Mathematical Structures in Computer Science (2025), **35**, e28, pp. 1–96
doi:10.1017/S0960129525100285

CAMBRIDGE UNIVERSITY PRESS

PAPER

A quantitative approach to global state composition

Sandra Alves[1,2], Delia Kesner[3] and Miguel Ramos[1,3,4]

[1]Faculdade de Ciências, Universidade do Porto, Rua do Campo Alegre, 4169-007, Porto, Portugal, [2]CRACS, INESC-TEC, DCC, Porto, Portugal, [3]Université Paris Cité, CNRS, IRIF, Paris, France and [4]LIACC, DCC, Porto, Portugal
Corresponding author: Miguel Ramos; Email: jmiguelsramos@gmail.com

(Received 19 March 2024; revised 3 June 2025; accepted 30 August 2025)

Abstract

We show that recent approaches to quantitative analysis based on non-idempotent typing systems can be extended to programming languages with effects. In particular, we consider two cases: the weak open call-by-name (CBN) and call-by-value (CBV) variants of the λ-calculus, equipped with operations to write and read from a global state. In order to capture quantitative information with respect to time and space for both CBN and CBV, we design for each of them a quantitative type system based on a (tight) multi-type system. One key observation of this work is how CBN and CBV influence the composition of state types. That is, each type system is developed by taking into account how each language manages the global state: in CBN, the composition of state types is almost straightforward, since function application does not require evaluation of its argument; in CBV, however, the interaction between functions and arguments makes the composition of state types more subtle since only values can be passed as actual arguments. The main contribution of this paper is the design of type systems capturing quantitative information about effectful CBN and CBV programming languages. Indeed, we develop type systems that are qualitatively and quantitatively sound and complete.

Keywords: Quantitative types; global state; effects; composition

1. Introduction

The main aim of this paper is to extend *quantitative* techniques based on *multi-types* to programs with *effects*.

Effectful programs

Programming languages produce different kinds of *effects* (observable interactions with the environment), such as raising and handling exceptions, reading/writing from/to a global memory, accessing a database or a file, performing pure non-deterministic or probabilistic choices, etc. The degree to which these side effects are visible to the programmer depends on each programming paradigm (Jones and Wadler, 1993): imperative programming usually makes invisible the distinction between pure computation and effects, while functional programming tries to keep this distinction explicit and clear. In general, dealing with side effects in a clear way facilitates the formal verification of programs, which in turn allows for the use of static analysis tools to ensure their correctness. For example, the functional programming language Haskell encapsulates side effects into *monadic* actions, a clean approach that is able to put in evidence the distinction between pure and effectful computations. This idea continues to attract growing attention. Indeed, for a

© The Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

pure function $f : A \to B$, which for any argument of type A returns a result of type B, it is possible to define an *effectful* function g that behaves like f but also produces some effect along the computation, in which case we can specify that $g : A \to T(B)$,[1] where $T(B)$ makes it clear that g returns a result of type B inside a useful wrapper represented by a monad that encapsulates the effect (Wadler, 1995). This approach allows programming languages to combine the best of both the imperative and declarative worlds: programs can produce effects, but effects are encoded in such a way that formal verification can still be performed using usual tools, by simply lifting them naturally into the monadic setting.

Quantitative properties

We address quantitative properties of programs with effects using *multi-types*, which originated in the theory of *intersection* type systems (Coppo and Dezani-Ciancaglini, 1978). They extend simple types with a new constructor $\cap$ in such a way that a program t is typable with $\sigma \cap \tau$ if t is typable with both types σ and τ, independently. Intersection types were first introduced as a model for capturing computational properties of the λ-calculus in a broader sense. For example, termination of different evaluation strategies can be characterized by typability in some appropriate intersection type system; that is, a program t is terminating if and only if t is typable. Naturally, intersection enjoys associativity (*i.e.*, $(\sigma \cap \tau) \cap \rho = \sigma \cap (\tau \cap \rho)$), commutativity (*i.e.*, $\sigma \cap \tau = \tau \cap \sigma$), and in particular, idempotency (*i.e.*, $\sigma \cap \sigma = \sigma$). Thus, an intersection of types can be naturally denoted by a set of types. By considering instead a *non-idempotent* intersection constructor (*i.e.*, $\sigma \cap \sigma \neq \sigma$), one naturally comes to represent an intersection type as a multiset, which is why non-idempotent intersection types (Gardner, 1994; Kfoury, 2000) are usually referred to as multi-types. Just like their idempotent precursors, multi-types capture interesting computational properties of the λ-calculus, but they also grant a substantial improvement: they provide *quantitative measures* about these properties (de Carvalho, 2007, 2018). For example, typability in a type system using non-idempotent intersections not only characterizes a *qualitative property*, such as termination for the λ-calculus, but also provides quantitative information such as *upper bounds* for the number of evaluation steps needed to reach a normal form. This shift of perspective, from idempotent to non-idempotent intersection types, goes beyond lowering the logical complexity of the proof: the quantitative information provided by typing derivations in the non-idempotent setting unveils crucial quantitative relations between typing (statics) and reduction (dynamics) of programs.

Upper bounds and exact split measures

Following the pioneering work by Gardner (1994), Kfoury (2000), and de Carvalho (2007, 2018), multi-types have been extensively used to reason about programming languages from a quantitative point of view. This is due to the fact that they are able to provide *upper bounds* for the number of steps needed to reach a normal form. However, a major drawback of this approach is the fact that the upper bound that is obtained actually bounds both the number of steps to reach a normal form and the size of this normal form. As it is well known, the size of a normal form can be exponentially bigger than the number of steps needed to reach that normal form, and thus, the resulting upper bound is not reasonable/interesting. More recently, it was shown (Accattoli et al., 2018, 2020) that it is indeed possible to extract better measures from a type system based on multi-types, thus providing *exact measures* instead of upper bounds. The key insight is to extract the measures only from *minimal type derivations*. This is achieved by introducing the key notion of *tightness* for type derivations. Very succinctly, minimal type derivations are obtained by imposing tightness conditions on the final judgment of type derivations. Moreover, by also splitting the set of typing rules so that terms that are consumed during evaluation (consuming) and terms that remain in the normal form (persistent) are typed differently, Accattoli et al. (2018, 2020) show

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

that it is not only possible to obtain exact measures but also *independent* ones (called *exact split* measures) for both the *number of steps* needed to reach a normal form and for its respective *size*. More precisely, quantitative typing systems[2] can be equipped with counters, in such a way that any term t is typable by a tight type derivation with counters b and s iff t reaches a normal form of size s in exactly b steps.

Evaluation strategies and open terms

Starting with Landin's seminal work (Landin, 1965), different incarnations of the λ-calculus continue to be developed and deeply studied in order to provide semantics for different programming languages and proof assistants. Two of the most well-known and well-studied variants of the λ-calculus can be found in Plotkin (1975): the call-by-value (CBV) and the call-by-name (CBN) λ-calculus. The CBV λ-calculus, in particular, has been used in the formalization of modern programming languages, such as the call discipline of C or the operational semantics of OCaml. And, while the original formulation of CBN λ-calculus is not as widely employed, its *memoized* variant, commonly referred to as call-by-need (Launchbury, 1993; Ariola and Felleisen, 1997), has been used to formalize the operational semantics of *lazy* programming languages such as Haskell. Evaluation in the CBN and CBV λ-calculi is *weak*: evaluation does not occur inside abstractions. Indeed, abstractions are used to denote functions, and programming languages do not evaluate the body of functions before all parameters are consumed. Another important aspect of functional programming languages is that programs are assumed to be *closed*, that is, contain no free variables. The CBV and CBN λ-calculi, however, are *open*. It has been established that the closed variant of the CBV λ-calculus is very well-behaved, since all normal forms are values; however, the open CBV λ-calculus gives rise to premature normal forms (see Accattoli and Guerrieri, 2016, 2022, for details). If we move from the realm of programming languages into that of proof assistants like Coq, then weak evaluation is not enough. In order to talk about equality of types in Coq's dependent type theory, strong evaluation, which relies on evaluation over open terms, is needed. In sum, and following along the lines of our previous work (Alves et al., 2023), we will consider *weak* and *open* versions of Plotkin's CBN and CBV λ-calculus so that our results can be easily adapted to the closed setting for both weak and strong evaluation. As shorthand, we will say *Weak Open CBN and CBV* whenever we mean *Weak Open CBN* and *Weak Open CBV*.

Contributions

This paper is about providing quantitative information for λ-calculi with effectful computations by means of intersection types. We focus in particular on a λ-calculus extended with (algebraic) operations for reading and writing on a global memory. In previous work (Alves et al., 2023), inspired by the monadic approach adopted in de'Liguoro and Treglia (2021), we designed a quantitatively sound and complete tight type system for a CBV evaluation strategy equipped with global memory. Soundness means that not only a tightly typable term t is always terminating but also that the tight type derivation of t provides exact (split) measures for both the number of β-steps and the number of memory accesses during the evaluation process. Moreover, the type system also provides the size of normal configurations. Dually for completeness. In this paper, we extend (Alves et al., 2023) both from an operational point of view and a denotational one.

Extending the operational semantics

In Alves et al. (2023), function applications were restricted to the syntactical form vt, where v is not an arbitrary term but a *value*. This follows the standard approach when considering Moggi's CBV translation of the λ-calculus, where applications are translated into CBV *let-expressions* (Moggi, 1989, 1991): if $(_)^v$ is the aforementioned translation, then $(tu)^v = (\texttt{let}\ \ x = u^v\ \ \texttt{in}\ \ t^v)$, which is

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

syntactic sugar for $(\lambda x.(\lambda y.xy)(u^v))(t^v)$ (Sabry and Wadler, 1997). But the application form vt is clearly not stable under CBN evaluation (*e.g.*, given $\mathtt{I} = \lambda z.z$, the term $(\lambda x.xy)(\mathtt{II})$ CBN evaluates to $(\mathtt{II})y$, which is not of the desired form), which means that we cannot take this representation when considering Plotkin's CBN λ-calculus. A possible solution would be to base our work on Moggi's metalanguage instead of the λ-calculus. In this work, we show that it is possible to take the standard syntax of the λ-calculus, extended with (algebraic) operations for reading and writing on a global memory, and reason not only about CBV (as in Alves et al. (2023)) but also CBN evaluation. For this, one needs to be careful when defining the operational semantics and the typing system according to each evaluation strategy. We leave the details to the later sections of this paper, but we will present some intuition at this moment. Just like in Alves et al. (2023), we will lift reduction over terms to reduction over *configurations*, which are pairs consisting of a term and a state. Let t be some term and s some state, then (t, s) is a configuration, and we write $(t, s) \to (t', s')$ to denote that term t reduces to some other term t' in one step and that the step changes the state from s to s' as a side effect. Let us focus on the behavior of applications over the following three scenarios:

(1) **Applications are of the form vu (recall that v is a value), and evaluation is CBV.** In this case, evaluation cannot occur on the left of applications but can occur on the right:

$$\frac{(v, s) \not\to \quad (u, s) \to (u', s')}{(vu, s) \to (vu', s')}$$

Therefore, state s changes with respect to the application vu in the same way as it changes with respect to u.

(2) **Applications are of the form tu, and evaluation is CBN.** In this case, evaluation cannot occur on the right of application but can only occur on the left:

$$\frac{(t, s) \to (t', s')}{(tu, s) \to (t'u, s')}$$

Therefore, state s changes with respect to the application tu in the same way as it changes with respect to t.

(3) **Applications are of the form tu, and evaluation is CBV.** In this case, evaluation should first be performed on the left as much as possible, and then it should continue on the right:

$$\frac{(t, s) \to (t', s')}{(tu, s) \to (t'u, s')} \qquad \frac{(t, s) \not\to \quad (u, s) \to (u', s')}{(tu, s) \to (tu', s')}$$

Therefore, state s changes with respect to the application tu in a *compositional* way. And we can see how the composition of states is essential to capture the evaluation of applications.

Defining reduction over configurations turns out to be crucial in order to capture the kind of results that we are interested in. Indeed, while it is possible to provide operational semantics for algebraic effects such as non-determinism, probabilistic choice, non-termination, and output, without mentioning any kind of external environment where effects occur – as was first done in Plotkin and Power (2001) and later explored in Gavazzo et al. (2024) – the same cannot be said for other effects, such as global state. This problem was immediately identified in Plotkin and Power (2001), but it was only in Plotkin and Power (2008) that the same authors provided a solution to this problem, by taking into account the coalgebraic structure of state. This then allowed them to provide an operational semantics for state based on configurations. In other words, in order to provide operational semantics for algebraic effects such as state, it is not enough to consider reduction over terms – one must consider reduction over configurations instead.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Extending the denotational semantics
In de'Liguoro and Treglia (2021) and Alves et al. (2023), the operational semantics is based on configurations, and the associated type systems turn out to be both sound and complete. This is in contrast with the more general approach to effectful computations in Gavazzo et al. (2024), where the operational semantics is not based on configuration, and thus, it cannot capture the particular case of the global state monad because subject expansion fails.

In this paper, we will consider an extended notion of state with a *log* (a mapping of locations to a *history* of values), which is inspired by Ahman and Uustalu (2013). This is not a new notion of state: its algebraic theory can already be found in Ahman and Uustalu (2013), and its coalgebraic properties (in the sense of Plotkin and Power (2008)) were studied in Ahman and Uustalu (2014). It turns out that the notion of state considered in Alves et al. (2023) can be easily recovered from this current proposal by simply ignoring the log (update history). Moreover, by keeping track of all updates to the state, we are now able to distinguish between configurations that were indistinguishable in de'Liguoro and Treglia (2021 and Alves et al. (2023).

Summary. In order to gradually present our ideas, we first introduce the *tight typing* techniques in Section 2 for *pure* Weak Open CBN and CBV; that is, no effects are produced during computation. Then, in Section 3, we apply these techniques to the effectful settings by lifting reduction and typing to configurations instead of terms and by adding operations for reading and writing on a global memory. In particular, Section 3.1.1 (resp. Section 3.2.1) introduces Weak Open CBN (resp. CBV) with Global State, and Section 3.1.2 (resp. Section 3.2.2) defines a quantitative type system capturing exact measures for the length of evaluation, number of state accesses, and size of normal forms. Quantitative soundness and completeness of the type system with respect to Weak Open CBN (resp. CBV) with global state are proved in Section 3.1.2 (resp. Section 3.2.2). We discuss related work in Section 4 and conclude in Section 5. In Appendix A, we include all auxiliary lemmas, as well as all the proofs in full detail. For now, we leave the proofs in full detail for the reviewers, but we intend to significantly shorten some straightforward proofs in order to gain space in the final version.

Preliminary general notations. We start with some general notations. Given a (one-step) reduction relation $\to_{\mathscr{R}}$, $\twoheadrightarrow_{\mathscr{R}}$ denotes the reflexive-transitive closure of $\to_{\mathscr{R}}$. Let $S_1, \ldots, S_n$ represent n different sets of possibly different kinds of reduction steps. We write $t \twoheadrightarrow_{\mathscr{R}}^{(I_1,\ldots,I_n)} u$, where $I_1, \ldots, I_n$ are positive integers, for a reduction sequence from t to u that takes I_1 steps using rules from set S_1, I_2 steps using rules from set S_2, . . ., and I_n steps using rules from set S_n. A term t is said to be (1) in $\mathscr{R}$-**normal form** or $\mathscr{R}$-**irreducible** (written $t \not\to_{\mathscr{R}}$) iff there is no u such that $t \to_{\mathscr{R}} u$ and (2) $\mathscr{R}$-**normalizing** iff there is some $\mathscr{R}$-normal form u such that $t \twoheadrightarrow_{\mathscr{R}} u$. The reduction relation $\mathscr{R}$ is normalizing iff every term is $\mathscr{R}$-normalizing. In general, reduction relations are non-deterministic. However, given any reduction relation $\to_{\mathscr{R}}$, it is always possible to consider a deterministic subset of $\to_{\mathscr{R}}$, which is usually referred to as an **evaluation strategy**. In particular, Plotkin's CBN and CBV variants of the λ-calculus are based on particular choices of evaluation strategies with respect to the (full) λ-calculus.

2. Weak Open CBN and CBV

In this paper, we use the *tight typing* technique in Accattoli et al. (2020) for obtaining quantitative bounds/measures, such as the length of reduction sequences to normal forms and the size of such normal forms. Originally, Accattoli et al. (2020) consider different evaluation strategies, such as head evaluation, leftmost-outermost evaluation, the maximal strategy, and linear head evaluation. In this paper, instead, we will consider *Weak Open* CBN and CBV. In particular, *pure* Weak Open CBN and CBV (*i.e.*, without effects) is first presented in this section, and then the general case is presented in Section 3.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\textbf{(Values)} \quad v, w ::= x \mid \lambda x.t$$
$$\textbf{(Terms)} \quad t, u ::= v \mid tu$$

Figure 1. The syntax of the λ-calculus.

$$\frac{}{(\lambda x.t)u \to t\{x\backslash u\}}\ (\beta_{\mathtt{n}}) \qquad \frac{}{(\lambda x.t)v \to t\{x\backslash v\}}\ (\beta_{\mathtt{v}})$$
$$\frac{t \to t'}{tu \to t'u}\ (\mathtt{appL}) \qquad \frac{t \not\to \quad u \to u'}{tu \to tu'}\ (\mathtt{appR})$$

Figure 2. Reduction rules for Weak Open CBN and CBV.

Weak Open CBN Size	**Weak Open CBV Size**
$\lvert x\rvert_{\mathtt{n}} := 0$	$\lvert x\rvert_{\mathtt{v}} := 0$
$\lvert \lambda x.t\rvert_{\mathtt{n}} := 0$	$\lvert \lambda x.t\rvert_{\mathtt{v}} := 0$
$\lvert tu\rvert_{\mathtt{n}} := 1 + \lvert t\rvert_{\mathtt{n}}$	$\lvert tu\rvert_{\mathtt{v}} := 1 + \lvert t\rvert_{\mathtt{v}} + \lvert u\rvert_{\mathtt{v}}$

Figure 3. Weak Open CBN and CBV size.

2.1 Syntax and operational semantics

We now introduce the syntax of the λ-calculus and the operational semantics for Weak Open CBN and CBV evaluations.

The syntax of the λ-calculus

The sets of **values** and **terms** of the λ-calculus are defined by the grammars in Figure 1. The sets of **free** and **bound** variables of terms are defined as usual, and syntactic equality of terms is considered up to α-conversion (Barendregt, 1985).

Evaluation in Weak Open CBN and CBV

Weak Open CBN and CBV evaluation notions are defined as subsets of the reduction rules in Figure 2. In particular, Weak CBN evaluation is denoted by $\to_{\mathtt{n}}$ and given by rules $(\beta_{\mathtt{n}})$ and (appL), and Weak CBV evaluation is denoted by $\to_{\mathtt{v}}$ and given by rules $(\beta_{\mathtt{v}})$ and (appL) and (appR).

Weak Open CBN and CBV size

The notions of **Weak Open CBN size** $\lvert t\rvert_{\mathtt{n}}$ and **Weak Open CBV size** $\lvert t\rvert_{\mathtt{v}}$ are defined in Figure 3. Since we are considering *weak* versions of CBN and CBV evaluation, both variables and abstractions have size zero. The difference is in applications: in CBN, only the left side of an application is evaluated, and thus, only this left side contributes to the overall size; in CBV, both sides of applications are evaluated, and thus, both sides contribute to the overall size.

Normal and neutral forms

In Figure 4, we present two grammars $\mathtt{no}_{\mathtt{n}}$ (CBN *normal* terms) and $\mathtt{no}_{\mathtt{v}}$ (CBV *normal* terms) that characterize all **normal forms** for Weak Open CBN and CBV, respectively. The key notion that

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

(Normal)	$\mathtt{no}_{\mathtt{n}} ::= \lambda x.t \mid \mathtt{ne}_{\mathtt{n}}$	$\mathtt{no}_{\mathtt{v}} ::= x \mid \lambda x.t \mid \mathtt{ne}_{\mathtt{v}}$
(Neutral)	$\mathtt{ne}_{\mathtt{n}} ::= x \mid \mathtt{ne}_{\mathtt{n}}\ t$	$\mathtt{ne}_{\mathtt{v}} ::= x\ \mathtt{no}_{\mathtt{v}} \mid \mathtt{no}_{\mathtt{v}}\ \mathtt{ne}_{\mathtt{v}} \mid \mathtt{ne}_{\mathtt{v}}\ \mathtt{no}_{\mathtt{v}}$

Figure 4. Normal and neutral terms for Weak Open CBN and CBV.

	Weak Open CBN	**Weak Open CBV**
(Tight Constants)	$\mathtt{tt}_{\mathtt{n}} ::= \mathtt{a} \mid \mathtt{n}$	$\mathtt{tt}_{\mathtt{v}} ::= \mathtt{v} \mid \mathtt{a} \mid \mathtt{n}$
(Multi-Types)	$\mathscr{M} ::= [\tau_i]_{i\in I}$	$\mathscr{M} ::= [\sigma_i]_{i\in I}$
(Types)	$\tau ::= \mathtt{tt}_{\mathtt{n}} \mid \mathscr{M} \to \tau$	$\tau ::= \mathtt{n} \mid \sigma \mid \mathscr{M}$
(Value Types)		$\sigma ::= \mathtt{v} \mid \mathtt{a} \mid \mathscr{M} \to \tau$

Figure 5. Types for Weak Open CBN and CBV.

allows us to define these grammars is that of *neutral* term. A neutral term is also a normal term; however, it has an additional property: it does not produce any new redexes when plugged into the left-hand side of some application. In Weak Open CBN evaluation, abstractions are the only normal terms that are not neutral. However, in Weak Open CBV evaluation, both abstractions and variables are (non-neutral) normal terms.

The following propositions, whose proofs are in Sections A.1.1 and A.1.2, respectively, show that the grammars for normal and neutral terms for Weak Open CBN and CBV in Figure 4 are sound and complete.

Proposition 1 (Normal forms for Weak Open CBN). *t be a term. Then $t \in \mathtt{no}_{\mathtt{n}}$ iff $t \not\to_{\mathtt{n}}$.*

Proposition 2 (Normal forms for Weak Open CBV). *Let t be a term. Then $t \in \mathtt{no}_{\mathtt{v}}$ iff $t \not\to_{\mathtt{v}}$.*

2.2 Quantitative types for Weak Open CBN and CBV

In this section, we present the quantitative type system for Weak Open CBN and CBV. Quantitative type systems are specified by *non-idempotent* intersection type systems (Gardner, 1994; Kfoury, 2000; de Carvalho, 2007), which are typing systems inspired by the resource-aware nature of linear logic. The latter was introduced by Girard (1987) as a refinement of intuitionistic (and classical) logic that highlights the role of formulas as resources. Due to Curry and Howard, we know that types and terms in the simply-typed λ-calculus correspond to proofs and propositions in (the implicational fragment of) intuitionistic logic, respectively. Therefore, one can look at Girard's translations of intuitionistic logic into linear logic (Girard, 1987) in order to obtain resource-aware versions for CBN and CBV.

In this section, we introduce all the necessary tools in order to present the quantitative type systems for Weak Open CBN and CBV.

Tight constants and multi-types

The types for Weak Open CBN and CBV are presented in Figure 5, where I is a finite set.

Notation. Let $\mathtt{tt}_0$ be any tight constant in $\mathtt{tt}_{\mathtt{n}}$ or $\mathtt{tt}_{\mathtt{v}}$. We will write $\overline{\mathtt{tt}_0}$ to denote the set $\mathtt{tt}_{\mathtt{n}} \setminus \mathtt{tt}_0$ or $\mathtt{tt}_{\mathtt{v}} \setminus \mathtt{tt}_0$, respectively. As an example, $\overline{\mathtt{a}}$ denotes $\{\mathtt{n}\}$ in Weak Open CBN and $\{\mathtt{v}, \mathtt{n}\}$ in Weak Open CBV.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Weak Open CBN

$$\frac{}{x : [\tau] \vdash^{(0,0)} x : \tau}\ (\mathtt{ax_n})$$

$$\frac{(\Gamma_i \vdash^{(b_i,s_i)} t : \tau_i)_{i\in I}}{+_{i\in I}\Gamma_i \vdash^{(+_{i\in I}b_i,+_{i\in I}s_i)} t : [\tau_i]_{i\in I}}\ (\mathtt{m_n})$$

$$\frac{\Gamma \vdash^{(b,s)} t : \tau}{\Gamma \setminus\!\setminus x \vdash^{(b,s)} \lambda x.t : \Gamma(x) \to \tau}\ (\lambda)$$

$$\frac{\Gamma \vdash^{(b,s)} t : \mathscr{M} \to \tau \qquad \Delta \vdash^{(b',s')} u : \mathscr{M}}{\Gamma + \Delta \vdash^{(1+b+b',s+s')} tu : \tau}\ (@)$$

$$\frac{}{\vdash^{(0,0)} \lambda x.t : \mathtt{a}}\ (\lambda_{\mathtt{p}})$$

$$\frac{\Gamma \vdash^{(b,s)} t : \mathtt{n}}{\Gamma \vdash^{(b,1+s)} tu : \mathtt{n}}\ (@_{\mathtt{p}})$$

Weak Open CBV

$$\frac{}{x : [\sigma] \vdash^{(0,0)} x : \sigma}\ (\mathtt{ax_v})$$

$$\frac{(\Gamma_i \vdash^{(b_i,s_i)} v : \sigma_i)_{i\in I}}{+_{i\in I}\Gamma_i \vdash^{(+_{i\in I}b_i,+_{i\in I}s_i)} v : [\sigma_i]_{i\in I}}\ (\mathtt{m_v})$$

$$\frac{\Gamma \vdash^{(b,s)} t : \overline{\mathtt{a}} \qquad \Delta \vdash^{(b',s')} u : \mathtt{tt_v}}{\Gamma + \Delta \vdash^{(b+b',1+s+s')} tu : \mathtt{n}}\ (@^1_{\mathtt{p}})$$

$$\frac{\Gamma \vdash^{(b,s)} t : \mathtt{tt_v} \qquad \Delta \vdash^{(b',s')} u : \mathtt{n}}{\Gamma + \Delta \vdash^{(b+b',1+s+s')} tu : \mathtt{n}}\ (@^2_{\mathtt{p}})$$

Figure 6. Typing rules for Weak Open CBN and CBV.

Tight constants are minimal types that are necessary to type normal forms in a tight (minimal) way. This means that the set of tight constants, which depends on the set of normal forms, will also depend on the underlying strategy. Terms in neutral form are typed with the neutral tight constant n. But, in order to type normal forms, we need to consider more tight constants:

- **Weak Open CBN.** The set of tight constants for Weak Open CBN consists of the tight constants a and n. The additional tight constant a is needed for Weak Open CBN, since abstractions are the only normal forms that are not neutral: applications of the form $(\lambda x.p)u$ are never normal forms.
- **Weak Open CBV.** The set of tight constants for Weak Open CBV consists of the tight constants v, a, and n. Indeed, variables that can be replaced by persistent abstractions are typed with tight constant a, and variables that can only be replaced by other variables are typed with tight constant v.

In Figure 5, the reader might notice that the sets of multi-types for both calculi are different. In Weak Open CBN, multi-types contain the n tight constant because variables can be replaced with any term, in particular, neutral terms (of type n). In Weak Open CBV, instead, multi-types do not contain any tight constant n because variables can only be replaced by values (of type v or a). Moreover, because in the open setting abstractions can return values and terms in neutral form, neutral tight constants can appear on the right of arrow types in both Weak Open CBN and CBV.

Typing systems for Weak Open CBN and CBV

The type systems for Weak Open CBN and CBV are defined as subsets of the typing rules in Figure 6. Let us introduce some necessary definitions:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- **Typing environments** are denoted by Γ, Δ and are defined as functions from variables to multi-types that assign the empty multi-type $[\,]$ to all but a finite set of variables. The **domain** of a typing environment Γ, denoted by $\mathtt{dom}(\Gamma)$, is defined as $\mathtt{dom}(\Gamma) := \{x \mid \Gamma(x) \neq [\,]\}$. The **union** of typing environments Γ and Δ is denoted as $\Gamma + \Delta$ and defined as $(\Gamma + \Delta)(x) = \Gamma(x) \sqcup \Delta(x)$, where $\sqcup$ denotes *multi-set union*. An example is $(x : [\tau_1], y : [\tau_2]) + (x : [\tau_1], z : [\tau_2]) = (x : [\tau_1, \tau_1], y : [\tau_2], z : [\tau_2])$. This notion is extended to a finite union of environments, written $+_{i \in I} \Gamma_i$ (the empty environment is obtained when $I = \emptyset$). We write $\Gamma \setminus\!\setminus x$ for the environment $(\Gamma \setminus\!\setminus x)(x) = [\,]$ and $(\Gamma \setminus\!\setminus x)(y) = \Gamma(y)$ if $y \neq x$, and we write $\Gamma; x : \mathscr{M}$ for $\Gamma + (x : \mathscr{M})$, when $x \notin \mathtt{dom}(\Gamma)$. Notice that Γ and $\Gamma; x : [\,]$ are different notations denoting the same environment.
- **Typing judgments** have the form $\Gamma \vdash^{(b,s)} t : \tau$, where Γ is a *typing environment*, τ is a type, and (b, s) is a counter made by two natural numbers (the meaning of b and s will be made precise later). Also, if the typing environment Γ is empty, we will sometimes write $\vdash^{(b,s)} t : \tau$ instead of $\emptyset \vdash^{(b,s)} t : \tau$.
- **Typing rules** are presented in the usual Gentzen-style layout: the typing judgments at the top of the rule are called the *premises*; the unique typing judgment at the bottom is its *conclusion*.
- We write $\triangleright\, \Gamma \vdash_{\mathtt{n}}^{(b,s)} t : \tau$ (resp. $\triangleright\, \Gamma \vdash_{\mathtt{v}}^{(b,s)} t : \tau$) if there is a **type derivation** of the judgment $\Gamma \vdash^{(b,s)} t : \tau$ (resp. $\Gamma \vdash^{(b,s)} t : \tau$) using the typing rules for Weak Open CBN (resp. CBV), in which case we say that the term t of its conclusion is **typable in Weak Open CBN** (resp. **CBV**). We use letters $\Phi, \Psi, \ldots$ to name type derivations, by writing, for example, $\Phi \triangleright\, \Gamma \vdash_{\mathtt{v}}^{(b,s)} t : \tau$.

We distinguish between *consuming* and *persistent* rules by subscripting the name of the latter with a p. The intuitions behind the rules are the following (for now, ignore the counters (b, s)):

- **Rules for Weak Open CBN:**
  - **Rules $(\mathtt{ax}_{\mathtt{n}})$, (λ), $(@)$, and $(\mathtt{m}_{\mathtt{n}})$.** These rules are the usual ones for CBN.
  - **Rule $(\lambda_{\mathtt{p}})$.** This rule is used to type with a tight constant $\mathtt{a}$, abstractions that persist during evaluation.
  - **Rule $(@_{\mathtt{p}})$.** This rule is used to type application terms of the form tu with the tight constant $\mathtt{n}$, whenever the application constructor persists during evaluation.
- **Rules for Weak Open CBV:**
  - **Rules $(\mathtt{ax}_{\mathtt{v}})$, (λ), and $(@)$.** These rules are the usual ones for CBV, with the slightly less standard presentation of $(@)$ using the fact that $!(A \to B) = !(A) \to !(B)$.
  - **Rule $(\mathtt{m}_{\mathtt{v}})$.** This rule is the exponential rule from linear logic, but, as stated before, it is introduced in the Weak Open CBV system in a restricted form: it can only be applied to values.
  - **Rule $(\lambda_{\mathtt{p}})$.** Just like for Weak Open CBN, this rule is used to type with tight constant $\mathtt{a}$, abstractions that will persist during evaluation.
  - **Rules $(@^1_{\mathtt{p}})$ and $(@^2_{\mathtt{p}})$.** These rules are actually *rule schemas* that depend on the particular choice of tight constants. In particular, rule $(@^1_{\mathtt{p}})$ unfolds into six different rules according to the particular choices of tight constants in $\overline{\mathtt{a}}$ and $\mathtt{tt}_{\mathtt{v}}$, and $(@^2_{\mathtt{p}})$ unfolds into three different rules according to the choice of tight constant in $\mathtt{tt}_{\mathtt{v}}$. Rules $(@^1_{\mathtt{p}})$ and $(@^2_{\mathtt{p}})$ are used to type application terms of the form tu with the tight constant $\mathtt{n}$, whenever the application constructor persists during evaluation. The rule depends on the types of the subterms t and u.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Counters
In order to keep track of the number of $\beta_{\mathtt{n}}$-steps and $\beta_{\mathtt{v}}$-steps, as well as the Weak Open CBN and CBV size of normal forms, typing judgments are decorated with counters of the form (b, s). Let $\Phi \triangleright \Gamma \vdash^{(b,s)} t : \tau$ be a typing derivation for term t, then b is the number of $\beta_{\mathtt{n}}$-steps (resp. $\beta_{\mathtt{v}}$-steps) needed to normalize t following the respective strategy, and s is the size of the corresponding CBN (resp. CBV) normal form of t.

- **Weak Open CBN.** Rule ($@$) is the only rule that increases the counter b for the number of $\beta_{\mathtt{n}}$-steps. Rule ($@_{\mathtt{p}}$) is the only rule that increases the counter s for the size of the Weak Open CBN normal form.
- **Weak Open CBV.** Rule ($@$) is the only rule that increases the counter b for the number of $\beta_{\mathtt{v}}$-steps. Rules ($@^1_{\mathtt{p}}$) and ($@^2_{\mathtt{p}}$) are the only rules that increase the counter s for the size of the Weak Open CBV normal form.

We call ($@$) a *consuming* application rule, and rules ($@^1_{\mathtt{p}}$), ($@^2_{\mathtt{p}}$), and ($@_{\mathtt{p}}$), *persistent* application rules. The intuition is the following:

- **Weak Open CBN.** Consuming application rules are used to type applications that evaluate to the form $(\lambda x.t)u$, and persistent application rules are used to type applications that evaluate to the form $\mathtt{ne}_{\mathtt{n}} t$.
- **Weak Open CBV.** Consuming application rules are used to type applications that evaluate to the form $(\lambda x.t)v$, and persistent application rules are used to type applications that evaluate to the form $\mathtt{ne}_{\mathtt{v}}$.

This difference of behavior between persistent and consuming rules is captured by the rules as a tightness (minimality) constraint. If an application is *not* consuming, we only need to know that the terms on the left and right of the application evaluate to terms that do not interact with each other (have tight types) and will persist as an application (with a tight type) during evaluation. However, if an application is consuming, we need to know more about its form. In particular, that the term on the left of the application evaluates to an abstraction (of type $\mathscr{M} \to \tau$) that expects an argument of a certain form (of type $\mathscr{M}$) and that after consuming that argument, its body will have a particular form (of type τ).

Tightness. The same term may have different typing derivations. These will always provide *upper bounds* for the number of steps and the size of normal forms. The interesting aspect of tight type systems, however, is that the class of typing derivations that provide exact bounds for these measures is captured by *tight derivations*. The notion of tightness for types, typing environments, type judgments, and typing derivations is as follows:

- A type τ is **tight** if τ is a tight constant (*i.e.*, $\tau \in \mathtt{tt}_{\mathtt{v}}$ for CBV and $\tau \in \mathtt{tt}_{\mathtt{n}}$ for CBN).
- A multi-type $\mathscr{M}$ is **tight** if, for all $\sigma \in \mathscr{M}$, σ is tight.
- A typing environment Γ is **tight** if, for all $x \in \mathtt{dom}(\Gamma)$, $\Gamma(x)$ is tight.
- A typing judgment $\Gamma \vdash^{(b,s)} t : \tau$ is **tight** if Γ and τ are tight.
- A type derivation Φ is **tight** if its conclusion is tight.

Soundness and completeness. The type system for Weak Open CBN (resp. CBV) is *quantitatively sound* and *complete* with respect to $\to_{\mathtt{n}}$ (resp. $\to_{\mathtt{v}}$). *Quantitative soundness* means that not only a *tightly* typable term t is terminating (*qualitative*) but also that exact and split measures concerning the evaluation of t to normal form (*quantitative*) can be extracted from its *tight* type derivation. More precisely, if $\Phi \triangleright \Gamma \vdash^{(b,s)}_{\mathtt{n}} t : \tau$ (resp. $\Phi \triangleright \Gamma \vdash^{(b,s)}_{\mathtt{v}} t : \tau$) is tight, then there exists $u \in \mathtt{no}_{\mathtt{n}}$ (resp.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$\mathtt{no}_{\mathtt{v}}$) such that $t \twoheadrightarrow_{\mathtt{n}}^{b} u$ (resp. $t \twoheadrightarrow_{\mathtt{v}}^{b} u$) with $|u|_{\mathtt{n}} = s$ (resp. $|u|_{\mathtt{v}} = s$). Dually for *quantitative completeness*. Most of the properties needed for quantitative soundness and completeness are shared between the two type systems. For this reason, we are going to present each statement only once and expect the reader to take the absence of any distinction between the two strategies as an indication of such between the statements. Also, we are going to present, in what follows, only the key properties and leave the auxiliary properties to Appendix A.1, along with their proofs.

To show quantitative soundness and completeness, we need to show that a term t is in Weak Open CBN (resp. CBV) normal form if, and only if, the first counter, which corresponds to the number of $\beta_{\mathtt{n}}$-steps (resp. $\beta_{\mathtt{v}}$-steps) needed to reach a normal form, is zero.

Lemma 1 (Normal forms take zero steps). *Let* $\Phi \rhd \Gamma \vdash^{(b,s)} t : \tau$ *be a tight type derivation. Then,* $b = 0$ *iff* $t \in \mathtt{no}$.

We also need to show that, given a term t, the type system infers the correct Weak Open CBN (resp. CBV) size of the normal form of t.

Lemma 2 (Normal forms have correct size). *Let* $\Phi \rhd \Gamma \vdash^{(b,s)} t : \tau$ *be a tight type derivation. If* $b = 0$, *then* $s = |t|$.

Finally, we show that, for any term t in Weak Open CBN (resp. CBV) normal form, there exists a *tight derivation*, which, moreover, infers the correct Weak Open CBN (resp. CBV) size of t.

