle\!\langle \lambda x.p\rangle\!\rangle u \to L\langle\!\langle p[x/u]\rangle\!\rangle = t'$, then we proceed by induction on L. Let $L = \langle\cdot\rangle$, then by construction $\Gamma = \Delta \uplus \Pi$ and we have the following derivation:

$$\frac{x{:}M;\Delta \vdash^{(b_p,e_p,r_p)} p : \tau \qquad \Pi \vdash^{(b',e',r')} u : M}{\Delta \uplus \Pi \vdash^{(b_p+b',e_p+e'+|M|,r_p+r'-|M|)} p[x/u] : \tau}$$

where $b = b_p + b'$, $e = e_p + e' + |M|$ and $r = r_p + r' - |M|$. We then construct the following derivation:

$$\frac{\dfrac{x{:}M;\Delta \vdash^{(b_p,e_p,r_p)} p : \tau}{\Delta \vdash^{(b_p+1,e_p+|M|,r_p-|M|)} \lambda x.p : M \to \tau} \qquad \Pi \vdash^{(b',e',r')} u : M}{\Delta \uplus \Pi \vdash^{(b_p+b'+1,e_p+e'+|M|,r_p+r'-|M|} (\lambda x.p)u : \tau}$$

For $L = L'[y/u]$, the statement follows from the *i.h.*

- If $t = H\langle\!\langle x\rangle\!\rangle[x/u] \to H\langle\!\langle u\rangle\!\rangle[x/u] = t'$, then by construction $\Gamma = \Delta \uplus \Pi$ and the type derivation of t' has the following form:

$$\frac{x{:}M;\Delta \vdash^{(b_H,e_H,r_H)} H\langle\!\langle u\rangle\!\rangle : \tau \qquad \Pi \vdash^{(b_u,e_u,r_u)} u : M}{\Delta \uplus \Pi \vdash^{(b,e,r)} H\langle\!\langle u\rangle\!\rangle[x/u] : \tau}$$

where $(b,e,r) = (b_H + b_u, e_H + e_u + |M|, r_H + r_u - |M|)$.
By Lemma 8.9 $\triangleright_{lhd}\ \Gamma_0 + x{:}[\sigma_1] \vdash^{(b',e',r')} H\langle\!\langle x\rangle\!\rangle : \tau$ and $\triangleright_{lhd}\ \Delta_1 \vdash^{(b_1,e_1,r_1)} u : \sigma_1$, where $b_H = b' + b_1$, $e_H = e' + e_1$ and $r_H = r' + r_1 - 1$. Note that $x \notin \mathtt{fv}(u)$. We let $I = K \uplus \{1\}$ where $M = [\sigma_i]_{i\in K}$. We have necessarily $\Gamma_0 = \Gamma_0'; x{:}[\sigma_k]_{k\in K}$.

We remark that u has necessarily been typed with a (many) rule so that there are derivations $\Pi_k \vdash^{(b_k,e_k,r_k)} u : \sigma_k$ $(k \in K)$, such that $\Pi = \uplus_{k\in K}\Pi_k$, and $M = +_{k\in K}\sigma_k$ and $b_u = +_{k\in K}b_k$, $e_u = +_{k\in K}e_k$, $r_u = +_{k\in K}r_k$. By applying rule (many) again we obtain $\Pi + \Delta_1 \vdash^{(b_u+b_1,e_u+e_1,r_u+r_1)} u : M + [\sigma_1]$.
We can now construct the following derivation:

$$\frac{\Gamma_0'; x{:}[\sigma_i]_{i\in I} \vdash^{(b',e',r')} H\langle\!\langle x\rangle\!\rangle : \tau \qquad \Pi \uplus \Delta_1 \vdash^{(b_u+b_1,e_u+e_1,r_u+r_1)} u : [\sigma_i]_{i\in I}}{\Gamma_0' \uplus \Pi \uplus \Delta_1 \vdash^{(b'+b_u+b_1,e'+e_u+e_1+|I|,r'+r_u+r_1-|I|)} H\langle\!\langle x\rangle\!\rangle[x/u] : \tau}$$

We conclude since $b' + b_u + b_1 = b_H + b_u = b$, $e' + e_u + e_1 + |I| = e_H + e_u + |I| = e_H + e_u + |M| + 1 = e + 1$, $r' + r_u + r_1 - |I| = r_H + r_u - |M| = r$.

- All the inductive cases are straightforward. □

Theorem 8.11 (Tight completeness for *lhd*)**.** *Let $t \to_{lhd}^{k} p$, where $\mathtt{normal}_{lhd}(p)$. Then there exists a tight type derivation $\Phi \triangleright_{lhd}\ \Gamma \vdash^{(k_1,k_2,|p|_{lhd})} t : A$, where $k = k_1 + k_2$. Moreover, if $\mathtt{neutral}_{lhd}(p)$, then $A = \mathtt{neutral}$, and if $\mathtt{abs}_{lhd}(p)$ then $A = \mathtt{abs}$.*

Proof By induction on $t \to_{lhd}{}^{k} p$. If $k = 0$ then $t = p$. Proposition 8.8 gives the existence of a tight typing $\Phi \triangleright_{lhd} \vdash^{(b,e,r)}_{\mathtt{tight}} t$. Proposition 8.4 then gives $r = |t|_{lhd} = |p|_{lhd}$ and $b = e = 0$. The property then holds for $k_1 = k_2 = 0$.

Let $0 < k = k' + 1$ and $t \to_{lhd} u \to_{lhd}{}^{k'} p$. By *i.h.* there exists a tight typing derivation $\Psi \triangleright_{lhd} \vdash^{(k_1',k_2',|p|_{lhd})}_{\mathtt{tight}} u$, where $k' = k_1' + k_2'$. By quantitative subject expansion

Proposition 8.10 there exists a typing derivation Φ of u with the same types in the ending judgement of Ψ – then Φ is tight – and with indices $(k_1' + 1, k_2', |p|_{lhd})$ or $(k_1', k_2' + 1, |p|_{lhd})$.

In the first case, we let $k_1 = k_1' + 1$ and $k_2 = k_2'$, so that $k = 1 + k' =_{i.h.} 1 + k_1' + k_2' = k_1 + k_2$ as required. Moreover, $\Phi \triangleright_{lhd} \vdash^{(k_1,k_2,|p|_{lhd})}_{\texttt{tight}} t$.

In the second case, we let $k_1 = k_1'$ and $k_2 = k_2' + 1$, so that $k = 1 + k' =_{i.h.} 1 + k_1' + k_2' = k_1 + k_2$ as required. Moreover, $\Phi \triangleright_{lhd} \vdash^{(k_1,k_2,|p|_{lhd})}_{\texttt{tight}} t$. $\square$

https://doi.org/10.1017/S095679682000012X Published online by Cambridge University Press