Lemma 3 (All normal forms are tightly typable). *If* $t \in \mathtt{no}$, *then there exists a tight type derivation* $\Phi \rhd \Gamma \vdash^{(b,s)} t : \tau$, *such that* $s = |t|$.

We now state a quantitative form of type preservation along evaluation.

Lemma 4 (Quantitative subject reduction). *Let* $\Phi_t \rhd \Gamma \vdash^{(b,s)} t : \tau$ *be a tight type derivation. If* $t \to t'$, *then there exists* $\Phi_{t'} \rhd \Gamma \vdash^{(b-1,s)} t' : \tau$.

Quantitative soundness follows from the correctness of the counters for normal forms (Lemma 1 and Lemma 2) and quantitative subject reduction (Lemma 4).

Theorem 1 (Quantitative soundness). *If* $\Phi \rhd \Gamma \vdash^{(b,s)} t : \tau$ *is a tight type derivation, then there exists* $u \in \mathtt{no}$ *such that* $t \twoheadrightarrow^{b} u$ *with* $|u| = s$.

The following property guarantees that typability is preserved back along evaluation.

Lemma 5 (Quantitative subject expansion). *Let* $\Phi_{t'} \rhd \Gamma \vdash^{(b,s)} t' : \tau$ *be a tight type derivation. If* $t \to t'$, *then there exists* $\Phi_t \rhd \Gamma \vdash^{(b+1,s)} t : \tau$.

Quantitative completeness follows from the typability of all normal forms in a tight way (Lemma 3) and quantitative subject expansion (Lemma 5).

Theorem 2 (Quantitative completeness). *If* $t \twoheadrightarrow^{b} u$ *with* $u \in \mathtt{no}$, *then there exists a tight type derivation* $\Phi_t \rhd \Gamma \vdash^{(b,|u|)} t : \tau$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\textbf{(Terms)}\ t, u, p ::= x \mid \lambda x.t \mid tu \mid \mathtt{get}_l(\lambda x.t) \mid \mathtt{set}_l(t, u)$$
$$\textbf{(States)}\quad s, q ::= \varepsilon \mid \mathtt{upd}_l(t, s)$$
$$\textbf{(Configurations)}\quad c ::= (t, s)$$

Figure 7. The extended syntax of the λ-calculus for Weak Open CBN with Global State.

3. Weak Open CBN and CBV with Global State

Now that we have seen how to use *tight typing techniques* to obtain quantitative information in the Weak Open CBN and CBV variants of the λ-calculus without effects, we are ready to talk about the extensions of these languages with (algebraic) operations for reading and writing on a global state. To do this, we will lift reduction over terms to reduction over *configurations*, which are pairs consisting of a term and a state. Then, we are going to develop two types of systems: one for Weak Open CBN with Global State and another for Weak Open CBV with Global State. Since the CBN type system is, in a way, dual to the one in Alves et al. (2023) for CBV, we are going to start by presenting the type system for Weak Open CBN with Global State first and then move to the generalization of the CBV type system in Alves et al. (2023).

3.1 Weak Open CBN with Global State

We start by extending the syntax presented in Section 2 with operations for reading and writing on a global state and introducing the syntax for the global state and configurations. Then, also based on the CBN language introduced in Section 2, we show how to lift reduction on terms to reduction on configurations and how operations on terms interact with the global state.

3.1.1 Syntax and operational semantics

We introduce the syntax for the extension of the Weak Open CBN λ-calculus with Global State and its operational semantics.

The syntax for Weak Open CBN with Global State. We consider a denumerable set of locations denoted by $l, l', \ldots$. The sets of **terms**, **states**, and **configurations** are presented in Figure 7.

The set of terms of the λ-calculus is extended with two operations: $\mathtt{get}_l(\lambda x.t)$ and $\mathtt{set}_l(t, u)$, where l is a location drawn from some set of location names. The former is used to *get* the term at location l and bind it to x in t. The latter is used to *set* a term t to location l in the state and then continue with term u. States are either the **empty state**, denoted by ε, or the result of some sequence of *update* operations over the empty state. Given some state s, an update over s consists in setting some term t to some location l, and it is denoted by $\mathtt{upd}_l(t, s)$. If the same location l is updated twice in a row, for example, $\mathtt{upd}_l(t, \mathtt{upd}_l(u, s))$, only t (the most recent update) should be accessible at location l according to our operational semantics in Figure 9. Configurations are pairs consisting of a term and a state. The sets of **free** and **bound** variables of terms, states, and configuration are as expected.

It is important to note that, if $\mathtt{upd}_l(t, \mathtt{upd}_l(u, s))$ is assumed to be equal to $\mathtt{upd}_l(t, s)$, as in de'Liguoro and Treglia (2021) and Alves et al. (2023), this corresponds to presenting the global state as a mapping from locations to terms, that is, the coalgebraic structure for the state monad (Plotkin and Power, 2008). We assume instead that if $\mathtt{upd}_l(t, \mathtt{upd}_l(u, s))$ *is not* equal to $\mathtt{upd}_l(t, s)$, this corresponds to presenting the global state as a mapping from locations to *lists* of terms, that is, the coalgebraic structure for the state-logging monad (Ahman and Uustalu, 2014). We also consider the update operation to be commutative in the following sense:

$$\mathtt{upd}_l(t, \mathtt{upd}_{l'}(u, s)) \equiv_{\mathrm{c}} \mathtt{upd}_{l'}(u, \mathtt{upd}_l(t, s)) \text{ if } l \neq l'$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Weak Open CBN Size Extended for Global State

$$
\begin{aligned}
&\cdots\\
|\mathtt{get}_l(\lambda x.t)|_{\mathtt{n}} &:= |t|_{\mathtt{n}}\\
|\mathtt{set}_l(t,u)|_{\mathtt{n}} &:= |u|_{\mathtt{n}}\\
|s|_{\mathtt{n}} &:= 0\\
|(t,s)|_{\mathtt{n}} &:= |t|_{\mathtt{n}}
\end{aligned}
$$

Figure 8. Weak Open CBN size extended for global state.

$$
\frac{}{((\lambda x.t)u,s) \rightsquigarrow_{\beta_{\mathtt{n}}} (t\{x\backslash u\},s)}\ (\beta_{\mathtt{n}}) \qquad \frac{(t,s) \rightsquigarrow_{\mathtt{r}} (t',q) \qquad \mathtt{r} \in \{\beta,\mathtt{g},\mathtt{s}\}}{(tu,s) \rightsquigarrow_{\mathtt{r}} (t'u,q)}\ (\mathtt{appL})
$$

$$
\frac{s \equiv \mathtt{upd}_l(u,q)}{(\mathtt{get}_l(\lambda x.t),s) \rightsquigarrow_{\mathtt{g}} (t\{x\backslash u\},s)}\ (\mathtt{get}) \qquad \frac{}{(\mathtt{set}_l(u,t),s) \rightsquigarrow_{\mathtt{s}} (t,\mathtt{upd}_l(u,s))}\ (\mathtt{set})
$$

Figure 9. Reduction rule for Weak Open CBN with Global State.

The equivalence relation generated by the algebraic equation $\equiv_{\mathtt{c}}$ is denoted by $\equiv$. We write $l \in \mathtt{labels}(s)$, if $s \equiv \mathtt{upd}_l(t,q)$, for some term t and state q. Moreover, for every location l, these t and q are *unique*. A consequence of the equivalence relation $\equiv$ on states is that, whenever we want to refer to the most up-to-date content of a location in a given state s, we can always write s so that the first update operation is the latest one over that particular location. Said differently, for every $l \in \mathtt{labels}(s)$, $s \equiv \mathtt{upd}_l(u,q)$, for some term u and some state q. From now on, if $s \equiv \mathtt{upd}_l(u,q)$, we write $s(l)$ to denote the (unique) term u.

Weak Open CBN size. The notion of **Weak Open CBN size** is extended to operations, states, and configurations according to Figure 8.

Evaluation in Weak Open CBN with Global State. Weak Open CBN evaluation $\rightarrow_{\mathtt{n}}$ is extended to global state by lifting it to configurations and taking into consideration the interaction between `get` and `set` operations and the global state. Evaluation for Weak Open CBN with Global State $\rightsquigarrow_{\mathtt{n}}$ is defined as the union of the reduction rules in Figure 9.

Rules $(\beta_{\mathtt{n}})$ and (appL) are the same as for the Weak Open CBN, if we take only the term part of the configuration. Rules (get) and (set) describe how the `get` and `set` operations interact with the global state. In particular, notice that, while rule (set) can always be fired, rule (get) can get stuck if it tries to interact with a location on the global state that is empty.

Blocked configurations. A configuration (t,s) is said to be **blocked** if: $t = \mathtt{get}_l(\lambda x.u)$ and $l \notin \mathtt{labels}(s)$; or $t = up$ and (u,s) is blocked. We say that (t,s) is an **unblocked** otherwise.

Example 1. *Let l, l' be two locations such that $l \neq l'$. Then, the three following configurations are all blocked:*

$$(\mathtt{get}_l(\lambda x.x),\varepsilon) \quad ((\mathtt{get}_l(\lambda x.x))(\lambda y.y),\mathtt{upd}_{l'}(t,\varepsilon)) \quad (((\mathtt{get}_l(\lambda x.x))(\lambda y.y))(\lambda z.z),\mathtt{upd}_{l'}(t,\varepsilon))$$

Normal configurations. Clearly, blocked configurations are a kind of normal form for configurations. However, they are not the desired result of a computation – even though they are in normal

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\textbf{(Normal)} \qquad \mathtt{no_n} ::= \lambda x.t \mid \mathtt{ne_n}$$
$$\textbf{(Neutral)} \qquad \mathtt{ne_n} ::= x \mid \mathtt{ne_n}\ t$$

Figure 10. Normal and neutral terms for Weak Open CBN with Global State.

Let S be the type of global states and $T(X) = S \Rightarrow (X \times S)$ stand for the global state monad.

(Types)

$$\mathtt{tt_n}^\dagger := \mathtt{tt_n}$$
$$([\tau_i]_{i\in I})^\dagger := [T(\tau_i^\dagger)]_{i\in I}$$
$$(\mathscr{M} \to \tau)^\dagger := (\mathscr{M}^\dagger \to T(\tau^\dagger))$$

(Typing Environments)

$$(x_1 : \mathscr{M}_1, \ldots, x_n : \mathscr{M}_n)^\dagger := (x_1 : \mathscr{M}_1^\dagger, \ldots, x_n : \mathscr{M}_n^\dagger)$$

(Typing Judgements for Terms)

$$(\Gamma \vdash t : \tau)^\dagger := \Gamma^\dagger \vdash t : T(\tau^\dagger)$$
$$(\Gamma \vdash t : \mathscr{M})^\dagger := \Gamma^\dagger \vdash t : \mathscr{M}^\dagger$$

Figure 11. The CBN monadic translation for types, typing environments, and judgments.

form – but rather a runtime error that should be caught by the type system (*i.e.*, blocked configurations will be untyped). A configuration (t, s) is said to be **normal** if it is a blocked configuration or $t \in \mathtt{no_n}$, where $\mathtt{no_n}$ is defined in Figure 10.

Example 2. *Consider the configuration* $c_0 = ((\lambda x.\mathtt{set}_l(z, x))(\mathtt{get}_l(\lambda y.y)), \varepsilon)$. *Then we can reach a normal configuration as follows:*

$$\begin{aligned}((\lambda x.\mathtt{set}_l(z, x))(\mathtt{get}_l(\lambda y.y)), \varepsilon) &\rightsquigarrow_{\beta_\mathtt{n}} (\mathtt{set}_l(z, \mathtt{get}_l(\lambda y.y)), \varepsilon)\\ &\rightsquigarrow_\mathtt{s} (\mathtt{get}_l(\lambda y.y), \mathtt{upd}_l(z, \varepsilon))\\ &\rightsquigarrow_\mathtt{g} (z, \mathtt{upd}_l(z, \varepsilon))\end{aligned}$$

The following proposition, whose proof is in Appendix A.2, shows that normal configurations are indeed irreducible.

Proposition 3. *Let* (t, s) *be a configuration. Then* (t, s) *is normal iff* $(t, s) \not\rightsquigarrow_\mathtt{n}$.

3.1.2 A quantitative type system for Weak Open CBN with Global State

In this section, we present the quantitative type system for Weak Open CBN with Global State. For this, we start by introducing the CBN monadic translation of the λ-calculus of Wadler (1990) by adapting it to account for tight constants and multi-types. We then instantiate this monadic translation with the monad for global state.

The CBN monadic translation. In order to lift the type system for Weak Open CBN into the global state monad, we are going to follow the CBN translations of the simply-typed λ-calculus in Wadler (1990). Naturally, we will have to slightly adapt the translation in order to accommodate for multi-types. The resulting translation of the Weak Open CBN types in Figure 5 is presented in Figure 11.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

(Tight Constants)	$\mathtt{tt_n} ::= \mathtt{a} \mid \mathtt{n}$
(Multi-types)	$\mathscr{M} ::= [\mu_i]_{i\in I}$ where I is a finite set
(Types)	$\tau ::= \mathtt{tt_n} \mid \mathscr{M} \to \mu$
(Monadic Types)	$\mu ::= \mathscr{T} \Rightarrow \kappa$
(Configuration Types)	$\kappa ::= \tau \times \mathscr{T}$
(List Multi-types)	$\mathscr{L} ::= \star \mid \mathscr{M} \cdot \mathscr{L}$
(State Types)	$\mathscr{T} ::= \{l_i : \mathscr{L}_i\}_{i\in I}$ where all l_i are distinct

Figure 12. Types for Weak Open CBN with Global State.

This figure deserves some comments. Notice that we choose to use two different arrows to denote our types: the first one $\Rightarrow$ is used for the global state monad, while the second one $\to$ is the usual functional arrow type. Also, the T translation is meant to be understood as a template for building the type system.

State and monadic (multi-)types. The grammar for types is presented in Figure 12.

State types, denoted by $\mathscr{T}$, can be seen as partial functions mapping all but a finite set of locations to the empty list multi-type $\star$. Notice that $\mathscr{T}$ is the set of all state types and corresponds to S in Figure 11. The **domain** of a state type $\mathscr{T}$ is denoted by $\mathtt{dom}(\mathscr{T})$ and defined as $\mathtt{dom}(\mathscr{T}) := \{l \mid (l : \mathscr{L}) \in \mathscr{T} \text{ and } \mathscr{L} \neq \star\}$. We write $\mathscr{T}(l)$ to denote the list multi-type associated with location l in $\mathscr{T}$, whenever $l \in \mathtt{dom}(\mathscr{T})$. To simplify the notation, we will omit $\star$ whenever a list multi-type is not empty; thus, for example, $[\mu] \cdot \star$ will be simply written as $[\mu]$. It is important to remark that the use of list multi-types is crucial in order to match the behavior exhibited by our notion of state; for example, if an empty location l is updated twice, first with a term of type $\mathscr{M}_1$ and then with a term of type $\mathscr{M}_2$, then it will be assigned type $\mathscr{M}_2 \cdot (\mathscr{M}_1 \cdot \star)$.

We write $\mathscr{T}; \mathscr{T}'$ to mean that $\mathtt{dom}(\mathscr{T}) \cap \mathtt{dom}(\mathscr{T}') = \emptyset$. Notice that $\{l : \star\}; \mathscr{T} = \mathscr{T}$ and $\{l : \mathscr{L}\}; \mathscr{T} \neq \mathscr{T}$ if $\mathscr{L} \neq \star$, so in particular $\{l : [\,]\}; \mathscr{T} \neq \mathscr{T}$. However, recall that $x : [\,]; \Gamma = \Gamma$.

Multi-types are now multi-sets of monadic types. And **monadic types** are lifted types of the form $\mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, where the state type $\mathscr{T}$ is called an **input** state, and $\tau \times \mathscr{T}'$ is a **configuration type** consisting of a type τ and a state type $\mathscr{T}'$ called **output** state. When dealing with effects, it is important to distinguish between pure and monadic types. Indeed, **pure types** are monadic types that have the same input and output state type, for example, $\mathscr{T} \Rightarrow (\tau \times \mathscr{T})$.

Typing system for Weak Open CBN with Global State. The type system for Weak Open CBN with Global State is defined by the typing rules in Figure 13.

Like in the case for Weak Open CBN and CBV, we now present some necessary definitions:

- The notions of **typing environments** are extended to global state as expected.
- **Typing rules** follow the structure introduced in the previous section.
- **Typing judgments** are of three different kinds:
  - **Term typing judgments** are used to type *terms* and are of the form $\Gamma \Vdash^{(b,m,d)} t : \mathscr{G}$, where (b, m, d) is a counter, Γ is a *typing context*, and $\mathscr{G}$ represents either a *multi-type* $\mathscr{M}$ or a *monadic type* μ.
  - **State typing judgments** are used to type *states* and are of the form $\Delta \Vdash^{(b,m,d)} s : \mathscr{T}$, where (b, m, d) is a counter, Δ is a *typing context*, and $\mathscr{T}$ a *state type*.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Rules for Terms

$$\frac{}{x : [\mu] \Vdash^{(0,0,0)} x : \mu}\ (\mathtt{ax}) \qquad \frac{\Gamma \Vdash^{(b,m,d)} t : \mu}{\Gamma \setminus\!\setminus x \Vdash^{(b,m,d)} \lambda x.t : \mathscr{T} \Rightarrow ((\Gamma(x) \to \mu) \times \mathscr{T})}\ (\lambda)$$

$$\frac{(\Gamma_i \Vdash^{(b_i,m_i,d_i)} t : \mu_i)_{i\in I}}{+_{i\in I}\Gamma_i \Vdash^{(+_{i\in I}b_i,+_{i\in I}m_i,+_{i\in I}d_i)} t : [\mu_i]_{i\in I}}\ (\mathtt{m})$$

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}') \qquad \Delta \Vdash^{(b',m',d')} u : \mathscr{M}}{\Gamma + \Delta \Vdash^{(1+b+b',m+m',d+d')} tu : \mathscr{T} \Rightarrow \kappa}\ (@)$$

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}{\Gamma \setminus\!\setminus x \Vdash^{(b,1+m,d)} \mathtt{get}_l(\lambda x.t) : \{l : \Gamma(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}\ (\mathtt{get})$$

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{M} \qquad \Delta \Vdash^{(b',m',d')} u : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}{\Gamma + \Delta \Vdash^{(b+b',1+m+m',d+d')} \mathtt{set}_l(t,u) : \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}\ (\mathtt{set})$$

Rules for Persistent Terms

$$\frac{}{\emptyset \Vdash^{(0,0,0)} \lambda x.t : \mathscr{T} \Rightarrow (\mathtt{a} \times \mathscr{T})}\ (\lambda_{\mathtt{p}}) \qquad \frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}{\Gamma \Vdash^{(b,m,1+d)} tu : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}\ (@_{\mathtt{p}})$$

Rules for States

$$\frac{}{\emptyset \Vdash^{(0,0,0)} \varepsilon : \emptyset}\ (\mathtt{emp}) \qquad \frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{M} \qquad \Delta \Vdash^{(b',m',d')} s : \{l : \mathscr{L}\}; \mathscr{T}}{\Gamma + \Delta \Vdash^{(b+b',m+m',d+d')} \mathtt{upd}_l(t,s) : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}}\ (\mathtt{upd})$$

Rule for Configurations

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{T} \Rightarrow \kappa \qquad \Delta \Vdash^{(b',m',d')} s : \mathscr{T}}{\Gamma + \Delta \Vdash^{(b+b',m+m',d+d')} (t,s) : \kappa}\ (\mathtt{conf})$$

Figure 13. Typing rules for Weak Open CBN with Global State.

  - **Configuration typing judgments** are used to type *configurations* and are of the form $\Gamma \Vdash^{(b,m,d)} (t,s) : \kappa$, where (b,m,d) is a counter, Γ is a *typing context*, and κ is a *configuration type*.

  The meaning of the counters b, m, and s will again be made precise later.
- The notions of **type derivation** and **typability** are extended to global state as expected.

We distinguish between *consuming* and *persistent* rules just like before. The intuitions behind the rules are the following (again, please ignore the counters (b,m,d) for now):

- **Rules (ax), (λ), ($\lambda_{\mathtt{p}}$), ($@_{\mathtt{p}}$), and (m).** These rules are essentially the same as for Weak Open CBN, but types are now replaced by monadic types.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- **Rule (@).** In this rule, we can see how CBN influences the type system. Since, according to CBN, we only reduce on the left-hand side of applications, only the term t on the left can (potentially) change the global state. Therefore, CBN tells us the following: (1) t is reduced until reaching an abstraction $\lambda x.p$, going from input state type $\mathscr{T}$ to an output state type $\mathscr{T}'$; then (2) the $\beta_{\mathtt{n}}$-rule will be fired, resulting in the term $p\{x \setminus u\}$; and then (3) reduction continues with $p\{x \setminus u\}$, thus going from input state type $\mathscr{T}'$ to the output state type in the configuration type κ. Remark that the arrow type $\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)$ appearing in the monadic type of t represents the fact that, after $\lambda x.p$ interacts with term u of type $\mathscr{M}$, $p\{x \setminus u\}$ continues the computation with behavior $\mathscr{T}' \Rightarrow \kappa$.
- **Rule (get).** This rule is used to type the get operation. Given some state s, $\mathtt{get}_l(\lambda x.t)$ binds a copy of $s(l)$ to x and returns $t\{x \setminus s(l)\}$. This is reflected by this typing rule: it assigns input state type $\{l : \Gamma(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}$ to $\mathtt{get}_l(\lambda x.t)$ under typing context $\Gamma \setminus\!\setminus x$, so that $l \in \mathtt{labels}(s)$ (by Lemma 42), and assigns input state s to t under typing context Γ, to indicate that evaluating t consumes a copy of $s(l)$. It is also important to note that the premise of the rule assigns input state $\{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}$ to t because t has access to the same input state as $\mathtt{get}_l(\lambda x.t)$. Of course, t is not required to use the contents of location l, in which case $\mathscr{M} = [\,]$.
- **Rule (set).** This rule is used to type the set operation, which is – in a sense – dual to the get operation. Given a state s and terms t and u, $\mathtt{set}_l(t, u)$ (possibly) extends s to location l by assigning t to $s(l)$ and returns u, which now has access to t in location l. This is reflected by this typing rule: it assigns input state type $\{l : \mathscr{L}\}; \mathscr{T}$ to $\mathtt{set}_l(t, u)$, which is the type of the state before it is changed by set, while assuming the input state type $\{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}$ for u, where $\mathscr{M}$ is the type of t.
- **Rule (emp).** This rule is used to type the empty state. By definition, the empty state has an empty set type $\emptyset$ because no location is set.
- **Rule (upd).** This rule is used to type the update operation and is very similar to rule (set). However, instead of describing the behavior of a term after a set operation, it describes the behavior of a state s after it is updated to $\mathtt{upd}_l(t, s)$, for some location l and term t. Of course, if location l was already defined in s, this information is already recorded in the type $\{l : \mathscr{L}\}; \mathscr{T}$ of s. Moreover, the new type of $\mathtt{upd}_l(t, s)$ keeps track of this old information by adding the multi-type $\mathscr{M}$ of the term t to the head of list multi-type $\mathscr{L}$.
- **Rule (conf).** This rule is used to type configurations. In order to type a configuration of the form (t, s), both t and s need to be typed. Note that s is the input state of t; therefore, the input state type of t needs to match the type of s. The type of the configuration is a configuration type, and it reflects the behavior of the normal configuration reached from (t, s).

Counters. Our counters are now of the form (b, m, d). This will allow us to not only keep track of the number of $\beta_{\mathtt{n}}$-steps and the size of normal configurations, just like for Weak Open CBN, but also the number of g/s-steps. We are going to use d to denote the size of normal configurations instead of s to avoid confusion with notation for states. Let $\Phi \triangleright \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t, s) : \kappa$. Then b is the number of $\beta_{\mathtt{n}}$-steps needed to normalize t, d is the size of the normal configurations reached from (t, s), and m is the number of g/s steps. Note that, while the first and last counters depend on the interplay between *consuming* and *persistent* typing rules, the second counter is simply counting the number of get and set operations that occur during evaluation. This is due to the fact that blocked configurations are not typable (by Proposition 4), and thus, get and set never get stuck. However, this is not the same thing as the total number of get and set operations in a configuration. As an example, consider configuration $(\lambda x.\mathtt{get}_l(\lambda y.t), s)$. This configuration has (at least) one get operation, but it is already a normal configuration. Therefore, no operation occurs during evaluation, and this should be reflected by the counter.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

CBN tightness. We generalize the notion of tightness for types, typing environments, type judgments, and typing derivations as follows:

- A type τ is **tight** if τ is a tight constant (*i.e.*, $\tau \in \mathtt{tt_n}$).
- A multi-type is **tight** if, for all $\mu \in \mathscr{M}$, μ is tight.
- A list multi-type $\mathscr{L}$ is **tight** if $\mathscr{L} = \star$, or $\mathscr{L} = \mathscr{M} \cdot \mathscr{L}'$, $\mathscr{M} = [\,]$, and $\mathscr{L}'$ is tight.
- A monadic type $\mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$ is **tight** if $\tau \in \mathtt{tt_n}$.
- A state type $\mathscr{T}$ is **tight** if, for all $l \in \mathtt{dom}(\mathscr{T})$, $\mathscr{T}(l)$ is tight. This tightness restriction on state types ensures that any information provided by the input state that is not used during evaluation is discarded by the (tight) type system.
- A configuration type $(\tau \times \mathscr{T})$ is **tight** if $\tau \in \mathtt{tt_n}$ and $\mathscr{T}$ is tight.
- A typing environment Γ is **tight** if, for all $x \in \mathtt{dom}(\Gamma)$, all $\mu \in \Gamma(x)$ are of the form $\mathscr{T} \Rightarrow (\tau \times \mathscr{T})$, and $\tau \in \mathtt{tt_n}$. This tightness restriction on typing environments has one crucial consequence: persistent variables cannot change the global state and thus are assigned types that reflect that fact, that is, types that have the same input and output state type (see Lemma 6).
- A term typing judgment $\Gamma \Vdash^{(b,m,d)} t : \mathscr{G}$ is **tight** if Γ is tight and $\mathscr{G}$ is tight.
- A state typing judgment $\Delta \Vdash^{(b,m,d)} s : \mathscr{T}$ is **tight** if Δ and $\mathscr{T}$ are tight.
- A configuration typing judgment $\Gamma \Vdash^{(b,m,d)} (t, s) : \kappa$ is **tight** if Γ and κ are tight.
- A typing derivation Φ is **tight** if it concludes with a tight judgment.

Example 3. *Consider configuration c_0 from Example 2. Let $\mu_0 = \{l : [\,]\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})$, and let $\mu_1 = \{l : \star\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})$, and recall that $\emptyset = \{l : \star\}$. Let Φ_0 be the following typing derivation:*

$$\dfrac{\dfrac{\dfrac{\dfrac{}{z : [\mu_0] \Vdash^{(0,0,0)} z : \mu_0}\,(\mathtt{ax})}{z : [\mu_0] \Vdash^{(0,0,0)} z : [\mu_0]}\,(\mathtt{m}) \qquad \dfrac{}{x : [\{l : [\mu_0]\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})] \Vdash^{(0,0,0)} x : \{l : [\mu_0]\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})}\,(\mathtt{ax})}{z : [\mu_0], x : [\{l : [\mu_0]\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})] \Vdash^{(0,1,0)} \mathtt{set}_l(z, x) : \{l : \star\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})}\,(\mathtt{set})}{z : [\mu_0] \Vdash^{(0,1,0)} \lambda x.\mathtt{set}_l(z, x) : \{l : \star\} \Rightarrow (([\{l : [\mu_0]\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})] \to \mu_1) \times \{l : \star\})}\,(\lambda)$$

and Φ_1 be the following typing derivation:

$$\dfrac{\dfrac{\dfrac{}{y : [\mu_0] \Vdash^{(0,0,0)} y : \mu_0}\,(\mathtt{ax})}{\Vdash^{(0,1,0)} \mathtt{get}_l(\lambda y.y) : \{l : [\mu_0]\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})}\,(\mathtt{get})}{\Vdash^{(0,1,0)} \mathtt{get}_l(\lambda y.y) : [\{l : [\mu_0]\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})]}\,(\mathtt{m})$$

Then, we can build the following tight derivation Φ_{c_0} for c_0:

$$\dfrac{\dfrac{\Phi_0 \qquad \Phi_1}{z : [\mu_0] \Vdash^{(1,2,0)} (\lambda x.\mathtt{set}_l(z, x))(\mathtt{get}_l(\lambda y.y)) : \{l : \star\} \Rightarrow (\mathtt{n} \times \{l : [\,]\})}\,(@) \qquad \dfrac{}{\emptyset \Vdash^{(0,0,0)} \varepsilon : \{l : \star\}}\,(\mathtt{emp})}{z : [\mu_0] \Vdash^{(1,2,0)} ((\lambda x.\mathtt{set}_l(z, x))(\mathtt{get}_l(\lambda y.y)), \varepsilon) : \mathtt{n} \times \{l : [\,]\}}\,(\mathtt{conf})$$

Soundness and completeness. We now extend the soundness and completeness results for Weak Open CBN to Weak Open CBN with Global State. The type system for Weak Open CBN with Global State is *quantitatively sound* and *complete* with respect to the extended reduction relation

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$\rightsquigarrow_{\mathtt{n}}$ to configurations. **Quantitative soundness** means that not only a *tightly* typable configuration (t, s) is terminating (*qualitative*) but also that exact and split measures (*quantitative*) concerning the evaluation of (t, s) to a normal configuration can be extracted from its *tight* type derivation. More precisely, if $\Phi \triangleright \Gamma \Vdash^{(b,m,d)}_{\mathtt{n}} (t, s) : \kappa$ is a tight derivation, then there exists a normal configuration (u, q) such that $(t, s) \rightsquigarrow^{(b,m)}_{\mathtt{n}} (u, q)$ with $|(u, q)|_{\mathtt{n}} = d$. Dually for **quantitative completeness**. The key properties necessary to show quantitative soundness and completeness for Weak Open CBN are extended to Weak Open CBN with Global State, and additional properties strictly related to the presence of the global state are necessary. All proofs are in Appendix A.2.

Correctness of the counters. To show quantitative completeness, we need to show that if a typable configuration (t, s) is normal, then the counters of its tight derivation are as follows: the first counter, which corresponds to the number of $\beta_{\mathtt{n}}$-steps needed to reach a Weak Open CBN with Global State normal configuration, is zero; the second counter, which corresponds to the number of g/s-steps, is zero; and the last counter is equal to the Weak Open CBN with Global State size of (t, s).

Given some term t in normal form, the type system will always infer a tight monadic type for t with the same input and output state types. This is due to the fact that normal forms do not interact with the state (at least in an effectful way) and is formalized as follows.

Lemma 6 (Normal forms have pure types). *Let* $\Phi \triangleright \Vdash^{(b,m,d)}_{\mathtt{n}} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, *such that* Γ *is tight. If* $t \in \mathtt{no}_{\mathtt{n}}$, *then* $\mathscr{T} = \mathscr{T}'$.

A stronger crucial property can (and will) be shown for the CBV setting (Lemma 15). For the CBN setting, this weaker property is enough.

The following property shows that terms in normal form and tight states have correct size.

Lemma 7 (Normal forms and tight states have correct size).

(1) *Let* $\Phi \triangleright \Gamma \Vdash^{(0,0,d)}_{\mathtt{n}} t : \mu$ *be a tight derivation. Then,* (*i*) $t \in \mathtt{no}_{\mathtt{n}}$ *and* (*ii*) $d = |t|_{\mathtt{n}}$

(2) *Let* $\Pi \triangleright \Delta \Vdash^{(b,m,d)}_{\mathtt{n}} : s : \mathscr{T}$ *be a tight derivation. Then* $b = m = d = 0$.

Lemma 8 (Normal forms take zero steps). *Let* $\Phi \triangleright \Gamma \Vdash^{(b,m,d)}_{\mathtt{n}} t : \mu$ *be a tight derivation. Then,* $b = m = 0$ *iff* $t \in \mathtt{no}_{\mathtt{n}}$.

The fact that unblocked normal configurations also have correct size follows from Lemmas 6 and 7.

Lemma 9 (Normal configurations have correct size). *Let* $\Phi \triangleright \Gamma \Vdash^{(b,m,d)}_{\mathtt{n}} (t, s) : \kappa$ *be a tight derivation of the following form:*

$$\frac{\Phi_t \triangleright \Gamma_t \Vdash^{(b_t,m_t,d_t)}_{\mathtt{n}} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}') \quad \Pi_s \triangleright \Delta_s \Vdash^{(b_t,m_t,d_t)}_{\mathtt{n}} s : \mathscr{T}}{\Gamma_t + \Delta_s \Vdash^{(b_t+b_s,m_t+m_s,d_t+d_s)} (t, s) : \tau \times \mathscr{T}'}(\mathtt{conf})$$

where $\Gamma = \Gamma_t + \Delta_s$, $\kappa = (\tau \times \mathscr{T}')$, $b = b_t + b_s$, $m = m_t + m_s$, *and* $d = d_t + d_s$. *If* $b_t = m_t = 0$, *then* (*i*) $d = |(t, s)|_{\mathtt{n}}$ *and* $b = m = 0$.

The fact that this is also the case for normal configurations follows from Lemma 8.

Lemma 10 (Normal configurations take zero steps). *Let* $\Phi \triangleright \Gamma \Vdash^{(b,m,d)}_{\mathtt{n}} (t, s) : \kappa$ *be a tight derivation. Then,* $b = m = 0$ *iff* (t, s) *is an unblocked normal configuration.*

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Another important property that needs to be shown is that the type system only types unblocked configurations.

Proposition 4 (Typed configurations are unblocked). *If* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t,s) : \kappa$, *then* (t,s) *is unblocked.*

This property allows us to conclude that a typable configuration (t,s) that takes zero steps is such that $t \in \mathtt{no}_{\mathtt{n}}$.

The following property is crucial in order to show that any state and normal form can be typed in a tight way.

Lemma 11 (Tight typability of states and normal forms).

(1) *Let* s *be a state. Then, there exists a tight derivation* $\Pi_s \rhd \emptyset \Vdash_{\mathtt{n}}^{(0,0,0)} s : \mathscr{T}$.

(2) *Let* $t \in \mathtt{no}_{\mathtt{n}}$. *Then, for any* $\mathscr{T}$, *there exists a tight derivation* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(0,0,d)} t : \mathscr{T} \Rightarrow (\mathtt{tt}_{\mathtt{n}} \times \mathscr{T})$, *such that* $d = |t|_{\mathtt{n}}$.

And the fact that any unblocked normal configuration (t,s) can be typed in a tight way follows easily from Lemma 11.

Lemma 12 (All unblocked normal configurations are tightly typable). *If* (t,s) *is a normal but unblocked configuration, then there exists a tight typing derivation* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(0,0,d)} (t,s) : \kappa$, *such that* $d = |(t,s)|_{\mathtt{n}}$.

We now show a quantitative form of type preservation along evaluation for configurations.

Lemma 13 (Quantitative subject reduction). *Let* $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, *where* $\mathtt{r} \in \{\beta_{\mathtt{n}}, \mathtt{g}, \mathtt{s}\}$. *If* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t,s) : \kappa$ *is a tight derivation, then* $\Phi' \rhd \Gamma \Vdash_{\mathtt{n}}^{(b',m',d)} (u,q) : \kappa$, *where* $\mathtt{r} = \beta_{\mathtt{n}}$ *implies* $b' = b-1$ *and* $m' = m$, *while* $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ *implies* $b' = b$ *and* $m' = m-1$.

Quantitative soundness follows from the correctness of the counters for configurations that are normal but unblocked (Lemma 7 and Lemma 10) and quantitative subject reduction (Lemma 13)

Theorem 3 (Quantitative soundness). *If* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t,s) : \kappa$ *a tight derivation, then there exists* (u,q) *such that* $u \in \mathtt{no}_{\mathtt{n}}$ *and* $(t,s) \rightsquigarrow_{\mathtt{n}}^{(b,m)} (u,q)$ *with* b $\beta_{\mathtt{n}}$*-steps,* m $\mathtt{g}/\mathtt{s}$*-steps, and* $|(u,q)|_{\mathtt{n}} = d$.

Proof. The proof is by induction on $b+m$, see Appendix A.2 for details. □

The following property guarantees that typability is also preserved back along evaluation.

Lemma 14 (Quantitative subject expansion). *Let* $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, *where* $\mathtt{r} \in \{\beta_{\mathtt{n}}, \mathtt{g}, \mathtt{s}\}$. *If* $\Phi' \rhd \Gamma \Vdash_{\mathtt{n}}^{(b',m',d')} (u,q) : \kappa$ *is a tight derivation, then* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d')} (t,s) : \kappa$, *where* $\mathtt{r} = \beta_{\mathtt{n}}$ *implies* $b' = b-1$ *and* $m' = m$, *while* $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ *implies* $b' = b$ *and* $m' = m-1$.

Quantitative completeness follows from the typability of all configurations normal but unblocked in a tight way (Lemma 12) and quantitative subject expansion (Lemma 14).

Theorem 4 (Quantitative completeness). *If* $(t,s) \rightsquigarrow_{\mathtt{n}}^{(b,m)} (u,q)$ *and* $u \in \mathtt{no}_{\mathtt{n}}$, *then there exists a tight derivation* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,|(u,q)|_{\mathtt{n}})} (t,s) : \kappa$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Example 4. *Consider again configuration c_0 from Example 2 and its tight derivation Φ_{c_0} from Example 3. The first two counters of Φ_{c_0} are different from 0: this means that c_0 is not normal but normalizes in one β_{n}-step ($b = 1$) and two $\mathtt{g}/\mathtt{s}$-steps ($m = 2$) to a normal unblocked configuration of size $d = 0 = |(z, \mathtt{upd}_l(z, \varepsilon))|_{\mathrm{n}}$.*

A discussion on the subtleties of subject expansion

In this subsection, we highlight the subtleties of the type system in order to capture – in particular – subject expansion. Consider the following reduction step:

$$c_1 = (\mathtt{set}_l(y, x), \mathtt{upd}_l(y, \varepsilon)) \rightsquigarrow (x, \mathtt{upd}_l(y, \mathtt{upd}_l(y, \varepsilon))) = c_2$$

Let us assume that c_2 is typable. In the following, we will omit the counters decorating the typing judgments, as they are irrelevant to the discussion. Moreover, let $\mathscr{T} = \{l : \mathscr{M}_1 \cdot \mathscr{M}_2\}$ and $\mathscr{M}_0 = [\mathscr{T} \Rightarrow \kappa]$, for some multi-types $\mathscr{M}_1, \mathscr{M}_2$ and configuration type κ, and recall that $\{l : \star\}; \mathscr{T} = \mathscr{T}$. The typing derivation for configuration c_2 must be of the following form:

$$\dfrac{\dfrac{}{x : \mathscr{M}_0 \Vdash x : \{l : \mathscr{M}_1 \cdot \mathscr{M}_2\} \Rightarrow \kappa}\,\mathtt{ax} \qquad \dfrac{\Phi_1 \rhd y : \mathscr{M}_1 \Vdash_{\mathrm{n}} y : \mathscr{M}_1 \qquad \dfrac{\dfrac{\Phi_2 \rhd y : \mathscr{M}_2 \Vdash_{\mathrm{n}} y : \mathscr{M}_2 \qquad \dfrac{}{\Vdash \varepsilon : \{l : \star\}}\,(\mathtt{emp})}{y : \mathscr{M}_2 \Vdash \mathtt{upd}_l(y, \varepsilon) : \{l : \mathscr{M}_2\}}\,(\mathtt{upd})}{y : \mathscr{M}_1 \sqcup \mathscr{M}_2 \Vdash \mathtt{upd}_l(y, \mathtt{upd}_l(y, \varepsilon)) : \{l : \mathscr{M}_1 \cdot \mathscr{M}_2\}}\,(\mathtt{upd})}{x : \mathscr{M}_0, y : \mathscr{M}_1 \sqcup \mathscr{M}_2 \Vdash (x, \mathtt{upd}_l(y, \mathtt{upd}_l(y, \varepsilon))) : \kappa}\,\mathtt{conf}$$

And the typing derivation for configuration c_1 can easily be recovered from the previous:

$$\dfrac{\dfrac{\Phi_1 \rhd y : \mathscr{M}_1 \Vdash_{\mathrm{n}} y : \mathscr{M}_1 \qquad \dfrac{}{x : \mathscr{M}_0 \Vdash x : \mathscr{T} \Rightarrow \kappa}\,\mathtt{ax}}{x : \mathscr{M}_0, y : \mathscr{M}_1 \Vdash \mathtt{set}_l(y, x) : \{l : \mathscr{M}_2\} \Rightarrow \kappa}\,(\mathtt{set}) \qquad \dfrac{\Phi_2 \rhd y : \mathscr{M}_2 \Vdash_{\mathrm{n}} y : \mathscr{M}_2 \qquad \dfrac{}{\Vdash \varepsilon : \{l : \star\}}\,(\mathtt{emp})}{y : \mathscr{M}_2 \Vdash \mathtt{upd}_l(y, \varepsilon) : \{l : \mathscr{M}_2\}}\,(\mathtt{upd})}{x : \mathscr{M}_0, y : \mathscr{M}_1 \sqcup \mathscr{M}_2 \Vdash (\mathtt{set}_l(y, x), \mathtt{upd}_l(y, \varepsilon)) : \kappa}\,(\mathtt{conf})$$

However, if we had assumed the equality $\mathtt{upd}_l(y, \mathtt{upd}_l(y, \varepsilon)) = \mathtt{upd}_l(y, \varepsilon)$ in our framework, then consequently $\mathscr{T} = \{l : \mathscr{M}_1 \cdot \mathscr{M}_2\} = \{l : \mathscr{M}_1\}$ and the (upd) rule should rather look as follows:

$$\dfrac{\Gamma \Vdash t : \mathscr{M} \quad \Delta \Vdash s : \{l : \mathscr{M}\}; \mathscr{T}}{\Gamma + \Delta \Vdash \mathtt{upd}_l(t, s) : \{l : \mathscr{M}\}; \mathscr{T}}\,(\mathtt{upd}^{\star})$$

But then, the typing derivation for c_2 would be of the following form instead:

$$\dfrac{\dfrac{}{x : \mathscr{M}_0 \Vdash x : \{l : \mathscr{M}_1\} \Rightarrow \kappa}\,\mathtt{ax} \qquad \dfrac{\Phi_1 \rhd y : \mathscr{M}_1 \Vdash_{\mathrm{n}} y : \mathscr{M}_1 \qquad \dfrac{}{\Vdash \varepsilon : \{l : \star\}}\,(\mathtt{emp})}{y : \mathscr{M}_1 \Vdash \mathtt{upd}_l(y, \varepsilon) : \{l : \mathscr{M}_1\}}\,(\mathtt{upd}^{\star})}{x : \mathscr{M}_0, y : \mathscr{M}_1 \Vdash (x, \mathtt{upd}_l(y, \mathtt{upd}_l(y, \varepsilon))) : \kappa}\,\mathtt{conf}$$

Now, observe that the type of x does not contain any information about $\mathscr{M}_2$. Therefore, rule (set) would infer a type of the form $\{l : \star\} \Rightarrow \kappa$ for $\mathtt{set}_l(y, x)$, and it is then no longer possible to match $\{l : \star\}$ with the type of $\mathtt{upd}_l(y, \varepsilon)$ in order to build a typing derivation for c_1. That is, subject expansion fails because $\mathtt{upd}_l(y, \mathtt{upd}_l(y, \varepsilon)) = \mathtt{upd}_l(y, \varepsilon)$, which is an equation that clearly does not hold in the current approach.

A comparison with previous approaches

We now compare our approach to de'Liguoro and Treglia (2021) and Alves et al. (2023), where updates to the same location are not tracked, and thus, there is no need to use lists of types. This

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

implies, in particular, that $\{l : \mathscr{M}\}; \mathscr{T} \neq \mathscr{T}$ holds, for all $\mathscr{M}$. de'Liguoro and Treglia (2021), there are two different rules for typing the state: one for typing a location as expected, and another one that allows one to "skip" typing a location whenever that location was not already typed before. The two rules can be presented in our formalist as follows:

$$\frac{\Gamma \Vdash t : \mathscr{M} \quad \Delta \Vdash s : \mathscr{T}}{\Gamma + \Delta \Vdash \mathtt{upd}_l(t,s) : \{l : \mathscr{M}\}; \mathscr{T}}(\mathtt{upd}_1^\star) \qquad \frac{\Gamma \Vdash s : \mathscr{T} \quad l \notin \mathtt{dom}(\mathscr{T})}{\Gamma \Vdash \mathtt{upd}_l(t,s) : \mathscr{T}}(\mathtt{upd}_2^\star)$$

It is worth noticing that this presentation is equivalent to requiring that $\{l : \mathscr{M}\}; \mathscr{T} \neq \mathscr{T}$ holds only whenever $\mathscr{M} \neq [\,]$. Consequently, rule $(\mathtt{upd}_2^\star)$ becomes redundant since rule $(\mathtt{upd}_1^\star)$ now captures both behaviors. This is precisely the approach taken in Alves et al. (2023), even if there is a small inaccuracy in the definition of operation; (it is missing the side condition requiring $\mathscr{M} \neq [\,]$).

However, even though it is possible to develop sound and complete type systems for these two alternative approaches, they lack the necessary discriminative power to distinguish configurations that have different meanings. As an example, consider the two following configurations:

$$c_3 = (y, \mathtt{upd}_l(t, \varepsilon)) \qquad c_4 = (y, \varepsilon)$$

Notice that location l is touched (or modified) in c_3, but no location is touched in c_4. Ideally, an accurate quantitative type system should be able to distinguish these two configurations, but if we consider $\{l : [\,]\}; \mathscr{T} = \mathscr{T}$ (as discussed above), both configurations can be assigned the same type. Indeed, let $\mathscr{M}_3 = [\emptyset \Rightarrow (\mathtt{tt_n} \times \emptyset)]$, for any $\mathtt{tt_n}$, and note that $\{l : [\,]\}; \emptyset = \emptyset$. The typing derivation for c_3 can be as follows (again, we omit the counters decorating the typing judgments):

$$\dfrac{\dfrac{}{y : \mathscr{M}_3 \Vdash y : \{l : [\,]\} \Rightarrow (\mathtt{tt_n} \times \{l : [\,]\})}(\mathtt{ax}) \quad \dfrac{\dfrac{\dfrac{}{\Vdash t : [\,]}(\mathtt{m}) \quad \dfrac{}{\Vdash \varepsilon : \emptyset}(\mathtt{emp})}{\Vdash \mathtt{upd}_l(t,\varepsilon) : \emptyset}(\mathtt{upd}_1^\star)}{\Vdash (y, \mathtt{upd}_l(t,\varepsilon)) : \mathtt{tt_n} \times \emptyset}(\mathtt{conf})$$

And the typing derivation for c_4 is as follows:

$$\dfrac{\dfrac{}{y : \mathscr{M}_3 \Vdash y : \emptyset \Rightarrow (\mathtt{tt_n} \times \emptyset)}(\mathtt{ax}) \quad \dfrac{}{\Vdash \varepsilon : \emptyset}(\mathtt{emp})}{\Vdash (y, \varepsilon) : \mathtt{tt_n} \times \emptyset}(\mathtt{conf})$$

Hopefully, by requiring that $\{l : \star\}; \mathscr{T} = \mathscr{T}$, but *not* $\{l : [\,]\}; \mathscr{T} = \mathscr{T}$, the type system we propose in this paper is able to clearly distinguish between the two configurations above. Let us make this purpose concrete. Let $\mathscr{M}_4 = [\{l : [\,]\} \Rightarrow (\mathtt{tt_n} \times \{l : [\,]\})]$. Indeed, the type derivation for c_3 is now as follows:

$$\dfrac{\dfrac{}{y : \mathscr{M}_4 \Vdash y : \{l : [\,]\} \Rightarrow (\mathtt{tt_n} \times \{l : [\,]\})}(\mathtt{ax}) \quad \dfrac{\dfrac{\dfrac{}{\Vdash t : [\,]}(\mathtt{m}) \quad \dfrac{}{\Vdash \varepsilon : \{l : \star\}}(\mathtt{emp})}{\Vdash \mathtt{upd}_l(t,\varepsilon) : \{l : [\,]\}}(\mathtt{upd})}{\Vdash (y, \mathtt{upd}_l(t,\varepsilon)) : \mathtt{tt_n} \times \{l : [\,]\}}(\mathtt{conf})$$

And the typing derivation for c_4 is as follows:

$$\dfrac{\dfrac{}{y : \mathscr{M}_3 \Vdash y : \emptyset \Rightarrow (\mathtt{tt_n} \times \emptyset)}(\mathtt{ax}) \quad \dfrac{}{\Vdash \varepsilon : \emptyset}(\mathtt{emp})}{\Vdash (y, \varepsilon) : \mathtt{tt_n} \times \emptyset}(\mathtt{conf})$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\begin{aligned}
\textbf{(Values)} \quad & v, w ::= x \mid \lambda x.t \\
\textbf{(Terms)} \quad & t, u, p ::= v \mid tu \mid \mathtt{get}_l(\lambda x.t) \mid \mathtt{set}_l(v, t) \\
\textbf{(States)} \quad & s, q ::= \varepsilon \mid \mathtt{upd}_l(v, s) \\
\textbf{(Configurations)} \quad & c ::= (t, s)
\end{aligned}$$

Figure 14. The extended syntax of the λ-calculus for Weak Open CBV with Global State.

This clearly illustrates the second contribution of the paper, extending the denotational semantics of our type system.

3.2 Weak Open CBV with Global State

Now that we have presented the extension of Weak Open CBN with Global State, we will present the more subtle extension of Weak Open CBV with Global State. The extension of the basic λ-calculus is similar to the one for Weak Open CBN with Global State, but we distinguish values from the other terms of the language, and the global state will now hold only values instead of arbitrary terms. Let us however briefly discuss this choice.

It would have been possible to consider the same notion of global state as in CBN (*i.e.,* holding arbitrary terms, and not only values), together with a unique rule for the `get` operator in both operational semantics have a similar shape. Indeed, the following rule (`get`*) would work for both CBN and CBV:

$$\frac{s \equiv \mathtt{upd}_l(u, q)}{(\mathtt{get}_l(\lambda x.t), s) \rightsquigarrow_{\mathtt{g}} ((\lambda x.t)u, s)}(\mathtt{get}^\star)$$

However, from a quantitative point of view, this is not equivalent to rule that is usually adopted for CBV:

$$\frac{s \equiv \mathtt{upd}_l(v, q)}{(\mathtt{get}_l(\lambda x.t), s) \rightsquigarrow_{\mathtt{g}} (t\{x \backslash v\}, s)}(\mathtt{get})$$

Clearly, both approaches are very different from a quantitative point of view. Suppose that the term u in rule (`get`*) evaluates to the value v in rule (`get`). If we consider rule (`get`*), term u will be evaluated to v after it is retrieved from the global state *every time*. If, instead, we consider rule (`get`), term u must first be evaluated to v before being placed in the global and thus will only be evaluated *once*. This puts into question the smart duplicative behavior that is usually associated with CBV strategies but might be worth exploring in future work. For now, we will stick to the usual approach and simply adopt rule (`get`) in Section 3.2.1.

3.2.1 Syntax and operational semantics

In this section, we introduce the syntax for the extension of the λ-calculus with global state and the operational semantics following CBV.

The syntax for Weak Open CBV with Global State. The sets of **values**, **terms**, **states**, and **configurations** are presented in Figure 14.

All notions regarding free and bound variables of terms, states, and configurations, as well as the notion of the domain of a state, are similar to the case of Weak Open CBN with Global State (*c.f.* Section 3.1).

Weak Open CBV size. The notion of **Weak Open CBV size** is extended to operations, states, and configurations according to Figure 15.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Weak Open CBV Size Extended for Global State

$$
\begin{aligned}
&\cdots\\
|\mathtt{get}_l(\lambda x.t)|_{\mathtt{v}} &:= |t|_{\mathtt{v}}\\
|\mathtt{set}_l(v,u)|_{\mathtt{v}} &:= |u|_{\mathtt{v}}\\
|s|_{\mathtt{v}} &:= 0\\
|(t,s)|_{\mathtt{v}} &:= |t|_{\mathtt{v}}
\end{aligned}
$$

Figure 15. Weak Open CBV size extended for global state.

(a)

$$\frac{}{((\lambda x.t)v,s) \rightsquigarrow_{\beta_{\mathtt{v}}} (t\{x\backslash v\},s)}\ (\beta_{\mathtt{v}})$$

$$\frac{(t,s) \rightsquigarrow_{\mathtt{r}} (u,q) \quad \mathtt{r} \in \{\beta_v, \mathtt{g}, \mathtt{s}\}}{(tp,s) \rightsquigarrow_{\mathtt{r}} (up,q)}\ (\mathtt{appL})$$

$$\frac{(t,s) \not\rightsquigarrow_{\mathtt{r}} \quad (t,s) \text{ unblocked} \quad (u,s) \rightsquigarrow_{\mathtt{r}} (u',q) \quad \mathtt{r} \in \{\beta_v, \mathtt{g}, \mathtt{s}\}}{(tu,s) \rightsquigarrow_{\mathtt{r}} (tu',q)}\ (\mathtt{appR})$$

$$\frac{s \equiv \mathtt{upd}_l(v,q)}{(\mathtt{get}_l(\lambda x.t),s) \rightsquigarrow_{\mathtt{g}} (t\{x\backslash v\},s)}\ (\mathtt{get}) \qquad \frac{}{(\mathtt{set}_l(v,t),s) \rightsquigarrow_{\mathtt{s}} (t,\mathtt{upd}_l(v,s))}\ (\mathtt{set})$$

Reduction Rules for Weak Open CBV with Global State.

(b)

A configuration (t,s) is a **blocked configuration** if: $t = \mathtt{get}_l(\lambda x.u)$ and $l \notin \mathtt{labels}(s)$; $t = up$ and (u,s) is blocked; or $t = up$, $(u,s) \not\rightsquigarrow$, (u,s) is not blocked, and (p,s) is blocked. We say that (t,s) is an **unblocked configuration** otherwise.

Blocked Configurations for Weak Open CBV with Global State.

Figure 16. Reduction rules and blocked configurations for Weak Open CBV with Global State.

Evaluation and blocked configurations in Weak Open CBV with Global State. Weak Open CBV evaluation is extended to global state in the same way as we did for Weak Open CBN with Global State. The extension is denoted by $\rightsquigarrow_{\mathtt{v}}$, and due to the particular nature of CBV, this notion is *mutually* defined with that of blocked configurations in Figure 16. Indeed, evaluation for Weak Open CBV with Global State is defined as the union of the reduction rule in Figure 16a, where the notion of **blocked configurations** is in Figure 16b.

In particular, note that reduction rule (appR) only evaluates terms on the right of applications if the term on the left is not blocked. Rule $(\beta_{\mathtt{v}})$ and (appL) are the same as for Weak Open CBV, and rules (get) and (set) are the same as for Weak Open CBN with Global State, with the only difference being the fact that the global state only holds values.

Example 5. *All configurations in Example 1 are also blocked configurations in this setting, and so are the following ones, when assuming* $l \neq l'$:

$$((\lambda y.y)(\mathtt{get}_l(\lambda x.x)), \mathtt{upd}_{l'}(t,\varepsilon)) \quad (((\lambda y.y)(\mathtt{get}_l(\lambda x.x)))(\lambda z.z), \mathtt{upd}_{l'}(t,\varepsilon))$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\textbf{(Normal)} \qquad \mathtt{no_v} ::= x \mid \lambda x.t \mid \mathtt{ne_v}$$
$$\textbf{(Neutral)} \qquad \mathtt{ne_v} ::= x\,\mathtt{no_v} \mid \mathtt{no_v}\,\mathtt{ne_v} \mid \mathtt{ne_v}\,\mathtt{no_v}$$

Figure 17. Normal and neutral terms for Weak Open CBV with Global State.

Normal configurations. The notion of *normal configuration* is also the same as for Weak Open CBN with Global State: a configuration (t, s) is said to be **normal** if it is a **blocked configuration** or $t \in \mathtt{no_v}$, where $\mathtt{no_v}$ is defined in Figure 17.

Example 6. *Consider configuration* $c_1 = ((\lambda x.\mathtt{get}_l(\lambda y.yx))(\mathtt{set}_l((\lambda w.w), z)), \varepsilon)$. *Then we can reach a normal configuration as follows:*

$$\begin{aligned}((\lambda x.\mathtt{get}_l(\lambda y.yx))(\mathtt{set}_l((\lambda w.w), z)), \varepsilon) &\rightsquigarrow_{\mathtt{s}} ((\lambda x.\mathtt{get}_l(\lambda y.yx))z, \mathtt{upd}_l((\lambda w.w), \varepsilon))\\ &\rightsquigarrow_{\beta_{\mathtt{v}}} (\mathtt{get}_l(\lambda y.yz), \mathtt{upd}_l((\lambda w.w), \varepsilon))\\ &\rightsquigarrow_{\mathtt{g}} ((\lambda w.w)z, \mathtt{upd}_l((\lambda w.w), \varepsilon))\\ &\rightsquigarrow_{\beta_{\mathtt{v}}} (z, \mathtt{upd}_l((\lambda w.w), \varepsilon))\end{aligned}$$

The following proposition, whose proof is in Appendix A.3, shows that normal configurations are indeed irreducible.

Proposition 5. *Let* (t, s) *be a configuration. Then* (t, s) *is normal iff* $(t, s) \not\rightsquigarrow_{\mathtt{v}}$.

3.2.2 A quantitative type system for Weak Open CBV with Global State

We now present the quantitative type system for Weak Open CBV with Global State. Following the same presentation as for the Weak Open CBN with Global State, we start by introducing the CBV monadic translation of the λ-calculus of Wadler (1990) by adapting it to account for tight constants and multi-types, and then instantiating the monadic translation with the monad for global state.

The CBV monadic translation. In order to lift the type system for Weak Open CBV into the global state monad, we are going to follow the CBV translations of the simply-typed λ-calculus in Wadler (1990). Again, we will have to slightly adapt the translation in order to accommodate for tight constants and multi-types. The resulting translation is presented in Figure 18.

State and monadic (multi-)types. The grammar for types is presented in Figure 19.

State Types for Weak Open CBV with Global State, and all related notions, are similar to those for Weak Open CBN with Global State.

Multi-types are multi-sets of *value* types. **Monadic types** and **configuration types** are exactly the same as for Weak Open CBN with Global State. Moreover, since some typing rules will apply both to value types and multi-types, we will use π as a new auxiliary symbol ranging over both these kinds of types. All other notions related to types are common to Weak Open CBN with Global State. Note that the set of tight constants now consists of *three* different symbols, as in Weak Open CBV.

Typing system for Weak Open CBV with Global State. The type system for Weak Open CBV with Global State is defined by the typing rules in Figure 20.

The notions of **typing environments**, **typing judgments**, **typing rules**, and **typability** are the same as those for Weak Open CBN with Global State.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Let S stand for the type of global states and $T(X) = S \Rightarrow (X \times S)$ stand for the global state monad.

(Types)

$$\mathtt{tt_v}^* := \mathtt{tt_v}$$
$$(\mathscr{M} \to \tau)^* := (\mathscr{M}^* \to T(\tau^*))$$
$$(\mathscr{M})^* := [\tau_i^*]_{i \in I}$$

(Typing Environments)

$$(x_1 : \mathscr{M}_1, \ldots, x_n : \mathscr{M}_n)^* := x_1 : \mathscr{M}_1^*, \ldots, x_n : \mathscr{M}_n^*$$

(Typing Judgements for Terms)

$$(\Gamma \vdash t : \tau)^* := \Gamma^* \vdash t : T(\tau^*)$$

Figure 18. The CBV monadic translation for types, typing environments, and judgments.

(Tight Constants)	$\mathtt{tt_v} ::= \mathtt{v} \mid \mathtt{a} \mid \mathtt{n}$
(Multi-types)	$\mathscr{M} ::= [\sigma_i]_{i \in I}$ where I is a finite set
(Types)	$\tau ::= \mathtt{n} \mid \sigma \mid \mathscr{M}$
(Value Types)	$\sigma ::= \mathtt{v} \mid \mathtt{a} \mid \mathscr{M} \to \mu$
(Monadic Types)	$\mu ::= \mathscr{T} \Rightarrow \kappa$
(Configuration Types)	$\kappa ::= \tau \times \mathscr{T}$
(List Multi-types)	$\mathscr{L} ::= \star \mid \mathscr{M} \cdot \mathscr{L}$
(State Types)	$\mathscr{T} ::= \{l_i : \mathscr{L}_i\}_{i \in I}$ where all l_i are distinct

Figure 19. Types for Weak Open CBV with Global State.

Like in Weak Open CBN with Global State, we distinguish between *consuming* and *persistent* rules. The intuition behind the rules is the following (again, ignore the counters (b, m, d)):

- **Rules (ax)**, **(λ)**, **($@^1_p$)**, **($@^2_p$)**, and **(m).** These rules are essentially the same as for Weak Open CBV, but with types replaced with monadic types.
- **Rule** (λ_p) is the same as for Weak Open CBV.
- **Rule ($@$).** In this rule, we see how CBV, which is more intricate than the CBN, influences the type system. Note that, just like in the case of Weak Open CBN with Global State, this rule is used to type *consuming* applications. Therefore, CBV tells us the following: (1) t is reduced until reaching an abstraction $\lambda x.p$, going from input state type $\mathscr{T}$ to an output state type $\mathscr{T}''$; then (2) u is reduced until reaching a value v, going from the input state type $\mathscr{T}''$ to an output state type $\mathscr{T}'$; then (3) the β_v-rule will be fired, resulting in the term $p\{x \setminus v\}$; and then (4) reduction continues with $p\{x \setminus v\}$, thus going from input state type $\mathscr{T}'$ to the output state type in the configuration type κ. Remark that the arrow type $\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)$ appearing in the monadic type of t represents the fact that, after $\lambda x.p$ interacts with value v of type $\mathscr{M}$, $p\{x \setminus v\}$ continues the computation with behavior $\mathscr{T}' \Rightarrow \kappa$.
- **Rule (↑).** In CBV, variables and abstractions are values. Rule (↑) lifts value types and multi-types to pure types, that is, monadic types with the same input and output state types.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Rules for Terms

$$\frac{}{x : [\sigma] \Vdash^{(0,0,0)} x : \sigma}\ (\mathtt{ax}) \qquad \frac{\Gamma \Vdash^{(b,m,d)} v : \pi}{\Gamma \Vdash^{(b,m,d)} v : \mathscr{T} \Rightarrow (\pi \times \mathscr{T})}\ (\uparrow)$$

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \mu}{\Gamma \setminus\!\setminus x \Vdash^{(b,m,d)} \lambda x.t : \Gamma(x) \to \mu}\ (\lambda) \qquad \frac{(\Gamma_i \Vdash^{(b_i,m_i,d_i)} v : \sigma_i)_{i\in I}}{+_{i\in I}\Gamma_i \Vdash^{(+_{i\in I}b_i,+_{i\in I}m_i,+_{i\in I}d_i)} v : [\sigma_i]_{i\in I}}\ (\mathtt{m})$$

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'') \qquad \Delta \Vdash^{(b',m',d')} u : \mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}')}{\Gamma + \Delta \Vdash^{(1+b+b',m+m',d+d')} tu : \mathscr{T} \Rightarrow \kappa}\ (@)$$

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}{\Gamma \setminus\!\setminus x \Vdash^{(b,1+m,d)} \mathtt{get}_l(\lambda x.t) : \{l : \Gamma(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}\ (\mathtt{get})$$

$$\frac{\Gamma \Vdash^{(b,m,d)} v : \mathscr{M} \qquad \Delta \Vdash^{(b',m',d')} t : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}{\Gamma + \Delta \Vdash^{(b+b',1+m+m',d+d')} \mathtt{set}_l(v,t) : \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}\ (\mathtt{set})$$

Rules for Persistent Terms

$$\frac{}{\emptyset \Vdash^{(0,0,0)} \lambda x.t : \mathtt{a}}\ (\lambda_{\mathtt{p}})$$

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}'') \qquad \Delta \Vdash^{(b',m',d')} u : \mathscr{T}'' \Rightarrow (\mathtt{tt_v} \times \mathscr{T}')}{\Gamma + \Delta \Vdash^{(b+b',m+m',1+d+d')} tu : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}\ (@^1_{\mathtt{p}})$$

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{T} \Rightarrow (\mathtt{tt_v} \times \mathscr{T}'') \qquad \Delta \Vdash^{(b',m',d')} u : \mathscr{T}'' \Rightarrow (\mathtt{n} \times \mathscr{T}')}{\Gamma + \Delta \Vdash^{(b+b',m+m',1+d+d')} tu : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}\ (@^2_{\mathtt{p}})$$

Rules for States

$$\frac{}{\emptyset \Vdash^{(0,0,0)} \varepsilon : \emptyset}\ (\mathtt{emp}) \qquad \frac{\Gamma \Vdash^{(b,m,d)} v : \mathscr{M} \qquad \Delta \Vdash^{(b',m',d')} s : \{l : \mathscr{L}\}; \mathscr{T}}{\Gamma + \Delta \Vdash^{(b+b',m+m',d+d')} \mathtt{upd}_l(v,s) : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}}\ (\mathtt{upd})$$

Rule for Configurations

$$\frac{\Gamma \Vdash^{(b,m,d)} t : \mathscr{T} \Rightarrow \kappa \qquad \Delta \Vdash^{(b',m',d')} s : \mathscr{T}}{\Gamma + \Delta \Vdash^{(b+b',m+m',d+d')} (t,s) : \kappa}\ (\mathtt{conf})$$

Figure 20. Typing rules for Weak Open CBV with Global State.

- **Rules (get), (set), (emp), (upd), and (conf).** These rules are essentially the same as for Weak Open CBN with Global State, with the only difference being that rules (set) and (upd) place values into locations instead of arbitrary terms.

Counters. The counters work as they did for Weak Open CBN with Global State.

CBV tightness. The notions of tightness for types, typing environments, type judgments, and typing derivations for CBV are as follows:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- A type τ is **tight** if τ is a tight constant (*i.e.*, $\tau \in \mathtt{tt}_{\mathtt{v}}$).
- The notions of tightness for multi-types, list multi-types, and monadic, state, and configuration types are defined just like for Weak Open CBN with Global State (Section 3.1.2).
- A typing environment Γ is **tight** if, for all $x \in \mathtt{dom}(\Gamma)$, $\Gamma(x)$ is tight.
- The notions of tightness for **typing judgments** and **typing derivations** are defined in the same way as for Weak Open CBN with Global State (Section 3.1.2).

Example 7. *Consider configuration c_1 from Example 6. Let $\mathscr{M}_0 = [[\mathtt{v}] \to \{l : [\,]\} \Rightarrow (\mathtt{v} \times \{l : [\,]\})]$, $\mathscr{T}_0 = \{l : [\,]\}$, and Φ_0 be the following typing derivation:*

$$\dfrac{\dfrac{\dfrac{\dfrac{}{y : \mathscr{M}_0 \Vdash^{(0,0,0)} y : [\mathtt{v}] \to \mathscr{T}_0 \Rightarrow (\mathtt{v} \times \mathscr{T}_0)}\,(\mathtt{ax})}{y : \mathscr{M}_0 \Vdash^{(0,0,0)} y : \mathscr{T}_0 \Rightarrow ([\mathtt{v}] \to \mathscr{T}_0 \Rightarrow (\mathtt{v} \times \mathscr{T}_0) \times \mathscr{T}_0)}\,(\uparrow) \qquad \dfrac{\dfrac{\dfrac{}{x : [\mathtt{v}] \Vdash^{(0,0,0)} x : \mathtt{v}}\,(\mathtt{ax})}{x : [\mathtt{v}] \Vdash^{(0,0,0)} x : [\mathtt{v}]}\,(\mathtt{m})}{x : [\mathtt{v}] \Vdash^{(0,0,0)} x : \mathscr{T}_0 \Rightarrow ([\mathtt{v}] \times \mathscr{T}_0)}\,(\uparrow)}{y : \mathscr{M}_0, x : [\mathtt{v}] \Vdash^{(1,0,0)} yx : \mathscr{T}_0 \Rightarrow (\mathtt{v} \times \mathscr{T}_0)}\,(@)}{x : [\mathtt{v}] \Vdash^{(1,1,0)} \mathtt{get}_l(\lambda y.yx) : \{l : \mathscr{M}_0\} \Rightarrow (\mathtt{v} \times \mathscr{T}_0)}\,(\mathtt{get})}{\Vdash^{(1,1,0)} \lambda x.\mathtt{get}_l(\lambda y.yx) : [\mathtt{v}] \to (\{l : \mathscr{M}_0\} \Rightarrow (\mathtt{v} \times \mathscr{T}_0))}\,(\lambda)}{\Vdash^{(1,1,0)} \lambda x.\mathtt{get}_l(\lambda y.yx) : \emptyset \Rightarrow ([\mathtt{v}] \to (\{l : \mathscr{M}_0\} \Rightarrow (\mathtt{v} \times \mathscr{T}_0)) \times \emptyset)}\,(\uparrow)$$

Let Φ_1 be the following typing derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{w : [\mathtt{v}] \Vdash^{(0,0,0)} w : \mathtt{v}}\,(\mathtt{ax})}{w : [\mathtt{v}] \Vdash^{(0,0,0)} w : \mathscr{T}_0 \Rightarrow (\mathtt{v} \times \mathscr{T}_0)}\,(\uparrow)}{\Vdash^{(0,0,0)} (\lambda w.w) : [\mathtt{v}] \to \mathscr{T}_0 \Rightarrow (\mathtt{v} \times \mathscr{T}_0)}\,(\lambda)}{\Vdash^{(0,0,0)} (\lambda w.w) : \mathscr{M}_0}\,(\mathtt{m}) \qquad \dfrac{\dfrac{\dfrac{}{z : [\mathtt{v}] \Vdash^{(0,0,0)} z : \mathtt{v}}\,(\mathtt{ax})}{z : [\mathtt{v}] \Vdash^{(0,0,0)} z : [\mathtt{v}]}\,(\mathtt{m})}{z : [\mathtt{v}] \Vdash^{(0,0,0)} z : \{l : \mathscr{M}_0\} \Rightarrow ([\mathtt{v}] \times \{l : \mathscr{M}_0\})}\,(\uparrow)}{z : [\mathtt{v}] \Vdash^{(0,1,0)} \mathtt{set}_l((\lambda w.w), z) : \emptyset \Rightarrow ([\mathtt{v}] \times \{l : \mathscr{M}_0\})}\,(\mathtt{set})$$

Then, we can build the following tight derivation Φ_{c_1} for c_1:

$$\dfrac{\dfrac{\Phi_0 \qquad \Phi_1}{z : [\mathtt{v}] \Vdash^{(2,2,0)} (\lambda x.\mathtt{get}_l(\lambda y.yx))(\mathtt{set}_l((\lambda w.w), z)) : \emptyset \Rightarrow (\mathtt{v} \times \mathscr{T}_0)}\,(@) \qquad \dfrac{}{\Vdash^{(0,0,0)} \varepsilon : \emptyset}\,(\mathtt{emp})}{z : [\mathtt{v}] \Vdash^{(2,2,0)} ((\lambda x.\mathtt{get}_l(\lambda y.yx))(\mathtt{set}_l((\lambda w.w), z)), \varepsilon) : \mathtt{v} \times \mathscr{T}_0}\,(\mathtt{conf})$$

Soundness and completeness. We now extend the soundness and completeness result for Weak Open CBN to Weak Open CBV with Global State. The key properties necessary to show quantitative soundness and completeness for Weak Open CBV with Global State are the same as for Weak Open CBN with Global State. However, a stronger pureness over normal forms is necessary in order to show quantitative completeness, due to the fact that reduction happens both on the left and on the right of applications. All proofs are in Appendix A.3.

Correctness of the counters. To show quantitative completeness, we need to show that if a configuration (t, s) is normal, then the counters of its tight derivation are as follows: the first counter,

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

which corresponds to the number of $\beta_{\mathtt{v}}$-steps needed to reach a Weak Open CBV with Global State normal configuration, is zero; the second counter, which corresponds to the number of g/s-steps, is zero; and the last counter is equal to the Weak Open CBN with Global State size of (t, s).

Given some term $t \in \mathtt{no}_{\mathtt{v}}$, in addition to the type system always inferring a tight monadic type for t with the same input and output state types, such a type derivation exists for any state type.

Lemma 15 (Normal forms have pure types). *Let* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, *such that* Γ *is tight. If* $t \in \mathtt{no}_{\mathtt{v}}$, *then* (*1*) $\mathscr{T} = \mathscr{T}'$ *and,* (*2*) *for any* $\mathscr{T}_0$, *there exists* $\Phi' \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mathscr{T}_0 \Rightarrow (\tau \times \mathscr{T}_0)$.

The previous property is a stronger version of Lemma 6. It is a crucial property in order to show the quantitative versions of the usual subject reduction (Lemma 19) and subject expansion (Lemma 20) properties for CBV. We now provide an intuitive explanation of why this is the case, and point the reader to the full proofs in Appendix A.3 for a better understanding. Consider the following configuration (tu, s), such that $(t, s) \not\rightsquigarrow_{\mathtt{v}}$ and $(u, s) \rightsquigarrow_{\mathtt{v}} (u', q)$. Moreover, let us assume that $s \neq q$. Then, $(tu, s) \rightsquigarrow_{\mathtt{v}} (tu', q)$, and it is easy to see that $(t, q) \not\rightsquigarrow_{\mathtt{v}}$ (by Proposition 5). Clearly, the typability of (tu, s) depends on the typability of (t, s), and the typability of (tu', q) depends on the typability of (t, q). However, the typability of t in (t, s) tells us that t takes s as its input state, and the typability of t in (t, q) tells us that t should also be able to take q as its input state. Hopefully, this example also clarifies why this property is not necessary in the CBN setting, where evaluation only occurs on the left of applications.

The following lemma tells us that unblocked normal configurations have correct size.

Lemma 16 (Tight configurations have correct size). *Let* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t, s) : \kappa$ *be a tight derivation of the following form:*

$$\frac{\Phi_t \rhd \Gamma_t \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}') \quad \Pi_s \rhd \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{\Gamma \Vdash^{(b,m,d)} (t, s) : \kappa}(\mathtt{conf})$$

where $\Gamma = \Gamma_t + \Delta_s$, $\kappa = (\tau \times \mathscr{T}')$, $b = b_t + b_s$, $m = m_t + m_s$, *and* $d = d_t + d_s$. *If* $b_t = m_t = 0$, *then* (*i*) $b = m = 0$ *and* (*ii*) $d = |(t, s)|_{\mathtt{v}}$.

Lemma 17 (Unblocked normal configurations take zero steps). *Let* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t, s) : \kappa$ *be a tight derivation. Then,* $b = m = 0$ *iff* (t, s) *is an unblocked normal configuration.*

Like in Weak Open CBN with Global State, it is also necessary to show that typed configurations that take zero steps are not blocked.

Proposition 6 (Typed configurations are unblocked). *Let* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t, s) : \kappa$. *Then* $\mathtt{tight}(\Gamma)$ *implies* (t, s) *is unblocked.*

Moreover, we also need to show that all unblocked normal configurations are tightly typable.

Lemma 18 (All configurations in normal form are tightly typable). *If* (t, s) *is an unblocked normal configuration, then there exists a tight typing derivation* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t, s) : \kappa$, *such that* $d = |(t, s)|_{\mathtt{v}}$.

We now show a quantitative form of type preservation along evaluation for configurations, which relies on Lemma 15.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 19 (Quantitative subject reduction). *Let* $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, *where* $\mathtt{r} \in \{\beta_{\mathtt{v}}, \mathtt{g}, \mathtt{s}\}$. *If* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t,s) : \kappa$ *is a tight derivation, there exists* $\Phi' \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b',m',d)} (u,q) : \kappa$, *where* $\mathtt{r} = \beta_{\mathtt{v}}$ *implies* $b' = b - 1$ *and* $m' = m$, *while* $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ *implies* $b' = b$ *and* $m' = m - 1$.

Quantitative soundness follows from the correctness of the counters for unblocked normal configurations (Lemma 17 and Lemma 16) and quantitative subject reduction (Lemma 19).

Theorem 5 (Quantitative soundness). *If* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t,s) : \kappa$ *is a tight derivation, then there exists* (u,q) *such that* $u \in \mathtt{no}_{\mathtt{v}}$ *and* $(t,s) \rightsquigarrow_{\mathtt{v}}^{(b,m)} (u,q)$ *with* b $\beta_{\mathtt{v}}$*-steps,* m $\mathtt{g}/\mathtt{s}$*-steps, and* $|(u,q)|_{\mathtt{v}} = d$.

The following property also relies on Lemma 15 and guarantees that typability is preserved back along evaluation.

Lemma 20 (Quantitative subject expansion). *Let* $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, *where* $\mathtt{r} \in \{\beta_{\mathtt{v}}, \mathtt{g}, \mathtt{s}\}$. *If* $\Phi' \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b',m',d)} (u,q) : \kappa$ *is a tight derivation, there exists* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t,s) : \kappa$, *where* $\mathtt{r} = \beta_{\mathtt{v}}$ *implies* $b' = b - 1$ *and* $m' = m$, *while* $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ *implies* $b' = b$ *and* $m' = m - 1$.

Quantitative completeness follows from the typability of all unblocked normal configurations in a tight way (Lemma 18) and quantitative subject expansion (Lemma 20).

Theorem 6 (Quantitative completeness). *If* $(t,s) \rightsquigarrow_{\mathtt{v}}^{(b,m,d)} (u,q)$ *and* $u \in \mathtt{no}_{\mathtt{v}}$, *then there exists a tight derivation* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,|(u,q)|_{\mathtt{v}})} (t,s) : \kappa$.

Example 8. *Consider again configuration* c_1 *from Example 6 and its tight derivation* Φ_{c_1} *from Example 7. The first two counters of* Φ_{c_1} *are different from* 0*: this means that* c_1 *is not normal but normalizes in two* $\beta_{\mathtt{v}}$*-steps (*$b = 2$*) and two* $\mathtt{g}/\mathtt{s}$*-steps (*$m = 2$*) to an unblocked normal configuration of size* $d = 0 = |(z, \mathtt{upd}_l((\lambda w.w), \varepsilon))|_{\mathtt{v}}$.

4. Related work

Several papers proposed quantitative approaches for different notions of CBV and CBN (without effects). There are several papers exploring exact *and* split measures for CBN, such as the pioneer work by Accattoli et al. (2018, 2020), as well as others, for example, Alves et al. (2019), which explores CBN enriched with patterns, and Kesner and Vial (2020), which generalized CBN to classical logic. The state of the art for CBV is as follows. The first non-idempotent intersection type system for Plotkin's CBV is Ehrhard (2012), where reduction is allowed under abstractions, and terms are considered to be closed. This work was further extended by Carraro and Guerrieri (2014), where commutation rules are added to the calculus. None of these contributions extracts quantitative bounds from the type derivations. A calculus for open CBV is proposed in Accattoli and Guerrieri (2018), where *fireballs* – normal forms – can be erased and duplicated. Quantitative results are obtained, but no split measures. Other similar approaches appear in Guerrieri (2018) and Accattoli et al. (2023). A logical characterization of CBV solvability is given in Accattoli and Guerrieri (2022). The resulting non-idempotent system gives quantitative information of the *solvable* associated reduction relation, and exact measures are obtained. A similar notion of solvability for both CBN and CBV for generalized applications was studied in Kesner and Peyrot (2022), together with a logical characterization provided by a quantitative system. Other non-idempotent systems for CBV were proposed (Manzonetto et al., 2019; Kerinec et al., 2021), but they do not enjoy subject reduction and expansion. Exact and split measures for (strong) open CBN and CBV

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

are developed in Kesner and Viso (2022), which relate the exact measure to those of a subsuming framework called the bang calculus, which is able to encode both strategies, developed in Bucciarelli et al. (2020, 2023). Dezani-Ciancaglini et al. (2009), a system with universally quantified intersection and reference types is introduced for a language belonging to the ML-family. However, intersections are idempotent and only (qualitative) soundness is proved.

With respect to CBN and CBV with effects, there has been a lot of work involving probabilistic versions of the lambda calculus. In Faggian and Ronchi Della Rocca (2019), extensions of the lambda calculus with a probabilistic choice operator for CBV and CBN are introduced. However, no quantitative results are provided. In Breuvart and Dal Lago (2018), monadic intersection types are used to obtain a (non-exact) quantitative type system for a probabilistic calculus identical to the one in Faggian and Ronchi Della Rocca (2019). More recently, in Dal Lago et al. (2021), non-idempotent intersection types were used to characterize almost sure termination and obtain exact split measures for a probabilistic calculus. This is done for both *weak* CBN and CBV, but on a *closed* setting, and not for other effects.

Concerning (exact) quantitative type systems for programming languages with global state, the state of the art is still underexplored. Some type systems are given in Davies and Pfenning (2000), and Benton et al. (2009), but they cannot be seen as models (since termination does not imply typability) and lack quantitative results. Our work is inspired by a recent idempotent (thus only qualitative and not quantitative) type system for CBV with global memory proposed by de'Liguoro and Treglia (2021) (a technical comparison to this work appears at the end of Section 3.1.2). This work was extended in Gavazzo et al. (2024) to a generic framework for algebraic effects. However, their approach lacks any quantitative results and does not hold for global state.

Languages with global state are often closely related to concurrent languages due to the challenges and complexities involved in managing shared mutable state in concurrent settings. Indeed, if we consider locations as channels and the `get` and `set` operations as read and write, we can find some similarities between our language and concurrent languages, such as that of Gay and Vasconcelos (2010), where buffered channels are used in order to obtain asynchronous communication, or that of Dal Lago et al. (2019), where an intersection type discipline is used to capture properties such as deadlock freedom and termination for a fragment of the π-calculus.

5. Conclusion

This paper provides a foundational step into the development of quantitative type systems for programming languages with effects. We focus on a simple language with global memory access capabilities. Due to the inherent lack of confluence in such a framework, we fix two particular evaluation strategies: Weak Open CBN and CBV. For each language, we provide a quantitative type system characterizing termination and capturing exact measures for the number of β-steps, memory accesses, and the size of unblocked normal configurations. This study provides a valuable insight into time and space analysis of languages with global memory, with respect to the length of evaluation and the size of unblocked normal configurations, respectively.

In future work, we would like to explore effectful computations involving global memory in a more general framework subsuming CBN and CBV evaluation, such as the bang calculus (Bucciarelli et al., 2020). Furthermore, we would like to apply our quantitative techniques to other (algebraic) effects, such as non-termination, exceptions, non-determinism, and I/O. Finally, we would also like to explore the connection between this work and that of Gay and Vasconcelos (2010) and Dal Lago et al. (2019), in order to understand how to extract quantitative information for asynchronous communications, which is not the focus of either of these works. Finally, we would also like to take advantage of our techniques in order to study lenses (Bohannon et al., 2006) and bidirectional programming languages (Bohannon et al., 2008).

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Acknowledgments. The authors would like to thank the reviewers for their helpful suggestions that improved this work and Andrej Bauer for providing them with good references explaining the state-logging monad. The last author also acknowledges the following funding: National Funds through the Portuguese funding agency – FCT – through the individual research grant 2021.04731.BD; Base Funding - UIDB/00027/2020 of the Artificial Intelligence and Computer Science Laboratory – LIACC – funded by national funds through the FCT/MCTES (PIDDAC); European Research Network on Formal Proofs COST Action CA20111.

Competing interests. The authors declare none.

Notes

1 For the purpose of this example, we are considering the monadic CBV translation of arrow types according to Moggi (1989, 1991). In the case of the monadic CBN translation of arrow types, we obtain $g : T(A) \to T(B)$ instead.
2 In this paper, by quantitative types we mean non-idempotent intersection types. Another meaning can be found in Atkey (2018).

References

Accattoli, B., Graham-Lengrand, S. and Kesner, D. (2018). Tight typings and split bounds. *Proceedings of the ACM on Programming Languages*. **2** (ICFP), 94:1–94:30.

Accattoli, B., Graham-Lengrand, S. and Kesner, D. (2020). Tight typings and split bounds, fully developed. *Journal of Functional Programming*. **30** (e14) 1–101.

Accattoli, B. and Guerrieri, G. (2016). Open call-by-value, Hanoi. In: Igarashi, A. (eds.) *Programming Languages and Systems – 14th Asian Symposium, APLAS 2016, Hanoi, Vietnam, November 21–23, 2016, Proceedings, Volume 10017 of Lecture Notes in Computer Science*, 206–226.

Accattoli, B. and Guerrieri, G. (2018) Types of fireballs, Wellington, *16th Asian Symposium Programming Languages and Systems, (APLAS), 2018, Wellington, New Zealand, Volume 11275 of Lecture Notes in Computer Science*, Springer, 45–66.

Accattoli, B. and Guerrieri, G. (2022). The theory of call-by-value solvability. *Proceedings of the ACM on Programming Languages*, **6** (ICFP) 855–885.

Accattoli, B., Guerrieri, G. and Leberle, M. (2023) Strong call-by-value and multi types, In: Brahám, Á., Dubslaff, E. and Tarifa, S. L. T. (eds.) *Theoretical Aspects of Computing - ICTAC 2023 - 20th International Colloquium, Lima, Peru, December 4–8, 2023, Proceedings, Volume 14446 of Lecture Notes in Computer Science* Springer, 196–215.

Ahman, D. and Uustalu, T. (2014). Coalgebraic update lenses. In: Jacobs, B., Silva, A. and Staton, S. (eds.) *Proceedings of the 30th Conference on the Mathematical Foundations of Programming Semantics, MFPS 2014, Ithaca, NY, USA, June 12-15, 2014, volume 308 of Electronic Notes in Theoretical Computer Science*, Elsevier, 25–48.

Ahman, D., Uustalu, T. and TYPES (2013) Update monads: cointerpreting directed containers. In: Matthes, R. and Schubert, A. (eds.) *19th International Conference on Types for Proofs and Programs, TYPES 2013, April 22-26, 2013, Toulouse, France, volume 26 of LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 1, 23.

Alves, S., Kesner, D. and Ramos, M. (2023) Quantitative global memory. In: Hansen, H. H., Scedrov, A. and de Queiroz, R. J. G. B. (eds.) *Logic, Language, Information, and Computation - 29th International Workshop, WoLLIC 2023, Halifax, NS, Canada, July 11-14, 2023, Proceedings, volume 13923 of Lecture Notes in Computer Science*, Springer, 53–68.

Alves, S., Kesner, D., Ventura, D. and TYPES(2019) A quantitative understanding of pattern matching, In: Bezem, M. and Mahboubi, A. (eds.) *25th International Conference on Types for Proofs and Programs, TYPES 2019, June 11–14, 2019, Oslo, Norway, volume 175 of LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik. 2019, 3:1–3:36.

Ariola, Z. M. and Felleisen, M. (1997). The call-by-need lambda calculus. *Journal of Functional Programming* **7** (3) 265–301.

Atkey, R. (2018). Syntax and semantics of quantitative type theory. In: Dawar, A. and Grädel, E. (eds.) *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, (LICS), 2018, Oxford, UK*, ACM, 56–65.

Barendregt, H. P. (1985). *The Lambda Calculus - Its Syntax and Semantics, Volume 103 of Studies in Logic and the Foundations of Mathematics*, North-Holland.

Benton, N., Kennedy, A., Beringer, L. and Hofmann, M. (2009) Relational semantics for effect-based program transformations: higher-order store. In: Porto, A. and López-Fraguas, F. J. (eds.) *11th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, (PPDP)*, ACM, 301–312.

Bohannon, A., Foster, J. N., Pierce, B. C., Pilkiewicz, A. and Schmitt, A. (2008). Boomerang: resourceful lenses for string data. In: Necula, G. C. and Wadler, P. (eds.) *Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7–12, 2008*, ACM, 407–419.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Bohannon, A., Pierce, B. C. and Vaughan, J. A. (2006). Relational lenses: a language for updatable views. In: Vansummeren, S. (eds.) *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 26–28, 2006, Chicago, Illinois, USA*, ACM, 338, 347.

Breuvart, F. and Dal Lago, U. (2018). On intersection types and probabilistic lambda calculi. In: Sabel, D. and Thiemann, P. (eds.) *Proceedings of the 20th International Symposium on Principles and Practice of Declarative Programming, (PPDP), 2018, Frankfurt am Main, Germany*, ACM, 8:1–8:13.

Bucciarelli, A., Kesner, D., Ríos, A. and Viso, A. (2020). The bang calculus revisited, In: *Functional and Logic Programming, 15th International Symposium, (FLOPS), 2020, Akita, Japan, Proceedings, Volume 12073 of Lecture Notes in Computer Science* Springer, 13–32.

Bucciarelli, A., Kesner, D., Ríos, A. and Viso, A. (2023). The bang calculus revisited. *Information and Computation*. **293**, 105047.

Carraro, A. and Guerrieri, G. (2014) A semantical and operational account of call-by-value solvability, Grenoble. In: *17th International Conference on Foundations of Software Science and Computation Structures, (FOSSACS), 2014, Grenoble, France, Volume 8412 of Lecture Notes in Computer Science*, Springer, 103–118.

Coppo, M. and Dezani-Ciancaglini, M. (1978). A new type assignment for lambda-terms. *Archiv für Mathematische Logik und Grundlagenforschung*, **19** 139–156.

Dal Lago, U., de Visme, M., Mazza, D. and Yoshimizu, A. (2019). Intersection types and runtime errors in the pi-calculus. *Proceedings of the ACM on Programming Languages*. **3** (POPL) 7:1–7:29.

Dal Lago, U., Faggian, C. and Ronchi Della Rocca, S. (2021). Intersection types and (positive) almost-sure termination. *Proceedings of the ACM on Programming Languages*, **5** (POPL) 1–32.

Davies, R. and Pfenning, F. (2000). Intersection types and computational effects. In: *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming, (ICFP), 2000, Montreal, Canada*, ACM, 198–208.

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

de Carvalho, D. (2018). Execution time of λ-terms via denotational semantics and intersection types. *Mathematical Structures in Computer Science*. **28** (7) 1169–1203.

de'Liguoro, U. and Treglia, R. (2021) Intersection types for a λ-calculus with global store. In: Veltri, N., Benton, N. and Ghilezan, S. (eds.) *23rd International Symposium on Principles and Practice of Declarative Programming, (PPDP), 2021, Tallinn, Estonia*, ACM, 5:1–5:11.

Dezani-Ciancaglini, M., Giannini, P. and Ronchi Della Rocca, S. (2009). Intersection, universally quantified, and reference types, Coimbra, Portugal. In: *18th Annual Conference of the EACSL on Computer Science Logic, 23rd International Workshop, (CSL), 2009, Coimbra, Portugal. Proceedings, Volume 5771 of Lecture Notes in Computer Science*, Springer, 209–224.

Ehrhard, T. (2012). Collapsing non-idempotent intersection types, Fontainebleau. In: *26th International Workshop/21st Annual Conference of the EACSL on Computer Science Logic, (CSL), France 16*. 259–273, Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Faggian, C. and Ronchi Della Rocca, S. (2019). Lambda calculus and probabilistic computation. In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, (LICS), 2019, Vancouver, BC, Canada*, IEEE, 1–13.

Gardner, P. (1994). Discovering needed reductions using type theory. In: Hagiya, M. and Mitchell, J. C. (eds.) *Theoretical Aspects of Computer Software, International Conference TACS '94*, Springer, 555–574.

Gavazzo, F., Treglia, R. and Vanoni, G. (2024-2024) Monadic intersection types, relationally, proceedings. In: Weirich, S. (eds.) *Programming Languages and Systems - 33rd European Symposium on Programming, ESOP 2024, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2024, Luxembourg City, Luxembourg, April 6-11, 2024, Proceedings, Part I, volume 14576 of Lecture Notes in Computer Science*, Springer, 22–51.

Gay, S. J. and Vasconcelos, V. T. (2010). Linear type theory for asynchronous session types. *Journal of Functional Programming*. **20** (1) 19–50.

Girard, J. (1987). Linear logic. *Theoretical Computer Science*. **50** 1–102.

Guerrieri, G. (2018) Towards a semantic measure of the execution time in call-by-value lambda-calculus. In: *12th Workshop on Developments in Computational Models and Ninth Workshop on Intersection Types and Related Systems, (DCM/ITRS), 2018, Oxford, UK, Volume 293 of EPTC*, 57–72.

Jones, S. L. P. and Wadler, P. (1993). Imperative functional programming. In: *Conference Record of the Twentieth Annual (ACM) (SIGPLAN-SIGACT) Symposium on Principles of Programming Languages, (POPL), 1993, Charleston, South Carolina, USA*, ACM Press, 71–84. 1993

Kerinec, A., Manzonetto, G. and Ronchi Della Rocca, S. (2021). Call-by-value, again!, Buenos Aires. In: *6th International Conference on Formal Structures for Computation and Deduction, (FSCD), 2021, Buenos Aires, Argentina (Virtual Conference), Volume 195 of LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 7:1–7:18.

Kesner, D., Peyrot, L. and (FSCD). (2022). Solvability for generalized applications. In: *7th International Conference on Formal Structures for Computation and Deduction*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 18:1–18:22.

Kesner, D. and Vial, P. (2020). Non-idempotent types for classical calculi in natural deduction style. *Logical Methods in Computer Science*. **16** (1) 1–46.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Kesner, D. and Viso, A. (2022). Encoding tight typing in a unified framework, Göttingen, *30th EACSL Annual Conference on Computer Science Logic, (CSL), 2022, Göttingen, Germany (Virtual Conference), Volume 216 of LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 27:1–27:20.

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

Landin, P. J. (1965). Correspondence between ALGOL 60 and church's lambda-notation: part I. *Communications of the ACM* **8** (2) 89–101.

Launchbury, J. (1993) A natural semantics for lazy evaluation, In: Deusen, M. S. V. and Lang, B. (eds.) *Conference Record of the Twentieth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Charleston, South Carolina, USA*, ACM Press, 144–154.

Manzonetto, G., Pagani, M. and Ronchi Della Rocca, S. (2019). New semantical insights into call-by-value λ-calculus. *Fundamenta Informaticae* **170** (1-3) 241–265.

Moggi, E. (1989). Computational lambda-calculus and monads. In: *4th Annual Symposium on Logic in Computer Science, (LICS), 1989, Pacific Grove, California, USA*, IEEE Computer Society, 14–23.

Moggi, E. (1991). Notions of computation and monads. *Information and Computation* **93** (1) 55–92.

Plotkin, G. D. (1975). Call-by-name, call-by-value and the lambda-calculus. *Theoretical Computer Science* **1** (2) 125–159.

Plotkin, G. D. and Power, J. (2001). Adequacy for algebraic effects. In: Honsell, F. and Miculan, M. (eds.) *Foundations of Software Science and Computation Structures, 4th International Conference, FOSSACS 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2-6, 2001, Proceedings, volume 2030 of Lecture Notes in Computer Science*, Springer, 1–24.

Plotkin, G. D. and Power, J. (2008). Tensors of comodels and models for operational semantics. In: Bauer, A. and Mislove, M. W. (eds.) *Proceedings of the 24th Conference on the Mathematical Foundations of Programming Semantics, MFPS 2008, Philadelphia, PA, USA, May 22–25, 2008, Volume 218 of Electronic Notes in Theoretical Computer Science*, Elsevier, 295–311.

Sabry, A. and Wadler, P. (1997). A reflection on call-by-value. *ACM Transactions on Programming Languages and Systems* **19** (6) 916–941.

Wadler, P. (1990). Comprehending monads. In: *Proceedings of the 1990 ACM conference on LISP and functional programming, LFP '90, New York, NY, USA*, Association for Computing Machinery, 61–78.

Wadler, P. (1995) Monads for functional programming. In: *1st International Spring School on Advanced Functional Programming Techniques on Advanced Functional Programming (AFP)*, Springer, 24–52.

Appendix A. Proofs

A.1 Weak Open CBN and CBV

A.1.1 Weak Open CBN

Proposition 1 (Normal forms for Weak Open CBN). *Let t be a term. Then $t \in \mathtt{no}_{\mathtt{n}}$ iff $t \not\to_{\mathtt{n}}$.*

Proof. We refine the original statement as follows:

(1) $t \not\to_{\mathtt{n}}$ and $\neg\mathtt{abs}(t)$ iff $t \in \mathtt{ne}_{\mathtt{n}}$.
(2) $t \not\to_{\mathtt{n}}$ iff $t \in \mathtt{no}_{\mathtt{n}}$.

$\Rightarrow$) By induction over t:

(1) Let $t \not\to_{\mathtt{n}}$ and $\neg\mathtt{abs}(t)$:
   - Case $t = x$. Then $x \in \mathtt{ne}_{\mathtt{n}}$ by definition.
   - Case $t = \lambda x.u$. Then $\mathtt{abs}(t)$ holds, which contradicts the hypothesis. Therefore, this case cannot apply.
   - Case $t = up$. Since $up \not\to_{\mathtt{n}}$, no rule of $\to_{\mathtt{n}}$ applies to up. Then $\neg\mathtt{abs}(u)$, otherwise ($\beta_{\mathtt{n}}$) would apply, and $u \not\to_{\mathtt{n}}$, in which case (appL) would apply. By the *i.h.* ((1)) $u \in \mathtt{ne}_{\mathtt{n}}$, therefore $up \in \mathtt{ne}_{\mathtt{n}}$.

(2) Let $t \not\to_{\mathtt{n}}$:
   - Case $t = \lambda x.u$. Then $\lambda x.u \in \mathtt{no}_{\mathtt{n}}$ by definition.
   - Case $\neg\mathtt{abs}(t)$. Then $t \in \mathtt{ne}_{\mathtt{n}} \subseteq \mathtt{no}_{\mathtt{n}}$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$\Leftarrow$)

(1) By induction over $t \in \mathtt{ne}_\mathtt{n}$:

- Case $t = x$. Then $x \not\to_\mathtt{n}$ and $\neg\mathtt{abs}(x)$ by definition.
- Case $t = up$, such that $u \in \mathtt{ne}_\mathtt{n}$. By the *i.h.* (1), we know $u \not\to_\mathtt{n}$ and $\neg\mathtt{abs}(u)$. Therefore, neither (appL) nor ($\beta_\mathtt{n}$) apply.

(2) Let $t \in \mathtt{no}_\mathtt{n}$:

- Case $t = \lambda x.u$. Then $\lambda x.u \not\to_\mathtt{n}$ by definition.
- Case $\neg\mathtt{abs}(t)$. Then $t \in \mathtt{ne}_\mathtt{n}$ by definition, therefore $t \not\to_\mathtt{n}$, by point (1). □

Proposition 7 (Relevance). *Let* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_\mathtt{n}} t : \tau$. *Then* $\mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(t)$.

Proof. By a simple induction over Φ. □

Lemma 21 (Tight spreading). *Let* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_\mathtt{n}} t : \tau$, *such that* Γ *is tight. If* $t \in \mathtt{ne}_\mathtt{n}$, *then* τ *is a tight constant.*

Proof. The proof follows by induction over $t \in \mathtt{ne}_\mathtt{n}$:

- Case $t = x$. Then $\Phi \rhd x : [\tau] \overset{(0,0)}{\vdash_\mathtt{n}} x : \tau$, with $x : [\tau]$ tight, therefore $\tau \in \mathtt{tt}_\mathtt{n}$.
- Case $t = up$, such that $u \in \mathtt{ne}_\mathtt{n}$. Then Φ must end with rule ($@$) or ($@_\mathtt{p}$):
  - Case Φ ends with rule ($@$). Then Φ has two premises $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_\mathtt{n}} u : \mathscr{M} \to \tau$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_\mathtt{n}} p : \mathscr{M}$, such that $\Gamma = \Gamma_u + \Gamma_p$ is tight, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Moreover, Γ_u and Γ_p are tight. By the *i.h.* on Φ_u, we have $\mathscr{M} \to \tau \in \mathtt{tt}_\mathtt{n}$, which is a contradiction. Therefore, this case does not apply.
  - Case Φ ends with rule ($@_\mathtt{p}$). Then, Φ has one premise $\Phi_u \rhd \Gamma \overset{(b,s_u)}{\vdash_\mathtt{n}} u : \mathtt{n}$, meaning in particular that $\tau = \mathtt{n}$. Thus, we can conclude since $\tau \in \mathtt{tt}_\mathtt{n}$. □

Lemma 22 (Persistent abstractions have tight type). *If* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_\mathtt{n}} t : \tau$. *If* $\tau \in \overline{\mathtt{a}}$, *then* $\neg\mathtt{abs}(t)$.

Proof. By induction over Φ:

- Case Φ ends with rule (ax), ($@$), and ($@_\mathtt{p}$). Then $\neg\mathtt{abs}(t)$ holds by definition.
- Case Φ ends with rule (λ), ($\mathtt{m}$), or ($\lambda_\mathtt{p}$). Then $\tau \notin \overline{\mathtt{a}}$, and then these cases hold vacuously. □

Lemma 1 (Normal forms take zero steps). *Let* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_\mathtt{n}} t : \tau$ *be a tight type derivation. Then,* $b = 0$ *iff* $t \in \mathtt{no}$.

Proof.

$\Rightarrow$) We refine the original statement as follows:

(1) Let $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_\mathtt{n}} t : \tau$ be a tight derivation. Then $b = 0$ and $\neg\mathtt{abs}(t)$ implies $t \in \mathtt{ne}_\mathtt{n}$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

(2) Let $\Phi \triangleright \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \tau$ be a tight derivation. Then $b = 0$ implies $t \in \mathtt{no}_{\mathtt{n}}$.

(1) Let $b = 0$ and $\neg\mathtt{abs}(t)$. The proof follows by induction over Φ, according to the last rule of Φ:

- Rule (λ) or ($\lambda_{\mathtt{p}}$). Then $\mathtt{abs}(t)$ holds, which contradicts the hypothesis. Therefore, these cases cannot apply.
- Rule ($\mathtt{ax}$). Then $t = x$ and $x \in \mathtt{ne}_{\mathtt{n}}$.
- Rule ($\mathtt{m}$). Then $\tau \notin \mathtt{tt}_{\mathtt{n}}$, and thus, Φ is not tight, which contradicts the hypothesis. Therefore, this cannot apply.
- Rule ($@$). Then $b > 0$ so that this case holds vacuously.
- Rule ($@_{\mathtt{p}}$). Then $t = up$ and Φ has one premise $\Phi_u \triangleright \Gamma \overset{(0,s_u)}{\vdash_{\mathtt{n}}} u : \mathtt{n}$, such that $\tau = \mathtt{n}$. Also, since $\mathtt{n} \in \overline{\mathtt{a}}$ then $\neg\mathtt{abs}(u)$, according to Lemma 22. Therefore, by the *i.h.* (1), we have $u \in \mathtt{ne}_{\mathtt{n}}$. Thus, $up \in \mathtt{ne}_{\mathtt{n}}$ as required.

(2) Let $b = 0$:

- Case $t = \lambda x.u$. Then $\lambda x.u \in \mathtt{no}_{\mathtt{n}}$ by definition.
- Case $\neg\mathtt{abs}(t)$. Then $t \in \mathtt{ne}_{\mathtt{n}}$, by point (1), which implies $t \in \mathtt{no}_{\mathtt{n}}$.

$\Leftarrow$)

(1) The proof follows by induction over $t \in \mathtt{ne}_{\mathtt{n}}$:

- Case $t = x$. Then $b = 0$ and $\neg\mathtt{abs}(x)$, by definition.
- Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{n}}$. Then Φ must end with rule ($@$) or ($@_{\mathtt{p}}$):
  - Rule ($@$). Then Φ has two premises $\Phi_u \triangleright \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathscr{M} \to \tau$ and $\Phi_p \triangleright \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{n}}} p : \mathscr{M}$, such that $\Gamma = \Gamma_u + \Gamma_p$ tight, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Therefore, $\mathscr{M} \to \tau$ turns to be tight by Lemma 21, which is a contradiction. Therefore, this case cannot apply.
  - Rule ($@_{\mathtt{p}}$). Then Φ has one premise $\Phi_u \triangleright \Gamma \overset{(b,s_u)}{\vdash_{\mathtt{n}}} u : \mathtt{n}$, where $\tau = \mathtt{n}$. Therefore, $b = 0$ by applying point (2) to Φ_u. Moreover, $\neg\mathtt{abs}(up)$ trivially holds.

(2) Case $t \in \mathtt{no}_{\mathtt{n}}$:

- Case $t = \lambda x.u$. Then, Φ must end with (λ), ($\mathtt{m}$) or ($\lambda_{\mathtt{p}}$). The first two cases cannot apply, since $\tau \notin \mathtt{tt}_{\mathtt{n}}$, and thus, Φ cannot be tight. For the last case, $b = 0$ holds by definition.
- Case $\neg\mathtt{abs}(t)$. Then $t \in \mathtt{ne}_{\mathtt{n}}$ by definition. Therefore, $b = 0$ by applying point (1) to t.

□

Lemma 2 (Normal forms have correct size). *Let* $\Phi \triangleright \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \tau$ *be a tight type derivation. If* $b = 0$ *then* $s = |t|_{\mathtt{n}}$.

Proof. The proof follows by induction over Φ, according to the last rule of Φ:

- Rule ($\mathtt{ax}$) or ($\lambda_{\mathtt{p}}$). Then $t = x$ or $t = \lambda x.u$. Moreover, $s = 0 = |x| = |\lambda x.u|$ by definition.
- Rule (λ) or ($\mathtt{m}$). Then $\tau \notin \mathtt{tt}_{\mathtt{n}}$, which contradicts that Φ is tight. Therefore, these cases cannot apply.
- Rule ($@$). Then $b > 0$; therefore, this case holds vacuously.
- Rule ($@_{\mathtt{p}}$). Then $t = up$ and Φ has one premise $\Phi_u \triangleright \Gamma \overset{(b,s_u)}{\vdash_{\mathtt{n}}} u : \mathtt{n}$, where $\tau = \mathtt{n}$, $b = 0$, and $s = 1 + s_u$. Therefore, $s_u = |u|_{\mathtt{n}}$ by applying the *i.h.* to Φ_u. Thus, we can conclude with $s = 1 + s_u = 1 + |u|_{\mathtt{n}} = |up|_{\mathtt{n}}$.

□

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 3 (All normal forms are tightly typable). *If* $t \in \mathtt{no}_{\mathtt{n}}$, *then there exists a tight type derivation* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \tau$, *such that* $s = |t|_{\mathtt{n}}$.

We refine the original statement as follows:

(1) If $t \in \mathtt{ne}_{\mathtt{n}}$, there exists a tight derivation $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \mathtt{n}$ such that $s = |t|_{\mathtt{n}}$.

(2) If $t \in \mathtt{no}_{\mathtt{n}}$, there exists a tight derivation $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \mathtt{tt}_{\mathtt{n}}$ such that $s = |t|_{\mathtt{n}}$.

(1) The proof follows by induction over $t \in \mathtt{ne}_{\mathtt{v}}$:

- Case $t = x$. Then we can build Φ as follows:

$$\frac{}{x : [\mathtt{n}] \overset{(0,0)}{\vdash} x : \mathtt{n}}(\mathtt{ax})$$

  such that $\Gamma = (x : [\mathtt{n}])$, $b = 0$, and $s = 0 = |x|_{\mathtt{n}}$.
- Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{n}}$. By applying the *i.h.* (1) to Φ_u, there exists a tight derivation $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathtt{n}$, such that $|t|_{\mathtt{n}} = s_u$. Therefore, we can build Φ as follows:

$$\frac{\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathtt{n}}{\Gamma_u \overset{(b_u,1+s_u)}{\vdash} up : \mathtt{n}}(@_{\mathtt{p}})$$

  such that $\Gamma = \Gamma_u$, $b = b_u$, and $s = 1 + s_u = 1 + |u|_{\mathtt{n}} = |up|_{\mathtt{n}}$.

(2) The proof follows by induction over $t \in \mathtt{no}_{\mathtt{n}}$:

- Case $t = \lambda x.u$. Then we can build Φ as follows:

$$\frac{}{\emptyset \overset{(0,0)}{\vdash} \lambda x.u : \mathtt{a}}(\lambda_{\mathtt{p}})$$

  such that $\Gamma = \emptyset$, $b = 0$ and $s = 0 = |\lambda x.u|_{\mathtt{n}}$.
- Case $\neg\mathtt{abs}(t)$. Then $t \in \mathtt{ne}_{\mathtt{n}}$ by definition. Therefore, $s = |t|_{\mathtt{n}}$ by applying point (1) to t.

Lemma 23 (Multi-types can split and merge). *Let* $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b = +_{i\in I} b_i$, *and* $s = +_{i\in I} s_i$. *Then,* $\Phi_t \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \mathscr{M}$ *iff* $(\Phi^i_t \rhd \Gamma_i \overset{(b_i,s_i)}{\vdash_{\mathtt{n}}} t : \mathscr{M}_i)_{i\in I}$.

Proof.

$\Rightarrow$) We start by noting that Φ_t must end with the rule $(\mathtt{m})$. Therefore, we have $\Gamma = +_{j\in J}\Gamma_j$, $\mathscr{M} = [\tau_j]_{j\in J}$, $b = +_{j\in J} b_j$, $s = +_{j\in J} s_j$, and $(\Phi^j_t \rhd \Gamma_j \overset{(b_j,s_j)}{\vdash_{\mathtt{n}}} t : \tau_j)_{j\in J}$, for some J. Let $\mathscr{M}_i = [\tau_k]_{k\in K_i}$, for each $i \in I$, such that $J = +_{i\in I} K_i$. Then, by using rule $(\mathtt{m})$, we can build $\Phi^i_t \rhd \Gamma_i \overset{(b_i,s_i)}{\vdash_{\mathtt{n}}} t : \mathscr{M}_i$, for each $i \in I$, such that $\Gamma_i = +_{k\in K_i}\Gamma_k$, $b_i = +_{k\in K_i} b_k$, and $s_i = +_{k\in K_i} s_k$. So we can conclude with $\Gamma = +_{j\in J}\Gamma_j = +_{i\in I}(+_{k\in K_i}\Gamma_k) = +_{i\in I}\Gamma_i$, $b = +_{j\in J} b_j = +_{i\in I}(+_{k\in K_i} b_k) = +_{i\in I} b_i$, and $s = +_{j\in J} s_j = +_{i\in I}(+_{k\in K_i} s_k) = +_{i\in I} s_i$.

$\Leftarrow$) We start by noting that each Φ^i_t must end with the rule $(\mathtt{m})$. Therefore, for each $i \in I$, we have $\Gamma_i = +_{k\in K_i}\Gamma_k$, $\mathscr{M}_i = [\tau_k]_{k\in K_i}$, $b_i = +_{k\in K_i} b_k$ and $s_i = +_{k\in K_i} s_k$, and the following derivations

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$(\Phi_t^k \rhd \Gamma_k \overset{(b_k,s_k)}{\vdash_{\mathtt{n}}} t : \tau_k)_{k\in K_i}$. Let $J = +_{i\in I} K_i$ and $\mathscr{M} = [\tau_j]_{j\in J} = [\tau_k]_{k\in K_i, i\in I}$. We can use rule (m) to build $\Phi_t \rhd \Gamma \overset{(+_{j\in J} b_j, +_{j\in J} s_j)}{\vdash_{\mathtt{n}}} t : \mathscr{M}$. So we can conclude with $\Gamma = +_{j\in J}\Gamma_j = +_{i\in I}(+_{k\in K_i}\Gamma_k) = +_{i\in I}\Gamma_i$, $b = +_{j\in J} b_j = +_{i\in I}(+_{k\in K_i} b_k) = +_{i\in I} b_i$, and $s = +_{j\in J} s_j = +_{i\in I}(+_{k\in K_i} s_k) = +_{i\in I} s_i$. □

Lemma 24 (Substitution). *Let* $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \overset{(b_t,s_t)}{\vdash_{\mathtt{n}}} t : \tau$ *and* $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathscr{M}$, *then there exists* $\Phi_{t\{x\setminus u\}} \rhd \Gamma_t + \Gamma_u \overset{(b_t+b_u, s_t+s_u)}{\vdash_{\mathtt{n}}} t\{x \setminus u\} : \tau$.

Proof. We generalize the original statement by allowing $\Phi_{t\{x\setminus u\}}$ to conclude with either a type τ or a multi-type $\mathscr{M}$. Let $\mathscr{G} \in \{\tau, \mathscr{M}\}$. Then the statement is as follows:

Let $\Phi_{t\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}} \overset{(b,s)}{\vdash_{\mathtt{n}}} t\{x \setminus u\} : \mathscr{G}$. Then, there exist $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \overset{(b_t,s_t)}{\vdash_{\mathtt{n}}} t : \mathscr{G}$ and $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathscr{M}$, such that $\Gamma_{t\{x\setminus u\}} = \Gamma_t + \Gamma_u$, $b = b_t + b_u$, and $s = s_t + s_u$.

The proof follows by induction over Φ_t:

- Rule (ax). Then $t = y$ and we have two cases:
  - $y = x$. Then $\Gamma_t = \emptyset$, $\mathscr{G} = \mathscr{M}$, $b_t = 0$, and $s_t = 0$. Also, notice that $x\{x \setminus u\} = u$. Therefore, we can take $\Phi_{t\{x\setminus u\}} = \Phi_u$ and conclude with $\Gamma_u = \emptyset + \Gamma_u$, $b_u = 0 + b_u$, and $s_u = 0 + s_u$.
  - $y \neq x$. Then $\mathscr{M} = [\,]$. Therefore, $\Gamma_u = \emptyset$, $b_u = 0$, and $s_u = 0$. Also, notice that $y\{x \setminus u\} = y$. Therefore, we can take $\Phi_{t\{x\setminus u\}} = \Phi_t$ and conclude with $\Gamma_t = \Gamma_t + \emptyset$, $b_t = b_t + 0$, and $s_t = s_t + 0$.
- Rule (λ). Then $t\{x \setminus u\} = \lambda y.(p\{x \setminus u\})$ (by α-conversion, we can assume $y \neq x$), and Φ_t has one premise $\Phi_p \rhd (\Gamma_p; y : \mathscr{M}'); x : \mathscr{M} \overset{(b_t,s_t)}{\vdash_{\mathtt{n}}} p : \tau'$, such that $\Gamma_t = (\Gamma_p; y : \mathscr{M}')$, and $\mathscr{G} = \mathscr{M}' \to \tau'$. By applying the *i.h.* to Φ_p, there exists $\Phi_{p\{x\setminus u\}} \rhd (\Gamma_p; y : \mathscr{M}') + \Gamma_u \overset{(b_t+b_u, s_t+s_u)}{\vdash_{\mathtt{n}}} p\{x \setminus u\} : \tau'$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{\Phi_{p\{x\setminus u\}}}{\Gamma_p + \Gamma_u \overset{(b_t+b_u, s_t+s_u)}{\vdash} (\lambda y.p)\{x \setminus u\} : \mathscr{M}' \to \tau'}(\lambda)$$

- Rule (@). Then $t\{x \setminus u\} = (r\{x \setminus u\})(p\{x \setminus u\})$, and Φ_t has two premises $\Phi_r \rhd \Gamma_r; x : \mathscr{M}_1 \overset{(b_r,s_r)}{\vdash_{\mathtt{n}}} r : \mathscr{M}' \to \tau$ and $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \overset{(b_p,s_p)}{\vdash_{\mathtt{n}}} p : \mathscr{M}'$, such that $\Gamma_t = \Gamma_r + \Gamma_p$, $\mathscr{G} = \tau$, $\mathscr{M} = \mathscr{M}_1 \sqcup \mathscr{M}_2$, $b_t = 1 + b_r + b_p$, and $s_t = s_r + s_p$. By Lemma 23, there exist $(\Phi_u^i \rhd \Gamma_u^i \overset{(b_i,s_i)}{\vdash_{\mathtt{n}}} u : \mathscr{M}_i)_{i\in\{1,2\}}$, such that $\Gamma_u = \Gamma_u^1 + \Gamma_u^2$, $b_u = b_1 + b_2$, and $s_u = s_1 + s_2$. By applying the *i.h.* to Φ_r and Φ_p, there exist $\Phi_{r\{x\setminus u\}} \rhd \Gamma_r + \Gamma_u^1 \overset{(b_r+b_1, s_r+s_1)}{\vdash_{\mathtt{n}}} r\{x \setminus u\} : \mathscr{M}' \to \tau$ and $\Phi_{p\{x\setminus u\}} \rhd \Gamma_p + \Gamma_u^2 \overset{(b_p+b_2, s_p+s_2)}{\vdash_{\mathtt{n}}} p\{x \setminus u\} : \mathscr{M}'$, respectively. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{\Phi_{r\{x\setminus u\}} \quad \Phi_{p\{x\setminus u\}}}{(\Gamma_r + \Gamma_p) + (\Gamma_u^1 + \Gamma_u^2) \overset{(1+b_r+b_p+b_1+b_2, s_r+s_p+s_1+s_2)}{\vdash} (rp)\{x \setminus u\} : \tau}(@)$$

Thus, we can conclude with $(\Gamma_r + \Gamma_p) + (\Gamma_u^1 + \Gamma_u^2) = \Gamma_t + \Gamma_u$, $1 + b_r + b_p + b_1 + b_2 = b_t + b_u$, and $s_r + s_p + s_1 + s_2 = s_t + s_u$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Rule (m). Then Φ_t has premises $(\Phi_t^i \rhd \Gamma_t^i; x : \mathscr{M}_i \overset{(b_t^i, s_t^i)}{\vdash_{\mathtt{n}}} t : \tau_i)_{i\in I}$, such that $\Gamma_t = +_{i\in I}\Gamma_t^i$, $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $\mathscr{G} = [\tau_i]_{i\in I}$, $b_t = +_{i\in I} b_t^i$, and $s_t = +_{i\in I} s_t^i$. By Lemma 23, there exist $(\Phi_u^i \rhd \Gamma_u^i \overset{(b_u^i, s_u^i)}{\vdash_{\mathtt{n}}} u : \mathscr{M}_i)_{i\in I}$, such that $\Gamma_u = +_{i\in I}\Gamma_u^i$, $b_u = +_{i\in I} b_u^i$, and $s_u = +_{i\in I} s_u^i$. By applying the *i.h.* to each Φ_t^i, there exist $(\Phi_{t\{x\setminus u\}}^i \rhd \Gamma_t^i + \Gamma_u^i \overset{(b_t^i + b_u^i, s_t^i + s_u^i)}{\vdash_{\mathtt{n}}} t\{x \setminus u\} : \tau_i)_{i\in I}$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{(\Phi_{p\{x\setminus u\}}^i)_{i\in I}}{+_{i\in I}(\Gamma_t^i + \Gamma_u^i) \overset{(+_{i\in I}(b_t^i + b_u^i), +_{i\in I}(s_t^i + s_u^i))}{\vdash} t\{x \setminus u\} : [\tau_i]_{i\in I}}(\mathtt{m})$$

Thus, we can conclude with $+_{i\in I}(\Gamma_t^i + \Gamma_u^i) = \Gamma_t + \Gamma_u$, $+_{i\in I}(b_t^i + b_u^i) = b_t + b_u$, and $+_{i\in I}(s_t^i + s_u^i) = s_t + s_u$.

- Rule ($\lambda_{\mathtt{p}}$). Then $t\{x \setminus u\} = \lambda y.(p\{x \setminus u\})$ (by α-conversion, we can assume $y \neq x$), and $\Phi_t \rhd \emptyset \overset{(0,0)}{\vdash_{\mathtt{n}}} \lambda y.p : \mathtt{a}$, such that $\Gamma_t = \emptyset, \mathscr{G} = \mathtt{a}$, and $\mathscr{M} = \emptyset$. Therefore, $\Gamma_u = \emptyset$, $b_u = s_u = 0$, and we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{}{\emptyset \overset{(0,0)}{\vdash} \lambda y.(p\{x \setminus u\}) : \mathtt{a}}(\lambda_{\mathtt{p}})$$

Thus, we can conclude with $\emptyset = \Gamma_t + \Gamma_u$, $0 = b_t + b_u$, and $0 = s_t + s_u$.

- Rule ($@_{\mathtt{p}}$). Then, $t\{x \setminus u\} = (r\{x \setminus u\})(p\{x \setminus u\})$, and Φ_t has one premise $\Phi_r \rhd \Gamma_t; x : \mathscr{M} \overset{(b_r, s_r)}{\vdash_{\mathtt{n}}} r : \mathtt{n}$, such that $\mathscr{G} = \mathtt{n}$, $b_t = b_r$, and $s_t = 1 + s_r$. By applying the *i.h.* to Φ_r, there exists $\Phi_{r\{x\setminus u\}} \rhd \Gamma_r + \Gamma_u \overset{(b_r + b_u, s_r + s_u)}{\vdash_{\mathtt{n}}} r\{x \setminus u\} : \mathtt{n}$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{\Phi_{r\{x\setminus u\}}}{\Gamma_t + \Gamma_u \overset{(b_r + b_u, 1 + s_r + s_u)}{\vdash} (r\{x \setminus u\})(p\{x \setminus u\}) : \mathtt{n}}(@_{\mathtt{p}})$$

Thus, we can conclude with $b_r + b_u = b_t + b_u$, and $1 + s_r + s_u = s_t + s_u$. □

Lemma 4 (Quantitative subject reduction). *Let* $\Phi_t \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \tau$ *be a tight type derivation. If* $t \to t'$, *then there exists* $\Phi_{t'} \rhd \Gamma \overset{(b-1,s)}{\vdash_{\mathtt{n}}} t' : \tau$.

Proof. We prove a stronger version of the statement that allows us to reason inductively:

Let $\Phi_t \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \tau$, such that Γ is tight, and either τ is tight or $\neg\mathtt{abs}(t)$. If $t \to t'$, then there exists $\Phi_{t'} \rhd \Gamma \overset{(b-1,s)}{\vdash_{\mathtt{n}}} t' : \tau$.

The proof follows by induction over $t \to t'$:

- Case $t = (\lambda x.p)u \to p\{x \setminus u\} = t'$. Then Φ_t must end with rule (@) or ($@_{\mathtt{p}}$). Let us assume that Φ_t ends with rule ($@_{\mathtt{p}}$). Then, Φ_t has one premise $\Phi_{\lambda x.p}$. Moreover, $\Phi_{\lambda x.p}$ must conclude with type $\mathtt{n} \in \overline{\mathtt{a}}$. However, this is not possible according to Lemma 22. Therefore, this case does not apply. Now, let us assume that Φ_t ends with rule (@). Then, Φ_t has the following form:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\dfrac{\dfrac{\Phi_p \rhd \Gamma_p; x : \mathscr{M} \overset{(b_p,s_p)}{\vdash_{\mathtt{n}}} p : \tau}{\Gamma_p \overset{(b_p,s_p)}{\vdash} (\lambda x.p) : \mathscr{M} \to \tau}(\lambda) \qquad \Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathscr{M}}{\Gamma_p + \Gamma_u \overset{(1+b_p+b_u,s_p+s_u)}{\vdash} (\lambda x.p)u : \tau}(@)$$

such that $\Gamma = \Gamma_p + \Gamma_u$ tight, $\tau \in \mathtt{tt}_{\mathtt{n}}$, $b = 1 + b_p + b_u$, and $s = s_p + s_u$. By Lemma 24, there exists $\Phi_{p\{x\setminus u\}} \rhd \Gamma_p + \Gamma_u \overset{(b_p+b_u,s_p+s_u)}{\vdash_{\mathtt{n}}} p\{x \setminus u\} : \tau$. Therefore, we can take $\Phi_{t'} = \Phi_{p\{x\setminus u\}}$ and conclude with $b - 1 = b_p + b_u$.

- Case $t = up \to u'p = t'$, such that $u \to u'$. Then Φ_t must either end with $(@)$ or $(@_{\mathtt{p}})$:
  - Case Φ_t ends with rule $(@)$. Then Φ_t has the following form:

$$\dfrac{\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathscr{M} \to \tau \qquad \Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{n}}} p : \mathscr{M}}{\Gamma_u + \Gamma_p \overset{(1+b_u+b_p,s_u+s_p)}{\vdash} up : \tau}(@)$$

    such that $\Gamma = \Gamma_u + \Gamma_p$ tight, $\tau \in \mathtt{tt}_{\mathtt{n}}$, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Moreover, Γ_u and Γ_p are tight. Also, notice that $\neg\mathtt{abs}(u)$ since $u \to u'$, by definition. Therefore, by applying the *i.h.* to Φ_u, there exists $\Phi_{u'} \rhd \Gamma_u \overset{(b_u-1,s_u)}{\vdash_{\mathtt{n}}} u' : \mathscr{M} \to \tau$. Therefore, we can build $\Phi_{t'}$ as follows:

$$\dfrac{\Phi_{u'} \qquad \Phi_p}{\Gamma_u + \Gamma_p \overset{(b_u+b_p,s_u+s_p)}{\vdash} u'p : \tau}(@)$$

    Thus, we can conclude with $b - 1 = b_u + b_p$.
  - Case Φ_t ends with rule $(@_{\mathtt{p}})$. Then Φ_t has one premise $\Phi_u \rhd \Gamma \overset{(b,s_u)}{\vdash_{\mathtt{n}}} u : \mathtt{n}$, such that $s = 1 + s_u$. By applying the *i.h.* to Φ_u, there exists $\Phi_{u'} \rhd \Gamma \overset{(b-1,s_u)}{\vdash_{\mathtt{n}}} u' : \mathtt{n}$. Therefore, we can build $\Phi_{t'}$ as follows:

$$\dfrac{\Phi_{u'}}{\Gamma \overset{(b-1,1+s_u)}{\vdash} u'p : \mathtt{n}}(@_{\mathtt{p}})$$

    Thus, we can conclude with $s = 1 + s_u$. □

Theorem 1 (Quantitative soundness). *If* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t : \tau$ *is a tight type derivation, then there exists* $u \in \mathtt{no}_{\mathtt{n}}$ *such that* $t \twoheadrightarrow_{\mathtt{n}}^{b} u$ *with* $|u|_{\mathtt{n}} = s$.

Proof. The proof follows by induction over b:

- Case $b = 0$. Then $t \in \mathtt{no}_{\mathtt{n}}$ by Lemma 1. Moreover, $s = |t|_{\mathtt{n}}$, by Lemma 2. So we can conclude with $u = t$.
- Case $b > 0$. Then $t \notin \mathtt{no}_{\mathtt{n}}$ by Lemma 1. Therefore, by Proposition 1, there exists t' such that $t \to_{\mathtt{n}} t'$. Moreover, this derivation is also tight. By Lemma 4, there exists $\Phi_{t'} \rhd \Gamma \overset{(b-1,s)}{\vdash_{\mathtt{n}}} t' : \tau$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

By the *i.h.*, there exists $u \in \mathtt{no}_\mathtt{n}$, such that $t' \twoheadrightarrow_\mathtt{n}^{b-1} u$, such that $s = |u|_\mathtt{n}$. So we can conclude with $t \to_\mathtt{n} t' \twoheadrightarrow_\mathtt{n}^{b-1} u$, which means that $t \twoheadrightarrow_\mathtt{n}^{b} u$, as expected. □

Lemma 25 (Anti-substitution). *Let* $\Phi_{t\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}} \overset{(b,s)}{\vdash_\mathtt{n}} t\{x \setminus u\} : \tau$. *Then, there exist* $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \overset{(b_t,s_t)}{\vdash_\mathtt{n}} t : \tau$ *and* $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_\mathtt{n}} u : \mathscr{M}$, *such that* $\Gamma_{t\{x\setminus u\}} = \Gamma_t + \Gamma_u$, $b = b_t + b_u$, *and* $s = s_t + s_u$.

Proof. We generalize the original statement by allowing $\Phi_{t\{x\setminus u\}}$ to conclude with either a type τ or a multi-type $\mathscr{M}$. Let $\mathscr{G} \in \{\tau, \mathscr{M}\}$. Then the statement is as follows:

Let $\Phi_{t\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}} \overset{(b,s)}{\vdash_\mathtt{n}} t\{x \setminus u\} : \mathscr{G}$. Then, there exist $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \overset{(b_t,s_t)}{\vdash_\mathtt{n}} t : \mathscr{G}$ and $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_\mathtt{n}} u : \mathscr{M}$, such that $\Gamma_{t\{x\setminus u\}} = \Gamma_t + \Gamma_u$, $b = b_t + b_u$, and $s = s_t + s_u$.

The proof follows by induction over $\Phi_{t\{x\setminus u\}}$, according to the last rule of $\Phi_{t\{x\setminus u\}}$:

Let us assume $t = x$, such that $t\{x \setminus u\} = u$. Then, we have to consider two cases: $\mathscr{G} = \tau$, or $\mathscr{G} = \mathscr{M}'$. If $\Phi_{t\{x\setminus u\}}$ concludes with τ, we can build $\Phi_t \rhd x : [\tau] \overset{(0,0)}{\vdash_\mathtt{n}} x : \tau$, such that $\Gamma_t = \emptyset$, and $b_t = s_t = 0$, by using rule (ax), and build $\Phi_u \rhd \Gamma_{t\{x\setminus u\}} \overset{(b,s)}{\vdash_\mathtt{n}} u : \tau$, such that $\Gamma_u = \Gamma_{t\{x\setminus u\}}$, $b_u = b$, and $s_u = s$, by applying rule (m) to $\Phi_{t\{x\setminus u\}}$. If $\Phi_{t\{x\setminus u\}}$ concludes with $\mathscr{M}' = [\tau_i]_{i\in I}$, we pick $\Phi_u = \Gamma_{t\{x\setminus u\}}$, such that $\Gamma_u = \Gamma_{t\{x\setminus u\}}$, $b_u = b$, and $s_u = s$, and build Φ_t as follows:

$$\dfrac{\left(\dfrac{}{x : [\tau_i] \overset{(0,0)}{\vdash} x : \tau_i}\,(\mathtt{ax})\right)_{i\in I}}{x : [\tau_i]_{i\in I} \overset{(0,0)}{\vdash} x : [\tau_i]_{i\in I}}\,(\mathtt{m})$$

such that $\Gamma_t = \emptyset$, and $b_t = s_t = 0$. In both cases, we can conclude with $\Gamma_{t\{x\setminus u\}} = \Gamma_u = \Gamma_t + \Gamma_u$, $b = b_u = b_t + b_u$, and $s = s_u = s_t + s_u$. From now on, we will assume $t \neq x$:

- Rule (ax). Then $t = y$, $t\{x \setminus u\} = y$, and $\mathscr{M} = [\,]$. Therefore, we can pick $\Phi_t = \Phi_{t\{x\setminus u\}}$, such that $\Gamma_t = \Gamma_{t\{x\setminus u\}}$, $b_t = b$, and $s_t = s$, and build $\Phi_u \rhd \emptyset \overset{(0,0)}{\vdash_\mathtt{n}} u : [\,]$, such that $\Gamma_u = \emptyset$, and $b_u = s_u = 0$, by using rule (m) with no premises. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_t = \Gamma_t + \Gamma_v$, $b = b_t = b_t + b_v$, and $s = s_t = s_t + s_v$.
- Rule (λ). Then $t = \lambda y.p$ and $t\{x \setminus u\} = \lambda y.(p\{x \setminus u\})$. Moreover, $\Phi_{t\{x\setminus u\}}$ has one premise $\Phi_{p\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus v\}}; y : \mathscr{M}' \overset{(b,s)}{\vdash_\mathtt{n}} p\{x \setminus u\} : \tau'$, such that $\mathscr{G} = \mathscr{M}' \to \tau'$. By applying the *i.h.* to $\Phi_{p\{x\setminus u\}}$, there exist $\Phi_p \rhd (\Gamma_p; y : \mathscr{M}'); x : \mathscr{M} \overset{(b_p,s_p)}{\vdash_\mathtt{n}} p : \tau'$ and $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_\mathtt{n}} u : \mathscr{M}$, such that $(\Gamma_{t\{x\setminus u\}}; y : \mathscr{M}') = (\Gamma_p; y : \mathscr{M}') + \Gamma_u$, $b = b_p + b_u$, and $s = s_p + s_u$. Notice that $\Gamma_{t\{x\setminus u\}} = \Gamma_p + \Gamma_u$, since we may assume (by Proposition 7) that $y \notin \mathtt{dom}(\Gamma_u)$. Therefore, we can build Φ_t as follows:

$$\dfrac{\Phi_p}{\Gamma_p; x : \mathscr{M} \overset{(b_p,s_p)}{\vdash} \lambda y.p : \mathscr{M}' \to \tau'}\,(\lambda)$$

  Thus, we can conclude with $\Gamma_{t\{x\setminus u\}} = \Gamma_p + \Gamma_u$, $b = b_p + b_u$, and $s = s_p + s_u$.
- Rule (@). Then $t = rp$ and $t\{x \setminus u\} = (r\{x \setminus u\})(p\{x \setminus u\})$. Moreover, $\Phi_{t\{x\setminus u\}}$ has two premises $\Phi_{r\{x\setminus u\}} \rhd \Gamma_{r\{x\setminus u\}} \overset{(b_1,s_1)}{\vdash_\mathtt{n}} r\{x \setminus u\} : \mathscr{M}' \to \tau$ and $\Phi_{p\{x\setminus u\}} \rhd \Gamma_{p\{x\setminus u\}} \overset{(b_2,s_2)}{\vdash_\mathtt{n}} p\{x \setminus u\} : \mathscr{M}'$, such that $\Gamma_{t\{x\setminus u\}} = \Gamma_{r\{x\setminus u\}} + \Gamma_{p\{x\setminus u\}}$, $\mathscr{G} = \tau$, $b = 1 + b_1 + b_2$, and $s = s_1 + s_2$. By applying the *i.h.* to

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$\Phi_{r\{x\backslash u\}}$ and $\Phi_{p\{x\backslash u\}}$, there exist $\Phi_r \rhd \Gamma_r; x : \mathscr{M}_1 \overset{(b_r,s_r)}{\vdash_{\mathtt{n}}} r : \mathscr{M}' \to \tau$ and $\Phi_u^1 \rhd \Gamma_u^1 \overset{(b_u^1,s_u^1)}{\vdash_{\mathtt{n}}} u : \mathscr{M}_1$, such that $\Gamma_{r\{x\backslash u\}} = \Gamma_r + \Gamma_u^1$, $b_1 = b_r + b_u^1$, and $s_1 = s_r + s_u^1$, and there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \overset{(b_p,s_p)}{\vdash_{\mathtt{n}}} p : \mathscr{M}'$ and $\Phi_u^2 \rhd \Gamma_u^2 \overset{(b_u^2,s_u^2)}{\vdash_{\mathtt{n}}} u : \mathscr{M}_2$, such that $\Gamma_{p\{x\backslash u\}} = \Gamma_p + \Gamma_u^2$, $b_2 = b_p + b_u^2$, and $s_2 = s_p + s_u^2$, respectively. By Lemma 23, there exists $\Phi_u \rhd \Gamma_u^1 + \Gamma_u^2 \overset{(b_u^1+b_u^2,s_u^1+s_u^2)}{\vdash_{\mathtt{n}}} u : \mathscr{M}_1 \sqcup \mathscr{M}_2$, such that $\Gamma_u = \Gamma_u^1 + \Gamma_u^2$, $b_u = b_u^1 + b_u^2$, and $s_u = s_u^1 + s_u^2$. Therefore, we can build Φ_t as follows:

$$\frac{\Phi_r \qquad \Phi_p}{(\Gamma_r + \Gamma_p); x : \mathscr{M}_1 \sqcup \mathscr{M}_2 \overset{(1+b_u+b_p,s_u+s_p)}{\vdash} up : \tau}(@)$$

Thus, we can conclude with $\Gamma_{t\{x\backslash u\}} = \Gamma_{r\{x\backslash u\}} + \Gamma_{p\{x\backslash u\}} = \Gamma_t + \Gamma_u$, $\mathscr{M} = \mathscr{M}_1 \sqcup \mathscr{M}_2$, $b = 1 + b_1 + b_2 = 1 + b_r + b_u^1 + b_p + b_u^2 = b_t + b_v$, and $s = s_1 + s_2 = s_r + s_u^1 + s_p + s_u^2 = s_t + s_u$.

- Rule ($\mathtt{m}$). Then, $\Phi_{t\{x\backslash u\}}$ has premises $(\Phi_i \rhd \Gamma_i \overset{(b_i,s_i)}{\vdash_{\mathtt{n}}} t\{x \setminus u\} : \tau_i)_{i\in I}$, such that $\mathscr{G} = [\tau_i]_{i\in I}$, $\Gamma_{t\{x\backslash u\}} = +_{i\in I}\Gamma_i$, $b = +_{i\in I} b_i$, and $s = +_{i\in I} s_i$. By applying the *i.h.* to each Φ_i, there exist $\Phi_t^i \rhd \Gamma_t^i; x : \mathscr{M}_i \overset{(b_t^i,s_t^i)}{\vdash_{\mathtt{n}}} t : \tau_i$ and $\Phi_u^i \rhd \Gamma_u^i \overset{(b_u^i,s_u^i)}{\vdash_{\mathtt{n}}} u : \mathscr{M}_i$, such that $\Gamma_i = \Gamma_t^i + \Gamma_u^i$, $b_i = b_t^i + b_u^i$, and $s_i = s_t^i + s_u^i$, for each $i \in I$. By Lemma 23, there exists $\Phi_u \rhd +_{i\in I}\Gamma_u^i \overset{(+_{i\in I}b_u^i,+_{i\in I}s_u^i)}{\vdash_{\mathtt{n}}} u : \mathscr{M}$, such that $\Gamma_u = +_{i\in I}\Gamma_u^i$, $b_u = +_{i\in I} b_u^i$, and $s_u = +_{i\in I} s_u^i$. Moreover, we can build Φ_t as follows:

$$\frac{(\Phi_t^i \rhd \Gamma_t^i; x : \mathscr{M}_i \overset{(b_t^i,s_t^i)}{\vdash_{\mathtt{n}}} t : \tau_i)_{i\in I}}{+_{i\in I}\Gamma_t^i; x : \sqcup_{i\in I}\mathscr{M}_i \overset{(+_{i\in I}b_t^i,+_{i\in I}s_t^i)}{\vdash} t : [\tau_i]_{i\in I}}(\mathtt{m})$$

such that $\Gamma_t = +_{i\in I}\Gamma_t^i$, $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b_t = +_{i\in I} b_t^i$, and $s_t = +_{i\in I} s_t^i$. And we can conclude with $\Gamma_{t\{x\backslash u\}} = +_{i\in I}\Gamma_i = +_{i\in I}(\Gamma_t^i + \Gamma_u^i) = +_{i\in I}\Gamma_t^i +_{i\in I} \Gamma_u^i = \Gamma_t + \Gamma_u$, $b = +_{i\in I} b_i = +_{i\in I}(b_t^i + b_u^i) = +_{i\in I} b_t^i +_{i\in I} b_u^i = b_t + b_u$, and $s = +_{i\in I} s_i = +_{i\in I}(s_t^i + s_u^i) = +_{i\in I} s_t^i +_{i\in I} s_u^i = s_t + s_u$.
- Rule ($\lambda_{\mathtt{p}}$). Then, $t = \lambda y.p$ and $t\{x \setminus u\} = \lambda y.(p\{x \setminus u\})$. Moreover, $\Gamma_{t\{x\backslash u\}} = \emptyset$, $\tau = \mathtt{a}$, and $b = m = 0$. Therefore, we can build $\Phi_u \rhd \emptyset \overset{(0,0)}{\vdash_{\mathtt{n}}} u : [\,]$, such that $\Gamma_u = \emptyset$, $\mathscr{G} = \mathtt{a}$, $\mathscr{M} = [\,]$, and $b_u = s_u = 0$, by using rule ($\mathtt{m}$) with no premises, and build $\Phi_t \rhd \emptyset \overset{(0,0)}{\vdash_{\mathtt{n}}} \lambda y.p : \mathtt{a}$, such that $\Gamma_t = \emptyset$, and $b_t = s_t = 0$, by using ($\lambda_{\mathtt{p}}$). Thus, we can conclude with $\Gamma_{t\{x\backslash u\}} = \emptyset = \Gamma_t + \Gamma_u = \emptyset$, $b = b_t + b_u = 0$, and $s = s_t + s_u = 0$.
- Rule ($@_{\mathtt{p}}$). Then, $t = rp$ and $t\{x \setminus u\} = (r\{x \setminus u\})(p\{x \setminus u\})$. Moreover, $\Phi_{t\{x\backslash u\}}$ has one premise $\Phi_{r\{x\backslash u\}} \rhd \Gamma_{t\{x\backslash u\}} \overset{(b,s')}{\vdash_{\mathtt{n}}} r\{x \setminus u\} : \mathtt{n}$, such that $s = 1 + s'$, and $\mathscr{G} = \mathtt{n}$. By applying the *i.h.* to $\Phi_{r\{x\backslash u\}}$, there exist $\Phi_r \rhd \Gamma_r; \mathscr{M} \overset{(b_r,s_r)}{\vdash_{\mathtt{n}}} r : \mathtt{n}$ and $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathscr{M}$, such that $\Gamma_{t\{x\backslash u\}} = \Gamma_r + \Gamma_u$, $b = b_r + b_u$, and $s' = s_r + s_u$. Therefore, we can build Φ_t as follows:

$$\frac{\Phi_r}{\Gamma_r; x : \mathscr{M} \overset{(b_r,1+s_r)}{\vdash} rp : \mathtt{n}}(@_{\mathtt{p}})$$

such that $\Gamma_t = \Gamma_r$, $b_t = b_r$, and $s_t = 1 + s_r$. Thus, we can conclude with $s = 1 + s' = 1 + s_r + s_u = s_t + s_u$.

□

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 5 (Quantitative subject expansion). *Let* $\Phi_{t'} \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t' : \tau$ *be a tight type derivation. If* $t \to_{\mathtt{n}} t'$, *then there exists* $\Phi_t \rhd \Gamma \overset{(b+1,s)}{\vdash_{\mathtt{n}}} t : \tau$.

Proof. Similarly to Lemma 4, we prove a stronger version of the statement that allows us to reason inductively:

Let $\Phi_{t'} \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} t' : \tau$, such that Γ is tight, and either $\tau \in \mathtt{tt}_{\mathtt{n}}$ or $\neg\mathtt{abs}(t)$. If $t \to t'$, then there exists $\Phi_t \rhd \Gamma \overset{(b+1,s)}{\vdash_{\mathtt{n}}} t : \tau$.

The proof now follows by induction over $t \to t'$:

- Case $t = (\lambda x.p)u \to p\{x \setminus u\} = t'$. Then $\Phi_{t'} \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{n}}} p\{x \setminus u\} : \tau$. By Lemma 25, there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M} \overset{(b_p,s_p)}{\vdash_{\mathtt{n}}} p : \tau$ and $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathscr{M}$, such that $\Gamma = \Gamma_p + \Gamma_u$ tight, $\tau \in \mathtt{tt}_{\mathtt{n}}$, $b = b_p + b_u$, and $s = s_p + s_u$. Therefore, we can build Φ_t as follows:

$$\dfrac{\dfrac{\Phi_p}{\Gamma_p \overset{(b_p,s_p)}{\vdash} \lambda x.p : \mathscr{M} \to \tau}(\lambda) \qquad \Phi_u}{\Gamma_p + \Gamma_u \overset{(1+b_p+b_u,s_p+s_u)}{\vdash} (\lambda x.p)u : \tau}(@)$$

  Thus, we can conclude with $b + 1 = 1 + b_p + b_u$.
- Case $t = up \to u'p = t'$, such that $u \to u'$. Then, $\Phi_{t'}$ must end with rule $(@)$ or $(@_{\mathtt{p}})$:
  - Case $\Phi_{t'}$ ends with rule $(@)$. Then, $\Phi_{t'}$ has two premises, $\Phi_{u'} \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{n}}} u' : \mathscr{M}' \to \tau$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{n}}} p : \mathscr{M}'$, such that $\Gamma = \Gamma_u + \Gamma_p$ tight, $\tau \in \mathtt{tt}_{\mathtt{n}}$, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Notice that $\neg\mathtt{abs}(u)$, since $u \to u'$. Moreover, Γ_u and Γ_p are tight. Therefore, by applying the *i.h.* to $\Phi_{u'}$, there exists $\Phi_u \rhd \Gamma_u \overset{(1+b_u,s_u)}{\vdash_{\mathtt{n}}} u : \mathscr{M}' \to \tau$. Therefore, we can build Φ_t as follows:

$$\dfrac{\Phi_u \qquad \Phi_p}{\Gamma_u + \Gamma_p \overset{(1+1+b_u+b_p,s_u+s_p)}{\vdash} up : \tau}(@)$$

    Thus, we can conclude with $b + 1 = (1 + b_u + b_p) + 1$.
  - Case $\Phi_{t'}$ ends with rule $(@_{\mathtt{p}})$. Then $\Phi_{t'}$ has one premise $\Phi_{u'} \rhd \Gamma \overset{(b,s_{u'})}{\vdash_{\mathtt{n}}} u' : \mathtt{n}$, such that $s = 1 + s_{u'}$. By applying the *i.h.* to $\Phi_{u'}$, there exists $\Phi_u \rhd \Gamma \overset{(b+1,s_{u'})}{\vdash_{\mathtt{n}}} u : \mathtt{n}$. Therefore, we can build Φ_t as follows:

$$\dfrac{\Phi_u}{\Gamma \overset{(b+1,1+s_{u'})}{\vdash} up : \mathtt{n}}(@_{\mathtt{p}})$$

□

Theorem 2 (Quantitative completeness). *If* $t \twoheadrightarrow^{b}_{\mathtt{n}} u$ *with* $u \in \mathtt{no}_{\mathtt{n}}$, *then there exists a tight type derivation* $\Phi_t \rhd \Gamma \overset{(b,|u|_{\mathtt{n}})}{\vdash_{\mathtt{n}}} t : \tau$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Proof. The proof follows by induction over b:

- Case $b = 0$. Then $t = u$, which means that $t \in \mathtt{no}_\mathtt{n}$. Therefore, we can conclude by Lemma 3.
- Case $b > 0$. Then there exists t', such that $t \to t' \twoheadrightarrow_\mathtt{n}^{b-1} u$. By the *i.h.*, there exists a tight derivation $\Phi_{t'} \triangleright \Gamma \overset{(b-1,|u|_\mathtt{n})}{\vdash_\mathtt{n}} t' : \tau$. By Lemma 5, there exists a tight derivation $\Phi \triangleright \Gamma \overset{(b,|u|_\mathtt{n})}{\vdash_\mathtt{n}} t : \tau$. So, we can conclude. □

A.1.2 Weak Open CBV

Proposition 2 (Normal forms for Weak Open CBV). *Let t be a term. Then $t \in \mathtt{no}_\mathtt{v}$ iff $t \not\to_\mathtt{v}$.*

Proof. We refine the original statement into the two following ones:

(1) $t \not\to_\mathtt{v}$ and $\neg\mathtt{val}(t)$ iff $t \in \mathtt{ne}_\mathtt{v}$.
(2) $t \not\to_\mathtt{v}$ iff $t \in \mathtt{no}_\mathtt{v}$.

The proof now follows by simultaneous induction over both these statements:

⇒) By induction over t:

(1) Let $t \not\to_\mathtt{v}$ and $\neg\mathtt{val}(t)$. We want to show that $t \in \mathtt{ne}_\mathtt{v}$:
   - Case $t = x$ or $t = \lambda x.u$. Then $\mathtt{val}(t)$, which contradicts the hypothesis. Therefore, this cannot apply.
   - Case $t = up$. Since $up \not\to_\mathtt{v}$, no rule of $\to_\mathtt{v}$ applies to up. We consider two cases:
     * $\mathtt{abs}(u)$, then $\neg\mathtt{val}(p)$ otherwise rule $(\beta_\mathtt{v})$ would apply. Moreover, $p \not\to_\mathtt{v}$, otherwise t would be reducible by (appR). Therefore, $p \in \mathtt{ne}_\mathtt{v}$ by the *i.h.* (1). Furthermore, $\mathtt{val}(u)$ by definition, that is $u \in \mathtt{no}_\mathtt{v}$ and therefore $up \in \mathtt{ne}_\mathtt{v}$.
     * $\neg\mathtt{abs}(u)$, then $\neg\mathtt{val}(u)$ or $u = x$ (since variables are the only values other than abstractions). We can also conclude that $u \not\to_\mathtt{v}$ otherwise t would be reducible by (appL). And $p \not\to_\mathtt{v}$, otherwise t would be reducible by (appR), therefore the *i.h.* (2), we get $p \in \mathtt{no}_\mathtt{v}$. If $\neg\mathtt{val}(u)$, then we can apply the *i.h.* (1), and get $u \in \mathtt{ne}_\mathtt{v}$. In both cases, $up \in \mathtt{ne}_\mathtt{v}$.

(2) Let $t \not\to_\mathtt{v}$. We want to show that $t \in \mathtt{no}_\mathtt{v}$:
   - Case $\mathtt{val}(t)$. Then, $t \in \mathtt{no}_\mathtt{v}$ by definition.
   - Case $\neg\mathtt{val}(t)$. Then, $t \in \mathtt{ne}_\mathtt{v}$ by point (1), which implies $t \in \mathtt{no}_\mathtt{v}$.

⇐) By mutual induction over $t \in \mathtt{ne}_\mathtt{v}$ and $t \in \mathtt{no}_\mathtt{v}$:

(1) Let $t \in \mathtt{ne}_\mathtt{v}$. We want to show that $t \not\to_\mathtt{v}$ and $\neg\mathtt{val}(t)$:
   - Case $t = xp$, such that $p \in \mathtt{no}_\mathtt{v}$. Then, $p \not\to_\mathtt{v}$ by *i.h.* (2). Therefore, no rule of $\to_\mathtt{v}$ applies to xp. Moreover, $x\neg\mathtt{val}(p)$.
   - Case $t = up$, such that $u \in \mathtt{no}_\mathtt{v}$ and $p \in \mathtt{ne}_\mathtt{v}$. Then, $u \not\to_\mathtt{v}$, by *i.h.* (2), and $p \not\to_\mathtt{v}$ and $\neg\mathtt{val}(p)$, by *i.h.* (1). Therefore, no rule of $\to_\mathtt{v}$ applies to up. Moreover, $u\neg\mathtt{val}(p)$.
   - Case $t = up$, such that $u \in \mathtt{ne}_\mathtt{v}$ and $p \in \mathtt{no}_\mathtt{v}$. Then, $u \not\to_\mathtt{v}$ and $\neg\mathtt{val}(u)$, by *i.h.* (1), and $p \not\to_\mathtt{v}$, by *i.h.* (2). Therefore, no rule of $\to_\mathtt{v}$ applies to up. Moreover, $\neg\mathtt{val}(up)$.

(2) Let $t \in \mathtt{no}_\mathtt{v}$. We want to show that $t \not\to_\mathtt{v}$:
   - Case $\mathtt{val}(t)$. Then, $t \not\to_\mathtt{v}$, by definition.
   - Case $\neg\mathtt{val}(t)$. Then, $t \in \mathtt{ne}_\mathtt{v}$ by definition, and thus, $t \not\to_\mathtt{v}$ by *i.h.* (1). □

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 26 (Values are not neutral). *Let* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$. *If* $\mathtt{val}(t)$, *then* $\tau \neq \mathtt{n}$.

Proof. By case analysis on $\mathtt{val}(t)$:

- Case $t = x$. Then, Φ must end with rule (ax) or (m). In the former case, $\tau = \sigma$, and $\mathtt{n} \notin \sigma$. In the latter case, $\tau = \mathscr{M}$, and $\mathscr{M} \neq \mathtt{n}$. Thus, we can conclude that $\tau \neq \mathtt{n}$.
- Case $t = \lambda x.t$. Then, Φ must end with rule (λ), (m), or ($\lambda_{\mathtt{p}}$). In the first case, $\tau = \mathscr{M} \to \tau'$, and $\mathscr{M} \to \tau' \neq \mathtt{n}$. In the second case, $\tau = \mathscr{M}$, and $\mathscr{M} \neq \mathtt{n}$. In the last case, $\tau = \mathtt{a}$, and $\mathtt{a} \neq \mathtt{n}$. □

Proposition 8 (Relevance). *Let* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$. *Then* $\mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(t)$.

Proof. By a simple induction over Φ. □

Lemma 27 (Persistent abstractions have tight type). *If* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$. *If* $\tau \in \overline{\mathtt{a}}$, *then* $\neg\mathtt{abs}(t)$.

Proof. By induction over Φ, according to the last rule of Φ:

- Rule (ax), (@), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$). Then, $\neg\mathtt{abs}(t)$, by definition.
- Rule (λ), (m), or ($\lambda_{\mathtt{p}}$). Then, $\tau \notin \{\mathtt{v}, \mathtt{n}\} = \overline{\mathtt{a}}$. Therefore, these cases hold vacuously. □

Lemma 28 (Tight spreading). *Let* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$, *such that* Γ *is tight. If* $t \in \mathtt{ne}_{\mathtt{v}}$, *then* τ *is a tight constant.*

Proof. By induction over $t \in \mathtt{ne}_{\mathtt{v}}$:

- Case $t = xp$, such that $p \in \mathtt{no}_{\mathtt{v}}$. Then, Φ must end with rule (@), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
  - Rule (@). Then, Φ has two premises $\Phi_x \rhd x : [\mathscr{M} \to \tau] \overset{(0,0)}{\vdash_{\mathtt{v}}} x : \mathscr{M} \to \delta$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}$, such that $\Gamma = (x : [\mathscr{M} \to \tau]) + \Gamma_p$, $b = 1 + b_p$ and $s = s_p$. However, $\mathscr{M} \to \tau$ is not tight, which contradicts the tightness of Γ. Therefore, this case cannot apply.
  - Rule ($@^1_{\mathtt{p}}$) or ($@^2_{\mathtt{p}}$). Then, $\tau = \mathtt{n}$.
- Case $t = up$, such that $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}}$. Then, Φ must end with rule (@), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
  - Rule (@). Then, Φ has two premises $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M} \to \tau$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}$, such that $\Gamma = \Gamma_u + \Gamma_p$, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Therefore, Γ_u and Γ_p are tight. By applying the *i.h.* to Φ_p we get that $\mathscr{M} \in \mathtt{tt}_{\mathtt{v}}$, which is a contradiction. Therefore, this case cannot apply.
  - Rule ($@^1_{\mathtt{p}}$) or ($@^2_{\mathtt{p}}$). Then, $\tau = \mathtt{n}$.
- Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{v}}$ and $p \in \mathtt{no}_{\mathtt{v}}$. Then, Φ must end with rule (@), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
  - Rule (@). Then, Φ has two premises $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M} \to \tau$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}$, such that $\Gamma = \Gamma_u + \Gamma_p$, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Therefore, Γ_u and Γ_p are tight. By applying the *i.h.* to Φ_p, we get that $\mathscr{M} \in \mathtt{tt}_{\mathtt{v}}$, which is a contradiction. Therefore, this case cannot apply.
  - Rule ($@^1_{\mathtt{p}}$) or ($@^2_{\mathtt{p}}$). Then, $\tau = \mathtt{n}$. □

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 1 (Normal forms take zero steps). *Let* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$ *be a tight type derivation. Then,* $b = 0$ *iff* $t \in \mathtt{no}_{\mathtt{v}}$.

Proof.

$\Rightarrow$) We refine the original statement into the two following ones:

(1) Let $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$ be a tight derivation. Then $b = 0$ and $\neg\mathtt{val}(t)$ implies $t \in \mathtt{ne}_{\mathtt{v}}$.

(2) Let $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$ be a tight derivation. Then $b = 0$ implies $t \in \mathtt{no}_{\mathtt{v}}$.

The proof follows by mutual induction over Φ for both statements, according to the last rule of Φ:

(1) Let $b = 0$ and $\neg\mathtt{val}(t)$:

- Rule $(\mathtt{ax})$, (λ), $(\mathtt{m})$, or $(\lambda_{\mathtt{p}})$. Then, $\mathtt{val}(t)$, which contradiction the hypotheses. Therefore, these cases cannot apply.
- Rule $(@)$. Then, $b > 0$. Therefore, this case holds vacuously.
- Rule $(@^1_{\mathtt{p}})$. Then, $t = up$, and Φ has two premises $\Phi_u \rhd \Gamma_u \overset{(0,s_u)}{\vdash_{\mathtt{v}}} u : \overline{\mathtt{a}}$ and $\Phi_p \rhd \Gamma_p \overset{(0,s_p)}{\vdash_{\mathtt{v}}} p : \mathtt{tt}_{\mathtt{v}}$, such that $\Gamma = \Gamma_u + \Gamma_p$, $\tau = \mathtt{n}$, and $s = 1 + s_u + s_p$. Therefore, both Γ_u and Γ_p are tight. By applying the *i.h.* (2) to Φ_u and Φ_p, we have $u, p \in \mathtt{no}_{\mathtt{v}}$. Since $\tau = \mathtt{n} \in \overline{\mathtt{a}}$, then, by Lemma 27, we know that $\neg\mathtt{abs}(u)$. Therefore, either $u = x$ or $u \in \mathtt{ne}_{\mathtt{v}}$, by definition. In both cases, $up \in \mathtt{ne}_{\mathtt{v}}$.
- Rule $(@^2_{\mathtt{p}})$. Then, $t = up$, and Φ has two premises $\Phi_u \rhd \Gamma_u \overset{(0,s_u)}{\vdash_{\mathtt{v}}} u : \mathtt{tt}_{\mathtt{v}}$ and $\Phi_p \rhd \Gamma_p \overset{(0,s_p)}{\vdash_{\mathtt{v}}} p : \mathtt{n}$, such that $\Gamma = \Gamma_u + \Gamma_p$, $\tau = \mathtt{n}$, and $s = 1 + s_u + s_p$. Therefore, Γ_u and Γ_p are tight. By applying the *i.h.* (2) to Φ_u, we have that $u \in \mathtt{no}_{\mathtt{v}}$. Since $\tau = \mathtt{n}$, then by Lemma 26, we know that $\neg\mathtt{val}(t)$. By applying the *i.h.* (1) to Φ_p, we have that $p \in \mathtt{ne}_{\mathtt{v}}$. In which case, $up \in \mathtt{ne}_{\mathtt{v}}$.

(2) Let $b = 0$:

- Rule $(\mathtt{ax})$, (λ), $(\mathtt{m})$, or $(\lambda_{\mathtt{p}})$. Then, $\mathtt{val}(t) \subseteq \mathtt{no}_{\mathtt{v}}$, by definition.
- Rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@^2_{\mathtt{p}})$. Then $\neg\mathtt{val}(t)$ by definition. Therefore, $t \in \mathtt{ne}_{\mathtt{v}} \subseteq \mathtt{no}_{\mathtt{v}}$ by *i.h.* (1).

$\Leftarrow$) The proof follows by mutual induction over $t \in \mathtt{no}_{\mathtt{v}}$:

(1) Let $t \in \mathtt{ne}_{\mathtt{v}}$:

- Case $t = xp$, such that $p \in \mathtt{no}_{\mathtt{v}}$. Then, Φ must end with rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@_2)$:
  - Case Φ ends with $(@)$. Then, Φ has necessarily two premises $\Phi_x \rhd x : [\mathscr{M} \to \tau] \overset{(0,0)}{\vdash_{\mathtt{v}}} x : \mathscr{M} \to \tau$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}$, such that $\Gamma = (x : [\mathscr{M} \to \tau]) + \Gamma_p$, $b = 1 + b_p$, and $s = s_p$. However, Γ is not tight, because $\mathscr{M} \to \tau \notin \mathtt{tt}_{\mathtt{v}}$, which contradicts the hypothesis. Therefore, this case cannot apply.
  - Case Φ ends with $(@^1_{\mathtt{p}})$. Then, Φ has two premises $\Phi_x \rhd (x : [\mathtt{v}]) \overset{(0,0)}{\vdash_{\mathtt{v}}} x : \mathtt{v}$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathtt{tt}_{\mathtt{v}}$, such that $\Gamma = (x : [\mathtt{v}]) + \Gamma_p$, $\tau = \mathtt{n}$, $b = b_p$, and $s = 1 + s_p$. Therefore, Γ_p is tight. By applying the *i.h.* to Φ_p, we have that $b_p = 0$, and thus, $b = b_p = 0$. We then conclude.
  - Case Φ ends with $(@^2_{\mathtt{p}})$. Very similar to the previous case.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

– Case $t = up$, such that $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}}$. Then, Φ must end with rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@_2)$:

∗ Case Φ ends with $(@)$. Then, Φ has two premises $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M} \to \tau$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}$, such that $\Gamma = \Gamma_u + \Gamma_p$, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Therefore, Γ_u and Γ_p are tight. We can then apply Lemma 28 to Φ_u, and we get $\mathscr{M} \to \tau$ tight, which yields to a contradiction. Therefore, this case cannot apply.

∗ Case Φ ends with $(@^1_{\mathtt{p}})$ or $(@^2_{\mathtt{p}})$. Very similar to the cases for $t = xp$, such that $p \in \mathtt{no}_{\mathtt{v}}$.

– Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{v}}$ and $p \in \mathtt{no}_{\mathtt{v}}$. This case is very similar to the case where $t = up$, such that $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}}$.

(2) Let $t \in \mathtt{no}_{\mathtt{v}}$:

– Case $\mathtt{val}(t)$. Then, Φ must end with rule $(\mathtt{ax})$, (λ), $(\mathtt{m})$, or $(\lambda_{\mathtt{p}})$. Case Φ ends with rules (λ) or $(\mathtt{m})$ then, $\tau \notin \mathtt{tt}_{\mathtt{v}}$ and the statement hold vacuously. Otherwise, $b = 0$, by definition.

– Case $\neg\mathtt{val}(t)$. Then, $t \in \mathtt{ne}_{\mathtt{v}}$, by definition. Therefore, $b = 0$, by *i.h.* (1). □

Lemma 2 (Normal forms have correct size). *Let* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$ *be a tight type derivation. If* $b = 0$ *then* $s = |t|_{\mathtt{v}}$.

Proof. The proof follows by induction over Φ, according to the last rule of Φ:

- Rule $(\mathtt{ax})$ or $(\lambda_{\mathtt{p}})$. Then, $\mathtt{val}(t)$ and $s = 0 = |t|_{\mathtt{v}}$.
- Rule (λ) or $(\mathtt{m})$. Then, $\tau \notin \mathtt{tt}_{\mathtt{v}}$. Therefore, these cases holds vacuously.
- Rule $(@)$. Then, $b > 0$. Therefore, this case holds vacuously.
- Rule $(@^1_{\mathtt{p}})$. Then, $t = up$ and Φ has two premises $\Phi_u \rhd \Gamma_u \overset{(0,s_u)}{\vdash_{\mathtt{v}}} u : \overline{\mathtt{a}}$ and $\Phi_p \rhd \Gamma_p \overset{(0,s_p)}{\vdash_{\mathtt{v}}} p : \mathtt{tt}_{\mathtt{v}}$, such that $\Gamma = \Gamma_u + \Gamma_p$, $\tau = \mathtt{n}$, $b = 0$, and $s = 1 + s_u + s_p$. Therefore, in particular, Γ_u and Γ_p are tight. Since Φ_u and Φ_p are both tight, then, by applying the *i.h.* to Φ_u and Φ_p, we have $s_u = |u|_{\mathtt{v}}$ and $s_p = |p|_{\mathtt{v}}$. Therefore, $s = 1 + |u|_{\mathtt{v}} + |p|_{\mathtt{v}} = |up|_{\mathtt{v}}$.
- Rule $(@^2_{\mathtt{p}})$. Very similar to the previous case. □

Lemma 3 (All normal forms are tightly typable). *If* $t \in \mathtt{no}_{\mathtt{v}}$, *then there exists a tight type derivation* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$, *such that* $s = |t|_{\mathtt{v}}$.

Proof. We are going to split the original statement into the following two:

(1) If $t \in \mathtt{ne}_{\mathtt{v}}$, then there exists a tight derivation $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \mathtt{n}$, such that $s = |t|_{\mathtt{v}}$.

(2) If $t \in \mathtt{no}_{\mathtt{v}}$, then there exists a tight derivation $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \mathtt{tt}_{\mathtt{v}}$, such that $s = |t|_{\mathtt{v}}$.

The proof follows by mutual induction over $t \in \mathtt{no}_{\mathtt{v}}$:

(1) Let $t \in \mathtt{ne}_{\mathtt{v}}$:

- Case $t = xp$, such that $p \in \mathtt{no}_{\mathtt{v}}$. By applying the *i.h.* (2) to p, we have a tight derivation $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathtt{tt}_{\mathtt{v}}$, such that $|p|_{\mathtt{v}} = s_p$. Therefore, we can build Φ as follows:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\dfrac{\dfrac{}{x : [\mathtt{v}] \overset{(0,0)}{\vdash} x : \mathtt{v}}\,(\mathtt{ax}) \qquad \Phi_p}{x : [\mathtt{v}] + \Gamma_p \overset{(b_p,1+s_p)}{\vdash} xp : \mathtt{n}}\,(@^1_{\mathtt{p}})$$

where $\Gamma = (x : [\mathtt{v}]) + \Gamma_p$, $b = b_p$, and $s = 1 + s_p = 1 + 0 + |p|_{\mathtt{v}} = |xp|_{\mathtt{v}}$.

- Case $t = up$, such that $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}}$. By applying the *i.h.* (2) to u and *i.h.* (1) to p, there exists a tight derivation $\Phi_u \triangleright \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathtt{tt}_{\mathtt{v}}$, such that $|u|_{\mathtt{v}} = s_u$, and a tight derivation $\Phi_p \triangleright \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathtt{n}$, such that $|p|_{\mathtt{v}} = s_p$. Therefore, we can build Φ as follows:

$$\dfrac{\Phi_u \qquad \Phi_p}{\Gamma_u + \Gamma_p \overset{(b_u+b_p,1+s_u+s_p)}{\vdash} up : \mathtt{n}}\,(@^2_{\mathtt{p}})$$

where $\Gamma = \Gamma_u + \Gamma_p$, $b = b_u + b_p$, and $s = 1 + s_u + s_p = 1 + |u|_{\mathtt{v}} + |p|_{\mathtt{v}} = |up|_{\mathtt{v}}$, by definition.

- Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{v}}$ and $p \in \mathtt{no}_{\mathtt{v}}$. By the *i.h.* (1) to u and *i.h.* (2) to p, there exists a tight derivation $\Phi_u \triangleright \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathtt{n}$ such that $|u|_{\mathtt{v}} = s_u$, and a tight derivation $\Phi_p \triangleright \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathtt{tt}_{\mathtt{v}}$ such that $|p|_{\mathtt{v}} = s_p$. Therefore, we can build Φ as follows:

$$\dfrac{\Phi_u \qquad \Phi_p}{\Gamma_u + \Gamma_p \overset{(b_u+b_p,1+s_u+s_p)}{\vdash} up : \mathtt{n}}\,(@^1_{\mathtt{p}})$$

where $\Gamma = \Gamma_u + \Gamma_p$, $b = b_u + b_p$, and $s = 1 + s_u + s_p = 1 + |u|_{\mathtt{v}} + |p|_{\mathtt{v}} = |up|_{\mathtt{v}}$, by definition.

(2) Let $t \in \mathtt{no}_{\mathtt{v}}$:

- Case $t = x$. Then, we can build Φ as follows:

$$\dfrac{}{x : [\sigma] \overset{(0,0)}{\vdash} x : \sigma}\,(\mathtt{ax})$$

where $\sigma \in \overline{\mathtt{n}}$, $\Gamma = \emptyset$, $b = 0$, and $s = 0 = |x|_{\mathtt{v}}$.

- Case $t = \lambda x.u$. Then we can build Φ as follows:

$$\dfrac{}{\emptyset \overset{(0,0)}{\vdash} \lambda x.u : \mathtt{a}}\,(\lambda_{\mathtt{p}})$$

where $\Gamma = \emptyset$, $b = 0$, and $s = 0 = |\lambda x.u|_{\mathtt{v}}$.

- Case $\neg\mathtt{val}(t)$. Then, $t \in \mathtt{ne}_{\mathtt{v}}$, by definition. Therefore, there exists a tight derivation $\Phi_t \triangleright \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \mathtt{n}$ such that $s = |t|_{\mathtt{v}}$, by the *i.h.* (1). □

Lemma 29 (Multi-types can split and merge). *Let* $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b = +_{i\in I} b_i$, *and* $s = +_{i\in I} s_i$. *Then,* $\Phi_t \triangleright \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} v : \mathscr{M}$ *iff* $(\Phi^i_t \triangleright \Gamma_i \overset{(b_i,s_i)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_i)_{i\in I}$.

Proof. Very similar to the proof for Weak Open CBN. □

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 30 (Substitution). *Let* $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \overset{(b_t,s_t)}{\vdash_{\mathtt{v}}} t : \tau$ *and* $\Phi_v \rhd \Gamma_v \overset{(b_v,s_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}$, *then there exists* $\Phi_{t\{x\setminus v\}} \rhd \Gamma_t + \Gamma_v \overset{(b_t+b_v,s_t+s_v)}{\vdash_{\mathtt{v}}} t\{x \setminus v\} : \tau$.

Proof. The proof follows by induction over Φ_t, depending on the last rule of Φ_t:

- Rule (ax). Then, $t = y$. There are two cases:
  - $y = x$. Then, $\Gamma_t = \emptyset$, $\tau = \mathscr{M}$, $t\{x \setminus v\} = v$, $b_t = 0$, and $s_t = 0$. So we can take $\Phi_{t\{x\setminus v\}} = \Phi_v$ and conclude with $\Gamma_t + \Gamma_v = \Gamma_v$, $b_t + b_v = b_v$, and $s_t + s_v = s_v$.
  - $y \neq x$. Then $\mathscr{M} = [\,]$, $\Gamma_v = \emptyset$, $t\{x \setminus v\} = t$, $b_v = 0$, and $s_v = 0$. So we can take $\Phi_{t\{x\setminus v\}} = \Phi_t$ and conclude with $\Gamma_t + \Gamma_v = \Gamma_t$, $b_t + b_v = b_t$, and $s_t + s_v = s_t$.
- Rule (λ). Then, $t = \lambda y.u$ and Φ_t has one premise $\Phi_u \rhd \Gamma_u; x : \mathscr{M} \overset{(b_t,s_t)}{\vdash_{\mathtt{v}}} u : \tau'$, such that $\Gamma_t = (\Gamma_u \setminus\!\setminus y)$ (by α-conversion), and $\tau = \Gamma_u(y) \to \tau'$. By applying the *i.h.* to Φ_u, there exists $\Phi_{u\{x\setminus v\}} \rhd \Gamma_u + \Gamma_v \overset{(b_t+b_v,s_t+s_v)}{\vdash_{\mathtt{v}}} u\{x \setminus v\} : \tau$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{\Phi_{u\{x\setminus v\}}}{(\Gamma_u + \Gamma_v) \setminus\!\setminus y \overset{(b_t+b_v,s_t+s_v)}{\vdash} (\lambda y.u)\{x \setminus v\} : \Gamma_u(y) \to \tau'}(\lambda)$$

  And we can conclude with $(\Gamma_u + \Gamma_v) \setminus\!\setminus y = (\Gamma_u \setminus\!\setminus y) + \Gamma_v = \Gamma_t + \Gamma_v$, since we may assume (by α-conversion) that $y \notin \mathtt{dom}(\Gamma_v)$.
- Rule (@). Then, $t = up$ and Φ_t has two premises $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_1 \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M}' \to \tau$ and $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}'$, such that $\Gamma_t = (\Gamma_u + \Gamma_p)$, $\mathscr{M} = \mathscr{M}_1 \sqcup \mathscr{M}_2$, $b_t = 1 + b_u + b_p$, and $s_t = s_u + s_p$. By Lemma 29, there exist $(\Phi^i_v \rhd \Gamma^i_v \overset{(b_i,s_i)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_i)_{i\in\{1,2\}}$, such that $\Gamma_v = \Gamma^1_v + \Gamma^2_v$, $b_v = b_1 + b_2$, and $s_v = s_1 + s_2$. By applying the *i.h.* to Φ_u and Φ_p, we know there exist $\Phi_{u\{x\setminus v\}} \rhd \Gamma + \Gamma^1_v \overset{(b_u+b_1,s_u+s_1)}{\vdash_{\mathtt{v}}} u\{x \setminus v\} : \mathscr{M}' \to \tau$ and $\Phi_{p\{x\setminus v\}} \rhd \Delta + \Gamma^2_v \overset{(b_p+b_2,s_p+s_2)}{\vdash_{\mathtt{v}}} p\{x \setminus v\} : \mathscr{M}'$, respectively. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{\Phi_{u\{x\setminus v\}} \qquad \Phi_{p\{x\setminus v\}}}{(\Gamma + \Delta) + (\Gamma^1_v + \Gamma^2_v) \overset{(1+b_u+b_p+b_1+b_2,s_u+s_p+s_1+s_2)}{\vdash} (up)\{x \setminus v\} : \tau}(@)$$

  And we can conclude with $(\Gamma + \Delta) + (\Gamma^1_v + \Gamma^2_v) = \Gamma_t + \Gamma_v$, $1 + b_u + b_p + b_1 + b_2 = b_t + b_v$, and $s_u + s_p + s_1 + s_2 = s_t + s_v$.
- Rule (m). Then, $t = w$ and Φ has premises $(\Phi^i_w \rhd \Gamma_i; x : \mathscr{M}_i \overset{(b_i,s_i)}{\vdash_{\mathtt{v}}} w : \sigma_i)_{i\in I}$, such that $\tau = [\sigma_i]_{i\in I}$, $\Gamma_t = +_{i\in I}\Gamma_i$, $b_t = +_{i\in I}b_i$, and $s_t = +_{i\in I}s_i$. By Lemma 29, there exists $(\Phi^i_v \rhd \Gamma^i_v \overset{(b^i_v,s^i_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_i)_{i\in I}$, such that $\Gamma_v = +_{i\in I}\Gamma^i_v$, $b_v = +_{i\in I}b^i_v$, and $s_v = +_{i\in I}s^i_v$. By applying the *i.h.* to each Φ^i_w, there exist $(\Phi^i_{w\{x\setminus v\}} \rhd \Gamma_i + \Gamma^i_v \overset{(b_i+b^i_v,s_i+s^i_v)}{\vdash_{\mathtt{v}}} w\{x \setminus v\} : \sigma_i)_{i\in I}$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{(\Phi^i_{w\{x\setminus v\}})_{i\in I}}{+_{i\in I}(\Gamma_i + \Gamma^i_v) \overset{(+_{i\in I}(b_i+b^i_v),+_{i\in I}(s_i+s^i_v))}{\vdash} w\{x \setminus v\} : [\sigma_i]_{i\in I}}(\mathtt{m})$$

  And we can conclude with $+_{i\in I}(\Gamma_i + \Gamma^i_v) = +_{i\in I}\Gamma_i +_{i\in I} \Gamma^i_v = \Gamma_t + \Gamma_v$, $+_{i\in I}(b_i + b^i_v) = +_{i\in I}b_i +_{i\in I} b^i_v = b_t + b_v$, and $+_{i\in I}(s_i + s^i_v) = +_{i\in I}s_i +_{i\in I} s^i_v = s_t + s_v$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Rule $(\lambda_{\mathtt{p}})$. Then $t = \lambda y.u$, and $\Phi_t \triangleright \overset{(0,0)}{\vdash_{\mathtt{v}}} \lambda y.u : \mathtt{a}$, such that $\Gamma_t = \emptyset$, $\tau = \mathtt{a}$, $b_t = 0$ and $s_t = 0$. Therefore, $\mathscr{M} = [\,]$, $\Gamma_v = \emptyset$, $b_v = 0$, and $s_v = 0$. Also, notice that $t\{x \setminus v\} = \lambda y.(u\{x \setminus v\}) = (\lambda y.u)\{x \setminus v\}$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{}{\emptyset \overset{(0,0)}{\vdash} (\lambda y.u)\{x \setminus v\} : \mathtt{a}}\,(\lambda_{\mathtt{p}})$$

  And we can conclude with $\emptyset = \Gamma_t + \Gamma_v$, $0 = b_t + b_v$, and $0 = s_t + s_v$.
- Case Φ_t ends with rule $(@^1_{\mathtt{p}})$ or $(@^2_{\mathtt{p}})$. Very similar to the case where Φ_t ends with rule $(@)$. □

Lemma 4 (Quantitative subject reduction). *Let* $\Phi_{t'} \triangleright \Gamma \overset{(b-1,s)}{\vdash_{\mathtt{v}}} t' : \tau$ *be a tight type derivation. If* $t \to t'$, *then there exists* $\Phi_{t'} \triangleright \Gamma \overset{(b-1,s)}{\vdash_{\mathtt{v}}} t' : \tau$.

Proof. We prove a stronger version of the statement that allows us to reason inductively:

Let $\Phi_t \triangleright \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$, such that Γ is tight, and either τ is tight or $\neg\mathtt{val}(t)$. If $t \to_{\mathtt{v}} t'$, then there exists $\Phi_{t'} \triangleright \Gamma \overset{(b-1,s)}{\vdash_{\mathtt{v}}} t' : \tau$.

The proof follows by induction over $t \to_{\mathtt{v}} t'$:

- Case $t = (\lambda x.u)v \to_{\mathtt{v}} u\{x \setminus v\} = t'$. Then, Φ_t must end with rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@^2_{\mathtt{p}})$. Let us first assume that Φ_t ends with rule $(@^1_{\mathtt{p}})$. Then, Φ_t has two premises $\Phi_{\lambda x.u}$ and Φ_v for $\lambda x.u$ and v, respectively. Moreover, $\Phi_{\lambda x.u}$ must conclude with type $\overline{\mathtt{a}}$. However, this is not possible according to Lemma 27. Now, let us assume that Φ_t ends with rule $(@^2_{\mathtt{p}})$. Again, Φ_t has two premises $\Phi_{\lambda x.u}$ and Φ_v for $\lambda x.u$ and v, respectively. Moreover, Φ_v must conclude with type $\mathtt{n}$. However, this is not possible by Lemma 26. Finally, let us assume that Φ_t ends with rule $(@)$. Then, Φ_t has the following form:

$$\dfrac{\dfrac{\Phi_u \triangleright \Gamma_u; x : \mathscr{M} \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \tau}{\Gamma_u \overset{(b_u,s_u)}{\vdash} (\lambda x.u) : \mathscr{M} \to \tau}\,(\lambda) \qquad \Phi_v \triangleright \Gamma_v \overset{(b_v,s_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}}{\Gamma_u + \Gamma_v \overset{(1+b_u+b_v,s_u+s_v)}{\vdash} (\lambda x.u)v : \tau}\,(@)$$

  such that $\Gamma = \Gamma_u + \Gamma_v$ is tight, $\tau \in \mathtt{tt}_{\mathtt{v}}$, $b = 1 + b_u + b_v$, and $s = s_u + s_v$. By Lemma 30, there exists $\Phi_{u\{x\setminus v\}} \triangleright \Gamma_u + \Gamma_v \overset{(b_u+b_v,s_u+s_v)}{\vdash_{\mathtt{v}}} u\{x \setminus v\} : \tau$. Therefore, we can take $\Phi_{t'} = \Phi_{u\{x\setminus v\}}$ and conclude with $b - 1 = b_u + b_v$.
- Case $t = up \to_{\mathtt{v}} u'p = t'$, such that $u \to_{\mathtt{v}} u'$. Then Φ_t must either end with $(@)$, $(@^1_{\mathtt{p}})$, or $(@^2_{\mathtt{p}})$:
  - Case Φ_t ends with rule $(@)$. Then, Φ_t has the following form:

$$\frac{\Phi_u \triangleright \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M} \to \tau \qquad \Phi_p \triangleright \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}}{\Gamma_u + \Gamma_p \overset{(1+b_u+b_p,s_u+s_p)}{\vdash} up : \tau}\,(@)$$

    such that $\Gamma = \Gamma_u + \Gamma_p$ is tight, $\tau \in \mathtt{tt}_{\mathtt{v}}$, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Moreover, Γ_u and Γ_p are tight. Also, notice that $\neg\mathtt{val}(u)$, since $u \to_{\mathtt{v}} u'$ holds. Therefore, by applying the *i.h.*

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

to Φ_u, there exists $\Phi_{u'} \rhd \Gamma_u \overset{(b_u-1,s_u)}{\vdash_{\mathtt{v}}} u' : \mathscr{M} \to \tau$. We can then build $\Phi_{t'}$ as follows:

$$\frac{\Phi_{u'} \qquad \Phi_p}{\Gamma_u + \Gamma_p \overset{(b_u+b_p,s_u+s_p)}{\vdash} u'p : \tau}(@)$$

Thus, we can conclude with $b - 1 = b_u + b_p$.

– Case Φ_t ends with rule $(@^1_{\mathtt{p}})$ or $(@^2_{\mathtt{p}})$. These cases are very similar to the previous case.

- Case $t = up \to_{\mathtt{v}} up' = t'$, such that $u \not\to_{\mathtt{v}}$ and $p \to_{\mathtt{v}} p'$. Then, Φ_t must end with rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@^2_{\mathtt{p}})$:

  – Case Φ_t ends with rule $(@)$. Then, Φ_t has the following form:

$$\frac{\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M} \to \tau \qquad \Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}}{\Gamma_u + \Gamma_p \overset{(1+b_u+b_p,s_u+s_p)}{\vdash} up : \tau}(@)$$

such that $\Gamma = \Gamma_u + \Gamma_p$ is tight, $\tau \in \mathtt{tt}_{\mathtt{v}}$, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Moreover, Γ_u and Γ_p are tight. Also, notice that $\neg\mathtt{val}(p)$, since $p \to_{\mathtt{v}} p'$ holds. Therefore, by applying the *i.h.* to Φ_p, there exists $\Phi_{p'} \rhd \Gamma_p \overset{(b_p-1,s_p)}{\vdash_{\mathtt{v}}} p' : \mathscr{M}$. Therefore, we can build $\Phi_{t'}$ as follows:

$$\frac{\Phi_u \qquad \Phi_{p'}}{\Gamma_u + \Gamma_p \overset{(b_u+b_p,s_u+s_p)}{\vdash} up' : \tau}(@)$$

Thus, we can conclude with $b - 1 = b_u + b_p$.

– Case Φ_t ends with rule $(@^1_{\mathtt{p}})$ or $(@^2_{\mathtt{p}})$. These cases are very similar to the previous case. □

Theorem 1 (Quantitative soundness). *If* $\Phi \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t : \tau$ *is a tight type derivation, then there exists* $u \in \mathtt{no}$ *such that* $t \twoheadrightarrow^b_{\mathtt{v}} u$ *with* $|u|_{\mathtt{v}} = s$.

Proof. The proof follows by induction on b:

- Case $b = 0$. Then $t \in \mathtt{no}_{\mathtt{v}}$ by Lemma 1. Moreover, $s = |t|_{\mathtt{v}}$ by Lemma 2. So we can conclude with $u = t$.
- Case $b > 0$. Then $t \notin \mathtt{no}_{\mathtt{v}}$ by Lemma 1. Therefore, by Proposition 2, there exists t' such that $t \to t'$. Moreover, this derivation is also tight. By Lemma 4, there exists $\Phi_{t'} \rhd \Gamma \overset{(b-1,s)}{\vdash_{\mathtt{v}}} t' : \tau$. By the *i.h.*, there exists $u \in \mathtt{no}$, such that $t' \twoheadrightarrow^{b-1}_{\mathtt{v}} u$ and $s = |u|_{\mathtt{v}}$. We can then conclude with $t \to t' \twoheadrightarrow^{b-1}_{\mathtt{v}} u$, which means that $t \twoheadrightarrow^b_{\mathtt{v}} u$, as expected. □

Lemma 31 (Anti-substitution). *Let* $\Phi_{t\{x\setminus v\}} \rhd \Gamma_{t\{x\setminus v\}} \overset{(b,s)}{\vdash_{\mathtt{v}}} t\{x \setminus v\} : \tau$. *Then, there exist* $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \overset{(b_t,s_t)}{\vdash_{\mathtt{v}}} t : \tau$ *and* $\Phi_v \rhd \Gamma_v \overset{(b_v,s_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}$, *such that* $\Gamma_{t\{x\setminus v\}} = \Gamma_t + \Gamma_v$, $b = b_t + b_v$, *and* $s = s_t + s_v$.

Proof. The proof follows by induction over $\Phi_{t\{x\setminus v\}}$:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Let us assume $t = x$. Then, $t\{x \setminus v\} = v$. Therefore, $\tau \neq \mathtt{n}$, by Lemma 26. Now, we have to consider two cases: either $\tau = [\sigma_i]_{i\in I}$ or τ is not a multi-type. If $\tau = [\sigma_i]_{i\in I}$, we can take $\Phi_v = \Phi_{t\{x\setminus v\}}$ and $\mathscr{M} = [\sigma_i]_{i\in I}$, such that $\Gamma_v = \Gamma_{t\{x\setminus v\}}$, $b_v = b$, $s_v = s$, and build Φ_t as follows:

$$\dfrac{\left(\dfrac{}{x : [\sigma_i] \overset{(0,0)}{\vdash_{\mathtt{v}}} x : \sigma_i}\,(\mathtt{ax})\right)_{i\in I}}{x : [\sigma_i]_{i\in I} \overset{(0,0)}{\vdash_{\mathtt{v}}} x : [\sigma_i]_{i\in I}}\,(\mathtt{m})$$

such that $\Gamma_t = \emptyset$, and $b_t = s_t = 0$. If τ is not a multi-type, then we let $\mathscr{M} = [\tau]$, and we build $\Phi_v \rhd \Gamma_{t\{x\setminus v\}} \overset{(b,s)}{\vdash_{\mathtt{v}}} v : [\tau]$, such that $\Gamma_v = \Gamma_{t\{x\setminus v\}}$, $b_v = b$, and $s_v = s$, from $\Phi_{t\{x\setminus v\}}$ followed by rule (m), and $\Phi_t \rhd x : [\tau] \overset{(0,0)}{\vdash_{\mathtt{v}}} x : \tau$, such that $\Gamma_t = \emptyset$, $b_t = s_t = 0$, by using rule (ax). In both cases, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_v = \Gamma_t + \Gamma_v$, $b = b_v = b_t + b_v$, and $s = s_v = s_t + s_v$. From now on, we are going to assume $t \neq x$:

- Rule (ax). Then, $t = y$, $t\{x \setminus v\} = y$, and $\mathscr{M} = [\,]$. Therefore, we can pick $\Phi_t = \Phi_{t\{x\setminus v\}}$, such that $\Gamma_t = \Gamma_{t\{x\setminus v\}}$, $b_t = b$, and $s_t = s$, and build $\Phi_v \rhd \emptyset \overset{(0,0)}{\vdash_{\mathtt{v}}} v : [\,]$, such that $\Gamma_v = \emptyset$ and $b_v = s_v = 0$, by using rule (m) with no premises. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_t = \Gamma_t + \Gamma_v$, and $b = b_t = b_t + b_v$, and $s = s_t = s_t + s_v$.
- Rule (λ). Then, $t = \lambda y.u$ and $t\{x \setminus v\} = \lambda y.(u\{x \setminus v\})$. Therefore, $\Phi_{t\{x\setminus v\}}$ has one premise $\Phi_{u\{x\setminus v\}} \rhd \Gamma_{t\{x\setminus v\}};\ y : \mathscr{M}' \overset{(b,s)}{\vdash_{\mathtt{v}}} u\{x \setminus v\} : \tau'$, such that $\tau = \mathscr{M}' \to \tau'$. By applying the *i.h.* to $\Phi_{u\{x\setminus v\}}$, there exist $\Phi_u \rhd (\Gamma_u; y : \mathscr{M}'); x : \mathscr{M} \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \tau'$ and $\Phi_v \rhd \Gamma_v \overset{(b_v,s_v)}{\vdash_{\mathtt{v}}} : \mathscr{M}$, such that $(\Gamma_{t\{x\setminus v\}};\ y : \mathscr{M}') = (\Gamma_u; y : \mathscr{M}') + \Gamma_v$, $b = b_u + b_v$, and $s = s_u + s_v$. Notice that $\Gamma_{t\{x\setminus v\}} = \Gamma_u + \Gamma_v$, since we may assume (by Proposition 8), that $y \notin \mathtt{dom}(\Gamma_v)$. Therefore, we can build Φ_t as follows:

$$\dfrac{\Phi_u}{\Gamma_u; x : \mathscr{M} \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} \lambda y.u : \mathscr{M}' \to \tau'}\,(\lambda)$$

  such that $\Gamma_t = \Gamma_u$, $b_t = b_u$, and $s_t = s_u$. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_u + \Gamma_v$, $b = b_u + b_v$, and $s = s_u + s_v$.
- Rule (@). Then, $t = up$ and $t\{x \setminus v\} = (u\{x \setminus v\})(p\{x \setminus v\})$. Moreover, $\Phi_{t\{x\setminus v\}}$ has two premises $\Phi_{u\{x\setminus v\}} \rhd \Gamma_{u\{x\setminus v\}} \overset{(b_1,s_1)}{\vdash_{\mathtt{v}}} u\{x \setminus v\} : \mathscr{M}' \to \tau$ and $\Phi_{p\{x\setminus v\}} \rhd \Gamma_{p\{x\setminus v\}} \overset{(s_2,s_2)}{\vdash_{\mathtt{v}}} p\{x \setminus v\} : \mathscr{M}'$, such that $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} + \Gamma_{p\{x\setminus v\}}$, $b = 1 + b_1 + b_2$, and $s = s_1 + s_2$. By applying the *i.h.* to $\Phi_{u\{x\setminus v\}}$ and $\Phi_{p\{x\setminus v\}}$, there exist $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_1 \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M}' \to \tau$ and $\Phi^1_v \rhd \Gamma^1_v \overset{(b^1_v,s^1_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_1$, such that $\Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma^1_v$, $b_1 = b_u + b^1_v$, and $s_1 = s_u + s^1_v$, and there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}'$ and $\Phi^2_v \rhd \Gamma^2_v \overset{(b^2_v,s^2_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_2$, such that $\Gamma_{p\{x\setminus v\}} = \Gamma_p + \Gamma^2_v$, $b_2 = b_p + b^2_v$, and $s_2 = s_p + s^2_v$, respectively. By Lemma 29, there exists $\Phi_v \rhd \Gamma^1_v + \Gamma^2_v \overset{(b^1_v+b^2_v,s^1_v+s^2_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_1 \sqcup \mathscr{M}_2$, such that $\Gamma_v = \Gamma^1_v + \Gamma^2_v$, $b_v = b^1_v + b^2_v$, and $s_v = s^1_v + s^2_v$. Therefore, we can build Φ_t as follows:

$$\dfrac{\Phi_u \qquad \Phi_p}{(\Gamma_u + \Gamma_p); x : \mathscr{M}_1 \sqcup \mathscr{M}_2 \overset{(1+b_u+b_p,s_u+s_p)}{\vdash_{\mathtt{v}}} up : \tau}\,(@)$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

such that $\Gamma_t = \Gamma_u + \Gamma_p$, $b_t = 1 + b_u + b_p$, and $s_t = s_u + s_p$. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} + \Gamma_{p\{x\setminus v\}} = (\Gamma_u + \Gamma_v^1) + (\Gamma_p + \Gamma_v^2) = \Gamma_t + \Gamma_v$, $b = 1 + b_1 + b_2 = 1 + b_u + b_v^1 + b_p + b_v^2 = b_t + b_v$, and $s = s_1 + s_2 = s_u + s_v^1 + s_p + s_v^2 = s_t + s_v$.

- Rule ($\mathtt{m}$). Then, $\Phi_{t\{x\setminus v\}}$ has premises $(\Phi_i \rhd \Gamma_i \overset{(b_i,s_i)}{\vdash_{\mathtt{v}}} t\{x\setminus v\} : \sigma_i)_{i\in I}$, such that $\tau = [\sigma_i]_{i\in I}$, $\Gamma_{t\{x\setminus v\}} = +_{i\in I}\Gamma_i$, $b = +_{i\in I} b_i$, and $s = +_{i\in I} s_i$. By applying the *i.h.* to each Φ_i, there exist $\Phi_t^i \rhd \Gamma_t^i; x : \mathscr{M}_i \overset{(b_t^i,s_t^i)}{\vdash_{\mathtt{v}}} t : \sigma_i$ and $\Phi_v^i \rhd \Gamma_v^i \overset{(b_v^i,s_v^i)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_i$, such that $\Gamma_i = \Gamma_t^i + \Gamma_v^i$, $b_i = b_t^i + b_v^i$, and $s_i = s_t^i + s_v^i$, for each $i \in I$. By Lemma 29, there exists $\Phi_v \rhd +_{i\in I}\Gamma_v^i \overset{(+_{i\in I} b_v^i, +_{i\in I} s_v^i)}{\vdash_{\mathtt{v}}} v : \mathscr{M}$, such that $\Gamma_v = +_{i\in I}\Gamma_v^i$, $b_v = +_{i\in I} b_v^i$, and $s_v = +_{i\in I} s_v^i$. Moreover, we can build Φ_t as follows:

$$\frac{(\Phi_t^i \rhd \Gamma_t^i; x : \mathscr{M}_i \overset{(b_t^i,s_t^i)}{\vdash_{\mathtt{v}}} t : \sigma_i)_{i\in I}}{+_{i\in I}\Gamma_t^i; x : \sqcup_{i\in I}\mathscr{M}_i \overset{(+_{i\in I} b_t^i, +_{i\in I} s_t^i)}{\vdash_{\mathtt{v}}} t : [\sigma_i]_{i\in I}}(\mathtt{m})$$

such that $\Gamma_t = +_{i\in I}\Gamma_t^i$, $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b_t = +_{i\in I} b_t^i$, and $s_t = +_{i\in I} s_t^i$. And we can conclude with $\Gamma_{t\{x\setminus v\}} = +_{i\in I}\Gamma_i = +_{i\in I}(\Gamma_t^i + \Gamma_v^i) = +_{i\in I}\Gamma_t^i + _{i\in I}\ \Gamma_v^i = \Gamma_t + \Gamma_v$, $b = +_{i\in I} b_i = +_{i\in I}(b_t^i + b_v^i) = +_{i\in I} b_t^i + _{i\in I}\ b_v^i = b_t + b_v$, and $s = +_{i\in I} s_i = +_{i\in I}(s_t^i + s_v^i) = +_{i\in I} s_t^i + _{i\in I}\ s_v^i = s_t + s_v$.
- Rule ($\lambda_{\mathtt{p}}$). Then, $t = \lambda y.u$ and $t\{x\setminus v\} = \lambda y.(u\{x\setminus v\})$. Moreover, $\Gamma_{t\{x\setminus v\}} = \emptyset$, $\tau = \mathtt{a}$, and $b = m = d = 0$. Therefore, we can build $\Phi_v \rhd \emptyset \overset{(0,0)}{\vdash_{\mathtt{v}}} v : [\,]$, such that $\Gamma_v = \emptyset$, $\mathscr{M} = [\,]$, $b_v = s_v = 0$, by using rule ($\mathtt{m}$) with no premises, and build $\Phi_t \rhd \emptyset \overset{(0,0)}{\vdash_{\mathtt{v}}} \lambda y.u : \mathtt{a}$, such that $\Gamma_t = \emptyset$, and $b_t = s_t = 0$, by using rule ($\lambda_{\mathtt{p}}$). Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_t + \Gamma_v = \emptyset$, $b = b_t + b_v = 0$, and $s = s_t + s_v = 0$.
- Rule ($@_{\mathtt{p}}^1$). Then, $t = up$ and $t\{x\setminus v\} = (u\{x\setminus v\})(p\{x\setminus v\})$. Moreover, $\Phi_{t\{x\setminus v\}}$ has two premises $\Phi_{u\{x\setminus v\}} \rhd \Gamma_{u\{x\setminus v\}} \overset{(b_1,s_1)}{\vdash_{\mathtt{v}}} u\{x\setminus v\} : \overline{\mathtt{a}}$ and $\Phi_{p\{x\setminus v\}} \rhd \Gamma_{p\{x\setminus v\}} \overset{(b_2,s_2)}{\vdash_{\mathtt{v}}} p\{x\setminus v\} : \mathtt{tt}_{\mathtt{v}}$, such that $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} + \Gamma_{p\{x\setminus v\}}$, $b = b_1 + b_2$, and $s = 1 + s_1 + s_2$. By applying the *i.h.* to $\Phi_{u\{x\setminus v\}}$ and $\Phi_{p\{x\setminus v\}}$, there exist $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_1 \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \not\mathtt{a}$ and $\Phi_v^1 \rhd \Gamma_v^1 \overset{(b_v^1,s_v^1)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_1$, such that $\Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma_v^1$, $b_1 = b_u + b_v^1$, and $s_1 = s_u + s_v^1$, and there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathtt{tt}_{\mathtt{v}}$ and $\Phi_v^2 \rhd \Gamma_v^2 \overset{(b_v^2,s_v^2)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_2$, such that $\Gamma_{p\{x\setminus v\}} = \Gamma_p + \Gamma_v^2$, $b_1 = b_p + b_v^2$, and $s_2 = s_p + s_v^2$, respectively. By Lemma 29, there exists $\Phi_v \rhd \Gamma_v^1 + \Gamma_v^2 \overset{(b_v^1+b_v^2, s_v^1+s_v^2)}{\vdash_{\mathtt{v}}} v : \mathscr{M}_1 \sqcup \mathscr{M}_2$, such that $\Gamma_v = \Gamma_v^1 + \Gamma_v^2$, $b_v = b_v^1 + b_v^2$, and $s_v = s_v^1 + s_v^2$. Therefore, we can build Φ_t as follows:

$$\frac{\Phi_u \qquad \Phi_p}{(\Gamma_u + \Gamma_p); x : \mathscr{M}_1 \sqcup \mathscr{M}_2 \overset{(b_u+b_p,\,1+s_u+s_p)}{\vdash_{\mathtt{v}}} up : \tau}(@)$$

such that $\Gamma_t = \Gamma_u + \Gamma_p$, $b_t = b_u + b_p$, and $s_t = 1 + s_u + s_p$. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} + \Gamma_{p\{x\setminus v\}} = (\Gamma_u + \Gamma_v^1) + (\Gamma_p + \Gamma_v^2) = \Gamma_t + \Gamma_v$, $b = b_1 + b_2 = b_u + b_v^1 + b_p + b_v^2 = b_t + b_v$, and $s = 1 + s_1 + s_2 = 1 + s_u + s_v^1 + s_p + s_v^2 = s_t + s_v$.
- Rule ($@_{\mathtt{p}}^2$). This case is very similar to the previous case. □

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 5 (Quantitative subject expansion). *Let* $\Phi_{t'} \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t' : \tau$ *be a tight type derivation. If* $t \to_{\mathtt{v}} t'$, *then there exists* $\Phi_t \rhd \Gamma \overset{(b+1,s)}{\vdash_{\mathtt{v}}} t : \tau$.

Proof. Just like for the subject reduction property, we prove a stronger version of the statement that allows us to reason inductively:

Let $\Phi_{t'} \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} t' : \tau$, such that Γ is tight, and either $\tau \in \mathtt{tt}_{\mathtt{v}}$ or $\neg\mathtt{val}(t)$. If $t \to_{\mathtt{v}} t'$, then there exists $\Phi_t \rhd \Gamma \overset{(b+1,s)}{\vdash_{\mathtt{v}}} t : \tau$.

The proof now follows by induction over $t \to_{\mathtt{v}} t'$:

- Case $t = (\lambda x.u)v \to_{\mathtt{v}} u\{x \setminus v\} = t'$. Then, $\Phi_{t'} \rhd \Gamma \overset{(b,s)}{\vdash_{\mathtt{v}}} u\{x \setminus v\} : \tau$. By Lemma 31, there exist $\Phi_u \rhd \Gamma_u; x : \mathscr{M} \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \tau$ and $\Phi_v \rhd \Gamma_v \overset{(b_v,s_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}$, such that $\Gamma = \Gamma_u + \Gamma_v$ tight, $\tau \in \mathtt{tt}_{\mathtt{v}}$, $b = b_u + b_v$, and $s = s_u + s_v$. Therefore, we can build Φ_t as follows:

$$\dfrac{\dfrac{\Phi_u \rhd \Gamma_u; x : \mathscr{M} \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \tau}{\Gamma_u \overset{(b_u,s_u)}{\vdash} \lambda x.u : \mathscr{M} \to \tau}(\lambda) \qquad \Phi_v \rhd \Gamma_v \overset{(b_v,s_v)}{\vdash_{\mathtt{v}}} v : \mathscr{M}}{\Gamma_u + \Gamma_v \overset{(1+b_u+b_v,s_u+s_v)}{\vdash} (\lambda x.u)v : \tau}(@)$$

  Thus, we can conclude with $b + 1 = 1 + b_u + b_v$.
- Case $t = up \to_{\mathtt{v}} u'p = t'$, such that $u \to_{\mathtt{v}} u'$. Then, $\Phi_{t'}$ must end with rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@^2_{\mathtt{p}})$:
  - Case $\Phi_{t'}$ ends with rule $(@)$. Then, $\Phi_{t'}$ has two premises, $\Phi_{u'} \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u' : \mathscr{M}' \to \tau$ and $\Phi_p \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}'$, such that $\Gamma = \Gamma_u + \Gamma_p$ is tight, $\tau \in \mathtt{tt}_{\mathtt{v}}$, $b = 1 + b_u + b_p$, and $s = s_u + s_p$. Notice that $\neg\mathtt{val}(u)$ holds, since $u \to_{\mathtt{v}} u'$. Moreover, Γ_u and Γ_p are tight. Therefore, by applying the *i.h.* to $\Phi_{u'}$, there exists $\Phi_u \rhd \Gamma_u \overset{(1+b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M}' \to \tau$. Therefore, we can build $\Phi_{t'}$ as follows:

$$\dfrac{\Phi_u \qquad \Phi_p}{\Gamma_u + \Gamma_p \overset{(1+1+b_u+b_p,s_u+s_p)}{\vdash} up : \tau}(@)$$

    Thus, we can conclude with $b + 1 = (1 + b_u + b_p) + 1$.
  - Case $\Phi_{t'}$ ends with rule $(@^1_{\mathtt{p}})$ or $(@^2_{\mathtt{p}})$. These cases are very similar to the previous one.
- Case $t = up \to_{\mathtt{v}} up' = t'$, such that $u \not\to_{\mathtt{v}}$ and $p \to_{\mathtt{v}} p'$. Then, $\Phi_{t'}$ must end with rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@^2_{\mathtt{p}})$:
  - Case $\Phi_{t'}$ ends with rule (@). Then, $\Phi_{t'}$ has two premises $\Phi_u \rhd \Gamma_u \overset{(b_u,s_u)}{\vdash_{\mathtt{v}}} u : \mathscr{M}' \to \tau$ and $\Phi_{p'} \rhd \Gamma_p \overset{(b_p,s_p)}{\vdash_{\mathtt{v}}} p' : \mathscr{M}'$, such that $\Gamma = \Gamma_u + \Gamma_p$ is tight, $\tau \in \mathtt{tt}_{\mathtt{v}}$, $b = 1 + b_u + b_p$, $s_t = s_u + s_p$. Notice that $\neg\mathtt{val}(p)$ holds, since $p \to_{\mathtt{v}} p'$. Moreover, Γ_u and Γ_p are tight. Therefore, by applying the *i.h.* to Φ_p, there exists $\Phi_p \rhd \Gamma_p \overset{(b_p+1,s_p)}{\vdash_{\mathtt{v}}} p : \mathscr{M}' \to \tau$. Therefore, we can build Φ_t as follows:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\frac{\Phi_u \qquad \Phi_p}{\Gamma_u + \Gamma_p \overset{(1+b_u+b_p+1, s_u+s_p)}{\vdash} up : \tau}\,(@)$$

Thus, we can conclude with $b+1 = (1+b_u+b_p)+1 = 1+b_u+b_p+1$.

– Case $\Phi_{t'}$ ends with rule $(\mathbb{Q}^1_{\mathrm{p}})$ or $(\mathbb{Q}^2_{\mathrm{p}})$. These cases are very similar to the previous one. □

Theorem 2 (Quantitative completeness). *If* $t \twoheadrightarrow^{b}_{\mathrm{v}} u$ *with* $u \in \mathtt{no}_{\mathrm{v}}$, *then there exists a tight type derivation* $\Phi_t \rhd \Gamma \overset{(b,|u|_{\mathrm{v}})}{\vdash_{\mathrm{v}}} t : \tau$.

Proof. The proof follows by induction on b:

- Case $b = 0$. Then $t = u$, which means that $t \in \mathtt{no}_{\mathrm{v}}$. Therefore, we conclude by Lemma 3.
- Case $b > 0$. Then there exists t', such that $t \to t' \twoheadrightarrow^{b-1}_{\mathrm{v}} u$. By the *i.h.*, there exists a tight derivation $\Phi_{t'} \rhd \Gamma \overset{(b-1,|u|)}{\vdash_{\mathrm{v}}} t' : \tau$. By Lemma 5, there exists a tight derivation $\Phi \rhd \Gamma \overset{(b,|u|)}{\vdash_{\mathrm{v}}} t : \tau$, and this derivation is also tight. So, we can conclude. □

A.2 Weak Open CBN with Global State

Proposition 3. *Let* (t, s) *be a configuration. Then* (t, s) *is normal iff* $(t, s) \not\rightarrow_{\mathrm{n}}$.

Proof.

⇒) Let (t, s) be normal. Then, either (t, s) is blocked or $t \in \mathtt{no}_{\mathrm{n}}$:

– Case (t, s) is blocked. We reason by induction on (t, s):

∗ Case $t = \mathtt{get}_l(\lambda x.u)$, such that $l \notin \mathtt{labels}(s)$. Then, $(t, s) \not\rightarrow_{\mathrm{n}}$ by definition.
∗ Case $t = up$, such that (u, s) is blocked. Then, (u, s) is normal. By applying the *i.h.* to (u, s), we get $(u, s) \not\rightarrow_{\mathrm{n}}$. Therefore, $(up, s) \not\rightarrow_{\mathrm{n}}$ holds.

– Case $t \in \mathtt{no}_{\mathrm{n}}$. Then (t, s) is unblocked by definition. We reason by induction over $t \in \mathtt{no}_{\mathrm{n}}$:

∗ Case $t = \lambda x.p$. Then, $(\lambda x.p, s) \not\rightarrow_{\mathrm{n}}$ is straightforward.
∗ Case $\neg\mathtt{abs}(t)$. Then, $t \in \mathtt{ne}_{\mathrm{n}}$. There are two cases:

· Case $t = x$. Then, $(x, s) \not\rightarrow_{\mathrm{n}}$ by definition.
· Case $t = up$, such that $u \in \mathtt{ne}_{\mathrm{n}} \subseteq \mathtt{no}_{\mathrm{n}}$. Then, (u, s) is normal. By applying the *i.h.* to (u, s), we get $(u, s) \not\rightarrow_{\mathrm{n}}$. Therefore, $(up, s) \not\rightarrow_{\mathrm{n}}$ holds.

⇐) Let $(t, s) \not\rightarrow_{\mathrm{n}}$. We reason by induction on t:

– Case $t = x$. Then, $t \in \mathtt{ne}_{\mathrm{n}} \subseteq \mathtt{no}_{\mathrm{n}}$. Therefore, (t, s) is normal.
– Case $t = \lambda x.p$. Then $\lambda x.p \in \mathtt{no}_{\mathrm{n}}$. Therefore, (t, s) is normal.
– Case $t = up$. Since $(up, s) \not\rightarrow_{\mathrm{n}}$, then $(u, s) \not\rightarrow_{\mathrm{n}}$ and $\neg\mathtt{abs}(u)$, otherwise rules (appL) and (β_{n}) would be applicable, respectively. By applying the *i.h.* to (u, s), we get (u, s) is normal. Therefore, either (u, s) is blocked, or $u \in \mathtt{ne}_{\mathrm{n}}$ (since $u \in \mathtt{no}_{\mathrm{n}}$ and $\neg\mathtt{abs}(u)$). In the former case, (up, s) is blocked by definition. In the latter case, $up \in \mathtt{ne}_{\mathrm{n}} \subseteq \mathtt{no}_{\mathrm{n}}$. Therefore, (up, s) is normal.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case $t = \mathtt{get}_l(\lambda x.u)$. Since $(\mathtt{get}_l(\lambda x.u), s) \not\rightarrow_{\mathtt{n}}$, then $l \notin \mathtt{labels}(s)$. Therefore, $(\mathtt{get}_l(\lambda x.u), s)$ is blocked and thus normal.
- Case $t = \mathtt{set}_l(t, u)$. Then $(\mathtt{set}_l(t, u), s) \rightsquigarrow_{\mathtt{n}} (u, \mathtt{upd}_l(t, s))$, which leads to a contradiction with the hypothesis. Therefore, this case cannot apply.

□

Lemma 32 (Relevance). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} t : \mathscr{G}$ *(resp.* $\Pi \triangleright \Delta \Vdash_{\mathtt{n}}^{(b',m',d')} s : \mathscr{T}$*). Then* $\mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(t)$ *(resp.* $\mathtt{dom}(\Delta) \subseteq \mathtt{fv}(s)$*).*

Proof. By a simple induction over Φ. □

Lemma 33. *Let* $\Pi \triangleright \Delta \Vdash_{\mathtt{n}}^{(b,m,d)} s : \mathscr{T}$*. If* $l \in \mathtt{dom}(\mathscr{T})$*, then* $l \in \mathtt{labels}(s)$*.*

Proof. The proof is exactly the same as for Weak Open CBV. Just notice that, in Weak Open CBN with Global State, we can assign the empty multi-type to any term. □

Proposition 4 (Typed configurations are unblocked). *If* $\Phi \triangleright \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t, s) : \kappa$*, then* (t, s) *is unblocked.*

Proof. The proof follows by contraposition. We assume that (t, s) is blocked and show that (t, s) cannot be typable. We proceed by induction on (t, s) blocked:

- Case $t = \mathtt{get}_l(\lambda x.t)$, such that $l \notin \mathtt{labels}(s)$. Then, suppose that (t, s) is typable, so that the type derivation must be of the following form, where $\mathscr{L}' = \Gamma_u(x) \sqcup \mathscr{M} \cdot \mathscr{L}$:

$$\dfrac{\dfrac{\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \{l : \mathscr{M} \cdot \mathscr{L}; \mathscr{T}\} \Rightarrow \kappa}{\Gamma_u \setminus\!\setminus x \Vdash_{\mathtt{n}}^{(b_u,1+m_u,d_u)} \mathtt{get}_l(\lambda x.u) : \{l : \mathscr{L}'\}; \mathscr{T} \Rightarrow \kappa}\,(\mathtt{get}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{n}}^{(b_s,m_s,d_s)} s : \{l : \mathscr{L}'\}; \mathscr{T}}{(\Gamma_u \setminus\!\setminus x) + \Delta_s \Vdash_{\mathtt{n}}^{(b_u+b_s,1+m_u+m_s,d_u+d_s)} (\mathtt{get}_l(\lambda x.t), s) : \kappa}\,(\mathtt{conf})$$

  where $\Gamma = (\Gamma_u \setminus\!\setminus x) + \Delta_s$, $b = b_u + b_s$, $m = 1 + m_u + m_s$, and $d = d_u + d_s$. Thus, $l \in \mathtt{dom}(\{l : \Gamma_u(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T})$, and so, by Lemma 33, we get $l \in \mathtt{labels}(s)$, which leads to a contradiction. Therefore, (t, s) is not typable.
- Case $t = up$, such that (u, s) is blocked. Suppose that (t, s) is typable with derivation Φ. We have two cases:
  - Case Φ ends with rule (@). Then, Φ must be of the following form:

$$\dfrac{\begin{array}{c}\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}') \\ \dfrac{\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{M}}{\Gamma_u + \Gamma_p \Vdash_{\mathtt{n}}^{(1+b_u+b_p,m_u+m_p,d_u+d_p)} up : \mathscr{T} \Rightarrow \kappa}\,(@) \end{array} \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{n}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{(\Gamma_u + \Gamma_u) + \Delta_s \Vdash_{\mathtt{n}}^{(1+b_u+b_p+b_s,m_u+m_p+m_s,d_u+d_p+d_s)} (up, s) : \kappa}\,(\mathtt{conf})$$

    where $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$, $b = 1 + b_u + b_p + b_s$, $m = m_u + m_p + m_s$, and $d = d_u + d_p + d_s$. Thus, we can build a type derivation $\Phi_{(u,s)}$ as follows:

$$\dfrac{\Phi_u \qquad \Pi_s}{\Gamma_u + \Delta_s \Vdash^{(b_u+b_s,m_u+m_s,d_u+d_s)} (u, s) : (\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa) \times \mathscr{T}')}\,(\mathtt{conf})$$

    By applying the *i.h.* on (u, s) we get that (u, s) is not typable, which leads to a contradiction. Therefore, (t, s) is not typable.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

– Case Φ ends with rule $(@_{\mathtt{p}})$. Then, Φ must be of the following form:

$$\dfrac{\dfrac{\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}{\Gamma_u \Vdash^{(b_u,m_u,1+d_u)} up : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}\,(@_{\mathtt{p}}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{n}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{\Gamma_u + \Delta_s \Vdash^{(b_u+b_s,m_u+m_s,1+d_u+d_s)} (up, s) : (\mathtt{n} \times \mathscr{T}')}\,(\mathtt{conf})$$

where $\Gamma = \Gamma_u + \Delta_s$, $b = b_u + b_s$, $m = m_u + m_s$, and $d = 1 + d_u + d_s$. Therefore, we can build a type derivation $\Phi_{(u,s)}$ as follows:

$$\dfrac{\Phi_u \qquad \Pi_s}{\Gamma_u + \Delta_s \Vdash^{(b_u+b_s,m_u+m_s,d_u+d_s)} (u, s) : (\mathtt{n} \times \mathscr{T}')}(\mathtt{conf})$$

By applying the *i.h.* to (u, s), we get that (u, s) is not typable, which leads to a contradiction. Therefore, (t, s) is not typable. □

Lemma 34 (Tight spreading). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, *such that* Γ *is tight. If* $t \in \mathtt{ne}_{\mathtt{n}}$, *then* $\tau = \mathtt{n}$.

Proof. We proceed by induction over $t \in \mathtt{ne}_{\mathtt{n}}$:

- Case $t = x$. Then Φ must end with rule $(\mathtt{ax})$. Therefore, $\Phi \triangleright x : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}') \Vdash_{\mathtt{n}}^{(b,m,d)} x : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T})$, and thus, $\tau = \mathtt{n}$.
- Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{n}}$. Then, Φ must end with rule $(@)$ or $(@_{\mathtt{p}})$:
  - Rule $(@)$. Then, Φ has two premises $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow (\tau \times \mathscr{T}'))) \times \mathscr{T}'')$ and $\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{M}$, where $\Gamma = \Gamma_u + \Gamma_p$ is tight, $b = 1 + b_u + b_p$, $m = m_u + m_p$, and $d = d_u + d_p$. Moreover, Γ_u and Γ_p are tight. By applying the *i.h.* to $u \in \mathtt{ne}_{\mathtt{n}}$, we get that $(\mathscr{M} \to (\mathscr{T}'' \Rightarrow (\tau \times \mathscr{T}'))) = \mathtt{n}$, which is a contradiction. Therefore, this case cannot apply.
  - Rule $(@_{\mathtt{p}})$. Then, $\tau = \mathtt{n} \in \mathtt{tt}_{\mathtt{n}}$ by definition.

□

Lemma 6 (Normal forms have pure types). *Let* $\Phi \triangleright \Vdash_{\mathtt{n}}^{(b,m,d)} t : \Gamma\, \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, *such that* Γ *is tight. If* $t \in \mathtt{no}_{\mathtt{n}}$, *then* $\mathscr{T} = \mathscr{T}'$.

Proof. By induction on $t \in \mathtt{no}_{\mathtt{n}}$. There are two cases:

- Case $t = \lambda x.u$. Then, Φ must end with rule (λ), $(\uparrow)$, or $(\lambda_{\mathtt{p}})$:
  - Case Φ ends with rule (λ) or $(\lambda_{\mathtt{p}})$. Then, $\mathscr{T} = \mathscr{T}'$, by definition.
- Case $\neg\mathtt{abs}(t)$. Then $t \in \mathtt{ne}_{\mathtt{n}}$:
  - Case $t = x$. Since Γ is tight, then $\Phi \triangleright x : [\mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')] \Vdash_{\mathtt{n}}^{(b,m,d)} x : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, such that $\mathscr{T} = \mathscr{T}'$.
  - Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{n}}$. Then, Φ must end with rule $(@)$ or $(@_{\mathtt{p}})$:
    - Case Φ ends with $(@)$. Then, Φ has two premises $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}')$ and $\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{M}$, such that $\Gamma = \Gamma_u + \Gamma_p$, $b = 1 + b_u + b_p$, $m = m_u + m_p$, and $d = d_u + d_p$. By Lemma 34, $\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)$ should be tight, which is a contradiction. Therefore, this case cannot apply.
    - Case Φ ends with $(@_{\mathtt{p}})$. Then, Φ has one premise $\Phi_u \triangleright \Gamma \Vdash_{\mathtt{n}}^{(b,m,d_u)} u : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, such that $d = 1 + d_u$. By applying the *i.h.* to Φ_u, we get that $\mathscr{T} = \mathscr{T}'$.

□

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 7 (Normal forms and tight states have correct size).

(1) *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{n}}^{(0,0,d)} t : \mu$ *be a tight derivation. Then,* (*i*) $t \in \mathtt{no}_{\mathtt{n}}$ *and* (*ii*) $d = |t|_{\mathtt{n}}$.

(2) *Let* $\Pi \triangleright \Delta \Vdash_{\mathtt{n}}^{(b,m,d)} s : \mathscr{T}$ *be a tight derivation. Then* $b = m = d = 0$.

Proof.

(1) We split statement (1a) into two parts, but keep the same statement for (1b):

(a) If $\mu = \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, then $t \in \mathtt{ne}_{\mathtt{n}}$.
(b) If $\mu = \mathscr{T} \Rightarrow (\mathtt{tt}_{\mathtt{n}} \times \mathscr{T}')$, then $t \in \mathtt{no}_{\mathtt{n}}$.
(c) $d = |t|_{\mathtt{n}}$.

We reason by induction over Φ:

- Rule (ax). Then, $\Phi \triangleright x : [\mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T})] \Vdash_{\mathtt{n}}^{(0,0,0)} x : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T})$, where $\Gamma = (x : [\mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T})])$, $\mu = \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T})$, and $d = 0$. Thus, we can conclude with $x \in \mathtt{ne}_{\mathtt{n}}$ (satisfying 1a), and $d = 0 = |x|_{\mathtt{n}}$ (satisfying 1c).
- Rule (m). Then, the type of the conclusion is not a monadic type. Therefore, this case cannot apply.
- Rule (@). Then, the first counter must be greater than zero. Therefore, this case cannot apply.
- Rules (λ) or ($\lambda_{\mathtt{p}}$). Then, $t = \lambda x.u$ and either μ is not a tight monadic type, or $\mu = \mathscr{T} \Rightarrow (\mathtt{a} \times \mathscr{T})$ and $d = 0$, respectively. Thus, in the latter case, we can conclude with $\lambda x.u \in \mathtt{no}_{\mathtt{n}}$ (satisfying 1b) and $|\lambda x.u|_{\mathtt{n}} = 0 = d$ (satisfying 1c).
- Rule (get) or (set). Then, the second counter must be greater than zero. Therefore, this case does not apply.
- Rule ($@_{\mathtt{p}}$). Then, $t = up$, and Φ has one premise $\Phi_u \triangleright \Gamma \Vdash_{\mathtt{n}}^{(0,0,d_u)} u : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, such that $d = 1 + d_u$. By applying the *i.h.* to Φ_u, we get $u \in \mathtt{ne}_{\mathtt{n}}$ and $d_u = |t|_{\mathtt{n}}$. Thus, we can conclude that with $d = 1 + d_u = 1 + |t|_{\mathtt{n}} = |up|_{\mathtt{n}}$ (satisfying 1c) and $up \in \mathtt{ne}_{\mathtt{n}}$ (satisfying 1a).

(2) By induction over Π:

- Case Π ends with (emp). Then, $s = \varepsilon$ and $\Pi \triangleright \emptyset \Vdash_{\mathtt{n}}^{(0,0,0)} \varepsilon : \emptyset$, where $d = 0 = |\varepsilon|_{\mathtt{n}}$. Thus, we can conclude.
- Case Π ends with (upd). Then, $s = \mathtt{upd}_l(u, q)$ and Π has two premises $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}$ and $\Pi_q \triangleright \Delta_q \Vdash_{\mathtt{n}}^{(b_q,m_q,d_q)} q : \{l : \mathscr{L}\}; \mathscr{T}_q$, where $\Delta = \Gamma_u + \Delta_q$, $\mathscr{T} = \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_q$, and $d = d_u + d_q$. Since we are assuming that $\mathscr{T}$ is tight, then $\mathscr{M} = [\,]$. Therefore, $b_u = m_u = d_u = 0$. By applying the *i.h.* to Π_q, we get $b_q = m_q = d_q = 0$. Thus, we can conclude with $b = b_u + b_q = m = m_u + m_q = d = d_u + d_q = 0$. □

Lemma 8 (Normal forms take zero steps). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} t : \mu$ *be a tight derivation. Then,* $b = m = 0$ *iff* $t \in \mathtt{no}_{\mathtt{n}}$.

Proof.

⇒) By Lemma 7.1.1a.

⇐) By induction on $t \in \mathtt{no}_{\mathtt{n}}$:

– Case $t = \lambda x.u$. Then, Φ must end with (λ), (m), or ($\lambda_{\mathtt{p}}$):

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Rule (m). Then, Φ does not end with a tight monadic type. Therefore, this case does not apply.
- Rules (λ) or ($\lambda_{\mathtt{p}}$). Then, $\mu = \mathscr{T} \Rightarrow ((\mathscr{M} \to \mu) \times \mathscr{T}')$ or $\mu = \mathscr{T} \Rightarrow (\mathtt{a} \times \mathscr{T}')$, respectively. In the former case, Φ does not conclude with a tight type; therefore, it cannot apply. In the latter case $b = m = 0$ by definition, and we can conclude.

– Case $\neg\mathtt{abs}(t)$. Then $t \in \mathtt{ne}_{\mathtt{n}}$:

- Case $t = x$. Then Φ must end with rule (ax) and $b = m = 0$ by definition.
- Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{n}}$. Then Φ must end with rule (@), (m), or ($@_{\mathtt{p}}$):
  - Rule (@). Then, Φ has two premises $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}')$ and $\Phi_p \rhd \Gamma_p \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{M}$, such that $\Gamma = \Gamma_u + \Gamma_p$ is tight, $b = 1 + b_u + b_p$, $m = m_u + m_p$, and $d = d_u + d_p$. Moreover, Γ_u and Γ_p are tight. By applying Lemma 34 to u, we get that $\mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}') = \mathtt{n}$, which is a contradiction. Therefore, this case cannot apply.
  - Rule (m). Then, Φ does not end with a tight monadic type. Therefore, this case does not apply.
  - Rule ($@_{\mathtt{p}}$). Then, Φ has one premise $\Phi_u \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d_u)} u : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, such that $d = 1 + d_u$. By applying the *i.h.* to u, we have $b = m = 0$. Therefore, we can conclude. □

Lemma 9 (Normal configurations have correct size). *Let* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t, s) : \kappa$ *be a tight derivation of the following form:*

$$\frac{\Phi_t \rhd \Gamma_t \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}') \quad \Pi_s \rhd \Delta_s \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} s : \mathscr{T}}{\Gamma_t + \Delta_s \Vdash^{(b_t+b_s,m_t+m_s,d_t+d_s)} (t, s) : \tau \times \mathscr{T}'}(\mathtt{conf})$$

where $\Gamma = \Gamma_t + \Delta_s$, $\kappa = (\tau \times \mathscr{T}')$, $b = b_t + b_s$, $m = m_t + m_s$, *and* $d = d_t + d_s$. *If* $b_t = m_t = 0$, *then* (*i*) $d = |(t, s)|_{\mathtt{n}}$ *and* $b = m = 0$.

Proof. Since Γ is tight, then so are Γ_t and Δ_s. Since $b_t = m_t = 0$, then $t \in \mathtt{no}_{\mathtt{n}}$ by Lemma 7.1.1a and $d_t = |t|_{\mathtt{n}}$ by Lemma 7.1.1b. Therefore, $\mathscr{T} = \mathscr{T}'$ by Lemma 6. Therefore, $\mathscr{T}$ is tight and $b_s = m_s = d_s = 0$ by Lemma 7.2. Thus, we can conclude with $d = d_t + d_s = d_t = |t|_{\mathtt{n}} = |(t, s)|_{\mathtt{n}}$ and $b = b_t + b_s = m = m_t + m_s = 0$. □

Lemma 10 (Normal configurations take zero steps). *Let* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t, s) : \kappa$ *be a tight derivation. Then,* $b = m = 0$ *iff* (t, s) *is an unblocked normal configuration.*

Proof. Derivation Φ must be of the following form:

$$\frac{\Phi_t \rhd \Gamma_t \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}') \qquad \Pi_s \rhd \Delta_s \Vdash_{\mathtt{n}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{\Gamma_t + \Delta_s \Vdash^{(b_t+b_s,m_t+m_s,d_t+d_s)} (t, s) : \tau \times \mathscr{T}'}(\mathtt{conf})$$

where $\Gamma = \Gamma_t + \Delta_s$, $\kappa = (\tau \times \mathscr{T}')$, $b = b_t + b_s$, $m = m_t + m_s$, and $d = d_t + d_s$.

- ⇒) Let $b = m = 0$. Then, $b_t = m_t = 0$, and thus, $t \in \mathtt{no}_{\mathtt{n}}$, by Lemma 8. And we can conclude that (t, s) is in normal form.
- ⇐) Let (t, s) be in normal form. Then, $t \in \mathtt{no}_{\mathtt{n}}$, by definition. Thus, $b_t = m_t = 0$, by Lemma 8. And we can conclude that $b = m = 0$, by Lemma 9. □

Lemma 11 (Tight typability of states and normal forms).

(1) *Let s be a state. Then, there exists a tight derivation* $\Pi_s \rhd \emptyset \Vdash_{\mathtt{n}}^{(0,0,0)} s : \mathscr{T}$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

(2) *Let* $t \in \mathtt{no}_\mathtt{n}$. *Then, for any* $\mathscr{T}$, *there exists a tight derivation* $\Phi \triangleright \Gamma \Vdash_\mathtt{n}^{(0,0,d)} t : \mathscr{T} \Rightarrow (\mathtt{tt}_\mathtt{n} \times \mathscr{T})$, *such that* $d = |t|_\mathtt{n}$.

Proof.

(1) By induction on s:

- Case $s = \varepsilon$. Then, we can build $\Pi \triangleright \emptyset \Vdash_\mathtt{n}^{(0,0,0)} \varepsilon : \emptyset$ by rule (emp), such that $\mathscr{T} = \emptyset$ is tight.
- Case $s = \mathtt{upd}_l(t, q)$. By applying the *i.h.* to q, there exists a tight derivation $\Pi_q \triangleright \Vdash_\mathtt{n}^{(0,0,0)} q : \{l : \mathscr{L}\}; \mathscr{T}_0$. Therefore, we can build Π_s as follows:

$$\dfrac{\dfrac{}{\emptyset \Vdash^{(0,0,0)} t : [\,]}\,(\mathtt{m}) \qquad \Pi_q}{\emptyset \Vdash^{(0,0,0)} \mathtt{upd}_l(t, q) : \{l : [\,]\}; \{l : [\,] \cdot \mathscr{L}\}; \mathscr{T}_0}\,(\mathtt{upd})$$

where $\mathscr{T} = \{l : [\,] \cdot \mathscr{L}\}; \mathscr{T}_0$ is tight.

(2) We start by refining the original statement as follows:

a. Let $t \in \mathtt{ne}_\mathtt{n}$. Then, for any $\mathscr{T}$, there exists a tight derivation $\Phi \triangleright \Gamma \Vdash_\mathtt{n}^{(0,0,d)} t : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T})$, such that $d = |t|_\mathtt{n}$.

b. Let $t \in \mathtt{no}_\mathtt{n}$. Then, for any $\mathscr{T}$, there exists a tight derivation $\Phi \triangleright \Gamma \Vdash_\mathtt{n}^{(0,0,d)} t : \mathscr{T} \Rightarrow (\mathtt{tt}_\mathtt{n} \times \mathscr{T})$, such that $d = |t|_\mathtt{n}$.

The proof now goes as follows:

a. We reason by induction over $t \in \mathtt{ne}$:

- Case $t = x$. Then we can build Φ by using rule (ax), such that $b = m = d = 0 = |x|_\mathtt{n}$ and $\mathtt{tt}_\mathtt{n} = \mathtt{n}$.
- Case $t = up$, such that $u \in \mathtt{ne}_\mathtt{n}$. By applying the *i.h.* (a.) to u, we get $\Phi_u \triangleright \Gamma_u \Vdash_\mathtt{n}^{(0,0,d_u)} u : \mathscr{T} \Rightarrow (\mathtt{tt}_\mathtt{n}{}' \times \mathscr{T})$, such that Γ_u is tight and $d_u = |t|_\mathtt{n}$, for any $\mathscr{T}$. Moreover, since Γ_u and $u \in \mathtt{ne}_\mathtt{n}$, then $\mathtt{tt}_\mathtt{n} = \mathtt{n}$, according to Lemma 34. Therefore, we can build Φ is as follows:

$$\dfrac{\Phi_u}{\Gamma_u \Vdash^{(0,0,1+d_u)} up : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T})}\,(@_\mathtt{p})$$

such that $\Gamma = \Gamma_u$, and $d = 1 + d_u = 1 + |u|_\mathtt{u} = |up|_\mathtt{n}$.

b. We reason by induction over $t \in \mathtt{no}$:

- Case $t = \lambda x.u$. Then we can build Φ by using rule ($\lambda_\mathtt{p}$), such that $b = m = d = 0 = |\lambda x.u|_\mathtt{n}$ and $\mathtt{tt}_\mathtt{n} = \mathtt{a}$.
- Case $\neg\mathtt{abs}(t)$. Then $t \in \mathtt{ne}_\mathtt{n}$ and we can conclude by (a.).

□

Lemma 12 (All unblocked normal configurations are tightly typable). *If* (t, s) *is a normal but unblocked configuration, then there exists a tight typing derivation* $\Phi \triangleright \Gamma \Vdash_\mathtt{n}^{(0,0,d)} (t, s) : \kappa$, *such that* $d = |(t, s)|_\mathtt{n}$.

Proof. By Lemma 11. □

Lemma 35. *Let* $\Phi \triangleright \Gamma \Vdash_\mathtt{n}^{(b,m,d)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, *such that* Γ *is tight. If* $\tau = \mathtt{n}$, *then* $\neg\mathtt{abs}(t)$.

Proof. By induction over Φ:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case Φ ends with rule (ax), ($@$), (get), (set), ($@_{\mathtt{p}}$), then $\neg\mathtt{abs}(t)$ holds by definition.
- Case Φ ends with rule (m). Then Φ does not conclude with a monadic type. Therefore, these cases cannot apply.
- Case Φ ends with rules (λ) or ($\lambda_{\mathtt{p}}$). Then, $t = \lambda x.u$ and $\tau = \mathscr{M} \to \mu$ or $\tau = \mathtt{a}$, respectively. Therefore, these case do not apply.

□

Lemma 36 (Multi-types can split and merge). *Let* $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b = +_{i\in I} b_i$, $m = +_{i\in I} m_i$, *and* $d = +_{i\in I}$. *Then,* $\Phi_t \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} t : \mathscr{M}$ *iff* $(\Phi_t^i \rhd \Gamma_i \Vdash_{\mathtt{n}}^{(b_i,m_i,d_i)} t : \mathscr{M}_i)_{i\in I}$.

Proof. We omit this proof given its similarity with the proof for Weak Open CBN with Global State. □

Lemma 37 (States can Split). *Let* $\Pi_s \rhd \Delta \Vdash_{\mathtt{n}}^{(b,m,d)} s : \mathscr{T}$, *such that* $l \in \mathtt{dom}(\mathscr{T})$. *Then,* $s \equiv \mathtt{upd}_l(t,q)$, $\mathscr{T}(l) = \mathscr{M}\cdot\mathscr{L}$, $\Phi_t \rhd \Gamma_t \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mathscr{M}$ *and* $\Pi_q \rhd \Delta_q \Vdash_{\mathtt{n}}^{(b_q,m_q,d_q)} q : \{l : \mathscr{L}\}; \mathscr{T}'$, *such that* $\Delta = \Gamma_t + \Delta_q$, $\mathscr{T} = \{l : \mathscr{T}(l)\}; \mathscr{T}'$, $b = b_t + b_q$, $m = m_t + m_q$, *and* $d = d_t + d_q$.

Proof. The proof follows by induction on the structure of s:

- Case $s = \varepsilon$. Then the statement is vacuously true.
- Case $s = \mathtt{upd}_{l'}(u,q')$. Then Φ_s is of the form:

$$\frac{\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}' \qquad \Pi_{q'} \rhd \Delta_{q'} \Vdash_{\mathtt{n}}^{(b_{q'},m_{q'},d_{q'})} q' : \{l' : \mathscr{L}\}; \mathscr{T}_{q'}}{\Gamma_u + \Delta_{q'} \Vdash^{(b_u+b_{q'},m_u+m_{q'},d_u+d_{q'})} \mathtt{upd}_{l'}(u,q') : \{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \mathscr{T}_{q'}}(\mathtt{upd})$$

  where $\Delta = \Gamma_u + \Delta_{q'}$, $\mathscr{T} = \{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \mathscr{T}_{q'}$, $b = b_u + b_{q'}$, $m = m_u + m_{q'}$, and $d = d_u + d_{q'}$. We consider two cases:

  - Case $l' = l$. Then we simply take $t = u$, $q = q'$, $\mathscr{M}' = \mathscr{M}$, and $\mathscr{L}' = \mathscr{L}$, and we are done.
  - Case $l' \neq l$. Since $l \in \mathtt{dom}(\{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \mathscr{T}_{q'})$ and $l' \neq l$, then $l \in \mathtt{dom}(\mathscr{T}_{q'})$. By applying the *i.h.* to q', we have that $q' \equiv \mathtt{upd}_l(u', q'')$, $(\{l' : \mathscr{L}'\}; \mathscr{T}_{q'})(l) = \mathscr{T}_{q'}(l) = \mathscr{M}''\cdot\mathscr{L}''$, $\Phi_{u'} \rhd \Gamma_{u'} \Vdash_{\mathtt{n}}^{(b_{u'},m_{u'},d_{u'})} u' : \mathscr{M}''$ and $\Pi_{q''} \rhd \Delta_{q''} \Vdash_{\mathtt{n}}^{(b_{q''},m_{q''},d_{q''})} q'' : conjil : \mathscr{L}''; \{l : \mathscr{L}''\}; \mathscr{T}_{q''}$, such that $\Delta_{q'} = \Gamma_{u'} + \Delta_{q''}$, $\{l' : \mathscr{L}'\}; \mathscr{T}_{q'} = \{l : \mathscr{M}''\cdot\mathscr{L}''\}; \{l' : \mathscr{L}'\}; \mathscr{T}_{q''}$, $b_{q'} = b_{u'} + b_{q''}$, $m_{q'} = m_{u'} + m_{q''}$, and $d_{q'} = d_{u'} + d_{q''}$. Therefore, $\mathscr{T}_{q'} = \{l : \mathscr{M}''\cdot\mathscr{L}''\}; \mathscr{T}_{q''}$. But $s = \mathtt{upd}_{l'}(u, \mathtt{upd}_l(u', q'')) \equiv \mathtt{upd}_l(u', \mathtt{upd}_{l'}(u, q''))$, so we can take $t = u'$, $q = \mathtt{upd}_{l'}(u, q'')$, $\mathscr{M} = \mathscr{M}''$, $\mathscr{L} = \mathscr{L}''$, and $\mathscr{T}' = \{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \mathscr{T}_{q''}$, and consider Φ_q to be the following derivation:

$$\frac{\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}' \qquad \Pi_{q''} \rhd \Delta_{q''} \Vdash_{\mathtt{n}}^{(b_{q''},m_{q''},d_{q''})} q'' : \{l' : \mathscr{L}'\}; \{l : \mathscr{L}''\}; \mathscr{T}_{q''}}{\Gamma_u + \Delta_{q''} \Vdash^{(b_u+b_{q''},m_u+m_{q''},d_u+d_{q''})} \mathtt{upd}_{l'}(u,q'') : \{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \{l : \mathscr{L}''\}; \mathscr{T}_{q''}}(\mathtt{upd}) =$$

  where $\Delta_q = \Gamma_u + \Delta_{q''}$ and $\mathscr{T}_q = \{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \{l : \mathscr{L}''\}; \mathscr{T}_{q''}$. We can then conclude with the following observations:

  * $\Gamma_t + \Delta_q = \Gamma_{u'} + \Gamma_u + \Delta_{q''} = \Gamma_u + \Delta_{q'} = \Delta$,
  * Since $\mathscr{T} = \{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \mathscr{T}_{q'}$ and $l' \neq l$, then $\mathscr{T}(l) = \mathscr{T}_{q'}(l)$ and

$$\begin{aligned}\mathscr{T} = \{l' : \mathscr{M}'\cdot\mathscr{L}\}; \mathscr{T}_{q'} &= \{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \{l : \mathscr{M}''\cdot\mathscr{L}''\}; \mathscr{T}_{q''} \\ &= \{l : \mathscr{T}_{q'}(l)\}; \{l' : \mathscr{M}'\cdot\mathscr{L}'\}; \mathscr{T}_{q''} \\ &= \{l : \mathscr{T}(l)\}; \mathscr{T}'\end{aligned}$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

∗ $b_v + b_q = b_{u'} + b_u + b_{q''} = b_u + b_{q'} = b$, $m_v + m_q = m_{u'} + m_u + m_{q''} = m_u + m_{q'} = m$ and $d_v + d_q = d_{u'} + d_u + d_{q''} = d_u + d_{q'} = d$. □

Lemma 38 (Substitution). *If* $\Phi_t \triangleright \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mu$ *and* $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}$, *then* $\Phi_{t\{x\setminus u\}} \triangleright \Gamma_t + \Gamma_u \Vdash^{(b_t+b_u,m_t+m_u,d_t+d_u)} t\{x \setminus u\} : \mu$.

Proof. We generalize the original statement by allowing Φ_t to conclude with either a monadic type μ or a multi-type $\mathscr{M}_0$. Let $\mathscr{G} \in \{\mu, \mathscr{M}_0\}$. Then the statement is as follows:

If $\Phi_t \triangleright \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mathscr{G}$ and $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}$, then $\Phi_{t\{x\setminus u\}} \triangleright \Gamma_t + \Gamma_u \Vdash_{\mathtt{n}}^{(b_t+b_u,m_t+m_u,d_t+d_u)} t\{x \setminus u\} : \mathscr{G}$.

The proof follows by induction over Φ_t, according to the last rule of Φ_t:

- Case $\mathscr{G} = \mu$. Then, Φ_t must end with rule (ax), (λ), ($\lambda_{\mathtt{p}}$), (@), (get), (set), or (@$_{\mathtt{p}}$):
  - Rule (ax). Then, $t = y$ and there are two cases:
    - ∗ $y = x$. Then, $x\{x \setminus u\} = u$ and $\Phi_t \triangleright \emptyset; (x : [\mu]) \Vdash_{\mathtt{n}}^{(0,0,0)} x : \mu$, where $\Gamma_t = \emptyset$, $\mathscr{M} = [\mu]$, and $b_t = m_t = d_t = 0$. Moreover, Φ_u must end with rule (m), and have one premise $\Phi'_u \triangleright \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mu$. So we can take $\Phi_{t\{x\setminus u\}} = \Phi'_u$, and conclude with $\Gamma_t + \Gamma_u = \Gamma_u$, $b_t + b_u = b_u$, $m_t + m_u = m_u$, and $d_t + d_u = d_u$.
    - ∗ $y \neq x$. Then, $y\{x \setminus u\} = y$, $\Gamma_t = (y : [\mu])$, $\mathscr{M} = [\,]$, $b_t = m_t = d_t = 0$. Moreover, Φ_u ends with rule (m) with no premises. Therefore, $\Gamma_u = \emptyset$, and $b_u = m_u = d_u = 0$. So we can pick $\Phi_{t\{x\setminus u\}} = \Phi_t$, where $\Gamma_t + \Gamma_u = \Gamma_t$, $b_t + b_u = b_t = 0$, $m_t + m_u = m_t = 0$, and $d_t + d_u = d_t = 0$.
  - Case Φ_t ends with (λ). Then $t = \lambda y.p$, $\mu = \mathscr{T} \Rightarrow ((\mathscr{M}' \to \mu') \times \mathscr{T})$, and Φ_t has one premise $\Phi_p \triangleright (\Gamma_p; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mu'$, where $\Gamma_t = \Gamma_p$ (since we can assume that $y \neq x$ by α-conversion), $b_t = b_p$, $m_t = m_p$, and $d_t = d_p$. By applying the *i.h.* to Φ_p, there exists $\Phi_{p\{x\setminus u\}} \triangleright (\Gamma_p; y : \mathscr{M}') + \Gamma_u \Vdash_{\mathtt{n}}^{(b_p+b_u,m_p+m_u,d_p+d_u)} p\{x \setminus u\} : \mu'$. Moreover, by α-conversion, we can assume since $y \notin \mathtt{dom}(\Gamma_u)$, so that $(\Gamma_p; y : \mathscr{M}') + \Gamma_u = \Gamma_p + \Gamma_u; y : \mathscr{M}'$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{\Phi_{p\{x\setminus u\}}}{\Gamma_p + \Gamma_u \Vdash^{(b_p+b_u,m_p+m_u,d_p+d_u)} \lambda y.(p\{x \setminus u\}) : \mathscr{T} \Rightarrow ((\mathscr{M}' \to \mu) \times \mathscr{T})}(\lambda)$$

    And we can conclude since $\Gamma_t + \Gamma_u = \Gamma_p + \Gamma_u$, $b_t + b_u = b_p + b_u$, $m_t + m_u = m_p + m_u$, $d_t + d_u = d_p + d_u$, and $t\{x \setminus u\} = \lambda y.(p\{x \setminus u\})$.
  - Case Φ_t ends with ($\lambda_{\mathtt{p}}$). Then $t = \lambda y.p$, $\mu = \mathscr{T} \Rightarrow (\mathtt{a} \times \mathscr{T})$, and $\Phi_t \triangleright \emptyset; x : [\,] \Vdash_{\mathtt{n}}^{(0,0,0)} \lambda y.p : \mathtt{a}$, such that $\Gamma_t = \emptyset$, $\mathscr{M} = [\,]$, and $b_t = m_t = d_t = 0$. Therefore, Φ_u ends with rule (m) with no premises, and $\Gamma_u = \emptyset$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{}{\emptyset \Vdash^{(0,0,0)} (\lambda y.p)\{x \setminus u\} : \mathscr{T} \Rightarrow (\mathtt{a} \times \mathscr{T})}(\lambda_{\mathtt{p}})$$

    And we can conclude with $\Gamma_t + \Gamma_u = \emptyset$, $b_t + b_u = m_t + m_u = d_t + d_u = 0$, and $t\{x \setminus u\} = \lambda y.(p\{x \setminus u\})$.
  - Rule (@). Then $t = rp$, $\mu = \mathscr{T} \Rightarrow \kappa$, and Φ_t has two premises $\Phi_r \triangleright \Gamma_r; x : \mathscr{M}_1 \Vdash_{\mathtt{n}}^{(b_r,m_r,d_r)} r : \mathscr{T} \Rightarrow ((\mathscr{M}' \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}')$ and $\Phi_p \triangleright \Gamma_p; x : \mathscr{M}_2 \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{M}'$, where $\Gamma_t = \Gamma_r + \Gamma_p$, $\mathscr{M} = \mathscr{M}_1 \sqcup \mathscr{M}_2$, $b_t = 1 + b_r + b_p$, $m_t = m_r + m_p$, and $d_t = d_r + d_p$. By Lemma 36, there exist $(\Phi^i_u \triangleright \Gamma^i_u \Vdash_{\mathtt{n}}^{(b_i,m_i,d_i)} u : \mathscr{M}_i)_{i\in\{1,2\}}$, where $\Gamma_u = \Gamma^1_u + \Gamma^2_u$, $b_u = b_1 + b_2$, $m_u = m_1 + m_2$, and $d_u = d_1 + d_2$. By applying the *i.h.* to Φ_r and Φ_p, there exists $\Phi_{r\{x\setminus u\}} \triangleright \Gamma_r +$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$\Gamma_u^1 \Vdash_{\mathtt{n}}^{(b_r+b_1,m_r+m_1,d_r+d_1)} r\{x \setminus u\} : \mathscr{T} \Rightarrow ((\mathscr{M}' \to (\mathscr{T}' \Rightarrow \kappa) \times \mathscr{T}'))$ and $\Phi_{p\{x\setminus u\}} \rhd \Gamma_p + \Gamma_u^2 \Vdash_{\mathtt{n}}^{(b_p+b_2,m_p+m_2,d_p+d_2)} p\{x \setminus u\} : \mathscr{M}'$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{\Phi_{r\{x\setminus u\}} \qquad \Phi_{p\{x\setminus u\}}}{(\Gamma_r+\Gamma_p)+(\Gamma_u^1+\Gamma_u^2) \Vdash^{(1+b_r+b_p+b_1+b_2,m_r+m_p+m_1+m_2,d_r+d_p+d_1+d_2)} (r\{x \setminus u\})(p\{x \setminus u\}) : \mathscr{T} \Rightarrow \kappa}(@)$$

Thus, we can conclude with $t\{x \setminus u\} = (rp)\{x \setminus u\} = (r\{x \setminus u\})(p\{x \setminus u\})$, $\Gamma_t + \Gamma_u = (\Gamma_r + \Gamma_p) + (\Gamma_u^1 + \Gamma_u^2)$, $b_t + b_u = (1 + b_r + b_p) + (b_1 + b_2)$, $m_t + m_u = (m_r + m_p) + (m_1 + m_2)$, and $d_t + d_u = (d_r + d_p) + (d_1 + d_2)$.

- Case Φ_t ends with (get). Then, $t = \mathtt{get}_l(\lambda y.p)$, $\mu = \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and Φ_t has one premise $\Phi_p \rhd (\Gamma_t; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, where $b_t = b_p$, $m_t = 1 + m_p$, and $d_t = d_p$. By applying the *i.h.* over Φ_p, there exists $\Phi_{p\{x\setminus u\}} \rhd (\Gamma_t; y : \mathscr{M}') + \Gamma_u \Vdash_{\mathtt{n}}^{(b_p+b_u,m_p+m_u,d_p+d_u)} p\{x \setminus u\} : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$. Moreover, by α-conversion, we can assume since $y \notin \mathtt{dom}(\Gamma_u)$, so that $(\Gamma_t; y : \mathscr{M}') + \Gamma_u = \Gamma_t + \Gamma_u; y : \mathscr{M}'$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{\Phi_{p\{x\setminus u\}} \rhd \Gamma_t + \Gamma_u \Vdash_{\mathtt{n}}^{(b_p+b_u,m_p+m_u,d_p+d_u)} p\{x \setminus u\} : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}{\Gamma_t + \Gamma_u \Vdash^{(b_p+b_u,1+m_p+m_u,d_p+d_u)} \mathtt{get}_l(\lambda y.(p\{x \setminus u\})) : \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{get})$$

Thus, we can conclude with $\mathtt{get}_l(\lambda y.p)\{x \setminus u\} = \mathtt{get}_l(\lambda y.(p\{x \setminus u\}))$, and $b_t + b_u = b_p + b_u$, $m_t + m_u = 1 + m_p + m_u$, $d_t + d_u = d_p + d_u$.

- Case Φ_t ends with (set). Then $t = \mathtt{set}_l(r, p)$, $\mu = \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and Φ_t has two premises $\Phi_r \rhd \Gamma_r; x : \mathscr{M}_1 \Vdash_{\mathtt{n}}^{(b_r,m_r,d_r)} r : \mathscr{M}'$ and $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, where $\Gamma_t = \Gamma_r + \Gamma_p$, $\mathscr{M} = \mathscr{M}_1 \sqcup \mathscr{M}_2$, $b_t = b_r + b_p$, $m_t = 1 + m_r + m_p$, and $d_t = d_r + d_p$. By Lemma 36, there exist $\Phi_u^1 \rhd \Gamma_u^1 \Vdash_{\mathtt{n}}^{(b_u^1,m_u^1,d_u^1)} u : \mathscr{M}_1$ and $\Phi_u^2 \rhd \Gamma_u^2 \Vdash_{\mathtt{n}}^{(b_u^2,m_u^2,d_u^2)} u : \mathscr{M}_2$, where $\Gamma_u = \Gamma_u^1 + \Gamma_u^2$, $b_u = b_u^1 + b_u^2$, $m_u = m_u^1 + m_u^2$, and $d_u = d_u^1 + d_u^2$. By applying *i.h.* to Φ_r and Φ_p, there exist $\Phi_{r\{x\setminus u\}} \rhd \Gamma_r + \Gamma_u^1 \Vdash_{\mathtt{n}}^{(b_r+b_u^1,m_r+m_u^1,d_r+d_u^1)} r\{x \setminus u\} : \mathscr{M}'$ and $\Phi_{p\{x\setminus u\}} \rhd \Gamma_p + \Gamma_u^2 \Vdash_{\mathtt{n}}^{(b_p+b_u^2,m_p+m_u^2,d_p+d_u^2)} p\{x \setminus u\} : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{\Phi_{r\{x\setminus u\}} \qquad \Phi_{p\{x\setminus u\}}}{(\Gamma_r+\Gamma_p)+(\Gamma_u^1+\Gamma_u^2) \Vdash^{(b_r+b_p+b_u^1+b_u^2,1+m_r+m_p+m_u^1+m_u^2,d_r+d_p+d_u^1+d_u^2)} \mathtt{set}_l(r\{x \setminus u\}, p\{x \setminus u\}) : \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{set})$$

Thus, we can conclude with $\mathtt{set}_l(r, p)\{x \setminus u\} = \mathtt{set}_l((t\{x \setminus u\}), (p\{x \setminus u\}))$, $\Gamma_t + \Gamma_u = (\Gamma_r + \Gamma_p) + (\Gamma_u^1 + \Gamma_u^2)$, $b_t + b_u = (b_r + b_p) + (b_u^1 + b_u^2)$, $m_t + m_u = (1 + m_r + m_p) + (m_u^1 + m_u^2)$, and $d_t + d_u = (d_r + d_p) + (d_u^1 + d_u^2)$.

- Case Φ_t ends with ($@_{\mathtt{p}}$). Then $t = pr$, $\mu = \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, and Φ_t has one premise $\Phi_p \rhd \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, $b_t = b_p$, $m_t = m_p$, and $d_t = 1 + d_p$. By applying the *i.h.* to Φ_p, there exists $\Phi_{p\{x\setminus u\}} \rhd \Gamma_t + \Gamma_u \Vdash_{\mathtt{n}}^{(b_p+b_u,m_p+m_u,d_p+d_u)} p\{x \setminus u\} : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{\Phi_{p\{x\setminus u\}}}{\Gamma_t + \Gamma_u \Vdash^{(b_p+b_u,m_p+m_u,1+d_p+d_u)} (p\{x \setminus u\})(r\{x \setminus u\}) : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}(@_{\mathtt{p}})$$

Thus, we can conclude with $(pr)\{x \setminus u\} = (p\{x \setminus u\})(r\{x \setminus u\})$, $b_t + b_u = b_p + b_u$, $m_t + m_u = m_p + m_u$, and $d_t + d_u = 1 + d_p + d_u$.

- Case $\mathscr{G} = \mathscr{M}_0$. Then, Φ_t must end with rule ($\mathtt{m}$):
  - Case Φ_t ends with ($\mathtt{m}$). Then, Φ_t has premises $(\Phi_t^i \rhd \Gamma_t^i; x : \mathscr{M}_i \Vdash_{\mathtt{n}}^{(b_t^i,m_t^i,d_t^i)} t : \delta_i)_{i\in I}$, where $\Gamma_t = +_{i\in I}\Gamma_t^i$, $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $\mathscr{M}_0 = [\delta_i]_{i\in I}$, $b_t = +_{i\in I}b_t^i$, $m_t = +_{i\in I}m_t^i$, and $d_t = +_{i\in I}d_t^i$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

By Lemma 36, there exist $(\Phi^i_u \rhd \Gamma^i_u \Vdash_{\mathtt{n}}^{(b^i_u,m^i_u,d^i_u)} u : \mathscr{M}_i)_{i\in I}$, where $\Gamma_u = +_{i\in I}\Gamma^i_u$, $b_u = +_{i\in I} b^i_u$, $m_u = +_{i\in I} m^i_u$, and $d_u = +_{i\in I} d^i_u$. By the *i.h.* over each Φ^i_u, we have $(\Phi^i_{t\{x\setminus u\}} \rhd \Gamma^i_t + \Gamma^i_u \Vdash_{\mathtt{n}}^{(b^i_t+b^i_u,m^i_t+m^i_u,d^i_t+d^i_u)} t\{x \setminus u\} : \delta_i)_{i\in I}$. Therefore, we can build $\Phi_{t\{x\setminus u\}}$ as follows:

$$\frac{(\Phi^i_{t\{x\setminus u\}})_{i\in I}}{+_{i\in I}(\Gamma^i_t + \Gamma^i_u) \Vdash^{(+_{i\in I}(b_i+b^i_u),+_{i\in I}(m_i+m^i_u),+_{i\in I}(d_i+d^i_u))} t\{x \setminus u\} : [\delta_i]_{i\in I}}(\mathtt{m})$$

Thus, we can conclude with $\Gamma_t + \Gamma_u = +_{i\in I}\Gamma^i_t +_{i\in I} \Gamma^i_u = +_{i\in I}(\Gamma^i_t + \Gamma^i_u)$, $b_t + b_u = +_{i\in I} b^i_t +_{i\in I} b^i_u = +_{i\in I}(b^i_t + b^i_u)$, $m_t + m_u = +_{i\in I} m^i_t +_{i\in I} m^i_u = +_{i\in I}(m^i_t + m^i_u)$, and $d_t + d_u = +_{i\in I} d^i_t +_{i\in I} d^i_u = +_{i\in I}(d^i_t + d^i_u)$. □

Lemma 13 (Quantitative subject reduction). *Let* $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, *where* $\mathtt{r} \in \{\beta_{\mathtt{n}}, \mathtt{g}, \mathtt{s}\}$. *If* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t,s) : \kappa$ *is a tight derivation, then* $\Phi' \rhd \Gamma \Vdash_{\mathtt{n}}^{(b',m',d)} (u,q) : \kappa$, *where* $\mathtt{r} = \beta_{\mathtt{n}}$ *implies* $b' = b-1$ *and* $m' = m$, *while* $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ *implies* $b' = b$ *and* $m' = m-1$.

Proof. We show a stronger statement of the form:

Let $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, where $\mathtt{r} \in \{\beta_{\mathtt{n}}, \mathtt{g}, \mathtt{s}\}$. If $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t,s) : \kappa$, Γ is tight, and κ is tight or $\neg\mathtt{abs}(t)$, there exists $\Phi' \rhd \Gamma \Vdash_{\mathtt{n}}^{(b',m',d)} (u,q) : \kappa$, where $\mathtt{r} = \beta_{\mathtt{n}}$ implies $b' = b-1$ and $m' = m$, while $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ implies $b' = b$ and $m' = m-1$.

We proceed by induction over $(t,s) \rightsquigarrow_{\mathtt{n}} (u,q)$:

- Case $(t,s) = ((\lambda x.p)r, s) \rightsquigarrow_{\beta_{\mathtt{n}}} (p\{x \setminus r\}, s) = (u,q)$. Then, Φ has two premises $\Phi_t \rhd \Gamma_t \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow \kappa$ and $\Pi_s \rhd \Delta_s \Vdash_{\mathtt{n}}^{(b_s,m_s,d_s)} s : \mathscr{T}$, such that $\Gamma = \Gamma_t + \Delta_s$, $b = b_t + b_s$, $m = m_t + m_s$, and $d = d_t + d_s$. Then, Φ_t must end with rule $(@)$ or $(@_{\mathtt{p}})$. Let us assume that Φ_t ends with rule $(@_{\mathtt{p}})$. Then, Φ_t has one premise $\Phi_{\lambda x.p} \rhd \Gamma_{\lambda x.p} \Vdash_{\mathtt{n}}^{(b_{\lambda x.p},m_{\lambda x.p},d_{\lambda x.p})} \lambda x.p : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, such that $\Gamma_t = \Gamma_{\lambda x.p}$, $b_t = b_{\lambda x.p}$, $m_t = m_{\lambda x.p}$, and $d_t = d_{\lambda x.p}$. However, according to Lemma 35, $\neg\mathtt{abs}(\lambda x.p)$ should hold. This gives a contradiction, meaning that this case does not apply. Now, let us assume Φ_t ends with rule $(@)$. Then, Φ_t must be of the following form:

$$\frac{\dfrac{\Phi_p \rhd \Gamma_{\lambda x.p}; x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{T} \Rightarrow \kappa}{\Gamma_{\lambda x.p} \Vdash^{(b_p,m_p,d_p)} \lambda x.p : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T} \Rightarrow \kappa)) \times \mathscr{T})}(\lambda) \qquad \Phi_r \rhd \Gamma_r \Vdash_{\mathtt{n}}^{(b_r,m_r,d_r)} r : \mathscr{M}}{\Gamma_{\lambda x.p} + \Gamma_r \Vdash^{(1+b_p+b_r,m_p+m_r,d_p+d_r)} (\lambda x.p)r : \mathscr{T} \Rightarrow \kappa}(@)$$

where $\Gamma_t = \Gamma_{\lambda x.p} + \Gamma_r$, $b_t = 1 + b_p + b_r$, $m_t = m_p + m_r$, and $d_t = d_p + d_r$. By Lemma 38, there exists $\Phi_{p\{x\setminus r\}} \rhd \Gamma_{\lambda x.p} + \Gamma_r \Vdash_{\mathtt{n}}^{(b_p+b_r,m_p+m_r,d_p+d_r)} p\{x \setminus r\} : \mathscr{T} \Rightarrow \kappa$. Therefore, we can build $\Phi_{(u,q)}$ as follows:

$$\frac{\Phi_{p\{x\setminus r\}} \qquad \Pi_s}{\Gamma_{\lambda x.p} + \Gamma_r + \Delta_s \Vdash^{(b_p+b_r+b_s,m_p+m_r+m_s,d_p+d_r+d_s)} (p\{x \setminus r\}, s) : \kappa}(\mathtt{conf})$$

where $b' = b_p + b_r + b_s$, $m' = m_p + m_r + m_s$, and $d = d_p + d_r + d_s$. Thus, we can conclude with $b - 1 = (b_t + b_s) - 1 = (1 + b_p + b_r + b_s) - 1 = b'$, $m = m_t + m_s = m_p + m_r + m_s = m'$, and $d = d_t + d_s = d_p + d_r + d_s = d'$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case $(t, s) = (\mathtt{get}_l(\lambda x.p), s) \rightsquigarrow_{\mathtt{n}} (p\{x \setminus r\}, s) = (u, q)$, such that $s \equiv \mathrm{upd}_l(r, s')$. Then Φ must be of the following form, where $\mathscr{T} = \{l : \Gamma_p(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0$:

$$\dfrac{\dfrac{\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{n}}^{(b_p, m_p, d_p)} p : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0 \Rightarrow \kappa}{\Gamma_p \setminus\!\setminus x \Vdash^{(b_p, 1+m_p, d_p)} \mathtt{get}_l(\lambda x.p) : \mathscr{T} \Rightarrow \kappa}\,(\mathtt{get}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{n}}^{(b_s, m_s, d_s)} s : \mathscr{T}}{\Gamma_p \setminus\!\setminus x + \Delta_s \Vdash^{(b_p+b_s, 1+m_p+m_s, d_p+d_s)} (\mathtt{get}_l(\lambda x.p), s) : \kappa}\,(\mathtt{conf})$$

where $\Gamma = \Gamma_p \setminus\!\setminus x + \Delta_s$, $b = b_p + b_s$, $m = 1 + m_p + m_s$, and $d = d_p + d_s$. Since $\Pi_s \triangleright \Delta_s \Vdash_{\mathtt{n}}^{(b_s, m_s, d_s)} s : \{l : \Gamma_p(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0$, we know, by Lemma 37, that $s \equiv \mathrm{upd}_l(r, s')$, $\Phi_r \triangleright \Delta_r \Vdash_{\mathtt{n}}^{(b_r, m_r, d_r)} r : \Gamma_p(x) \sqcup \mathscr{M}$ and $\Pi_{s'} \triangleright \Delta_{s'} \Vdash_{\mathtt{n}}^{(b_{s'}, m_{s'}, d_{s'})} s' : \{l : \mathscr{L}\}; \mathscr{T}_0$, such that $\Delta_s = \Gamma_r + \Delta_{s'}$, $b_s = b_r + b_{s'}$, $m_s = m_r + m_{s'}$, and $d_s = d_r + d_{s'}$. Therefore, by Lemma 36, there exist $\Phi_r^1 \triangleright \Gamma_r^1 \Vdash_{\mathtt{n}}^{(b_r^1, m_r^1, d_r^1)} r : \Gamma_p(x)$ and $\Phi_r^2 \triangleright \Gamma_r^2 \Vdash_{\mathtt{n}}^{(b_r^2, m_r^2, d_r^2)} r : \mathscr{M}$, such that $\Gamma_r = \Gamma_r^1 + \Gamma_r^2$, $b_r = b_r^1 + b_r^2$, $m_r = m_r^1 + m_r^2$, and $d_r = d_r^1 + d_r^2$. By Lemma 38 (applied to Φ_p and Φ_r^1), there exists $\Phi_{p\{x\setminus r\}} \triangleright \Gamma_p \setminus\!\setminus x + \Delta_r^1 \Vdash_{\mathtt{n}}^{(b_p+b_r^1, m_p+m_r^1, d_p+d_r^1)} p\{x \setminus r\} : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0 \Rightarrow \kappa$. Moreover, we can construct an alternative type derivation Π_s' for s as follows:

$$\dfrac{\Phi_r^2 \qquad \Pi_{s'}}{\Gamma_r^2 + \Delta_{s'} \Vdash^{(b_r^2+b_{s'}, m_r^2+m_{s'}, d_r^2+d_{s'})} \mathrm{upd}_l(r, s') : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0}\,(\mathrm{upd})$$

Let $\Pi_q = \Pi_s'$, where $q = \mathrm{upd}_l(r, s') = s$ and $\Delta_q = \Gamma_r^2 + \Delta_{s'}$. Therefore, we can build Φ' as follows:

$$\dfrac{\Phi_{p\{x\setminus r\}} \qquad \Pi_q}{(\Gamma_p \setminus\!\setminus x + \Gamma_r^1) + (\Gamma_r^2 + \Delta_{s'}) \Vdash^{(b_p+b_r^1+b_r^2+b_{s'}, m_p+m_r^1+m_r^2+m_{s'}, d_p+d_r^1+d_r^2+d_{s'})} (p\{x \setminus r\}, s) : \kappa}\,(\mathtt{conf})$$

where $b' = b_p + b_r^1 + b_r^2 + b_{s'}$, $m' = m_p + m_r^1 + m_r^2 + m_{s'}$, and $d' = d_p + d_r^1 + d_r^2 + d_{s'}$. Thus, we can conclude with $\Gamma = \Gamma_p + \Delta_s = \Gamma_p + (\Gamma_r + \Delta_{s'}) = \Gamma_p + (\Gamma_r^1 + \Gamma_r^2) + \Delta_{s'}$, $b = b_p + b_s = b_p + (b_r + b_{s'}) = b_p + (b_r^1 + b_r^2) + b_{s'} = b'$, $m - 1 = (1 + m_p + m_s) - 1 = (1 + m_p + m_r + m_{s'}) - 1 = (1 + m_p + m_r^1 + m_r^2 + m_{s'}) - 1 = m'$, $d = d_p + d_s = d_p + (d_r + d_{s'}) = d_p + (d_r^1 + d_r^2) + d_{s'} = d'$.

- Case $(t, s) = (\mathtt{set}_l(r, p), s) \rightsquigarrow_{\mathtt{n}} (p, \mathrm{upd}_l(r, s)) = (u, q)$. Then, Φ' must be of the following form, where $\mathscr{T} = \{l : \mathscr{L}\}; \mathscr{T}_0$:

$$\dfrac{\dfrac{\Phi_r \triangleright \Gamma_r \Vdash_{\mathtt{n}}^{(b_r, m_r, d_r)} r : \mathscr{M} \qquad \Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{n}}^{(b_p, m_p, d_p)} p : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0 \Rightarrow \kappa}{\Gamma_r + \Gamma_p \Vdash^{(b_r+b_p, 1+m_r+m_p, s_r+s_p)} \mathtt{set}_l(r, p) : \mathscr{T} \Rightarrow \kappa}\,(\mathtt{set}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{n}}^{(b_s, m_s, d_s)} s : \mathscr{T}}{(\Gamma_r + \Gamma_p) + \Delta_s \Vdash^{(b_r+b_p+b_s, 1+m_r+m_p+m_s, d_r+d_p+d_s)} (\mathtt{set}_l(r, p), s) : \kappa}\,(\mathtt{conf})$$

where $\Gamma = (\Gamma_r + \Gamma_p) + \Delta_s$, $b = b_r + b_p + b_s$, $m = 1 + m_r + m_p + m_s$ and $d = d_r + d_p + d_s$. Therefore, we can build derivation Π_q for $\mathrm{upd}_l(r, s)$ as follows:

$$\dfrac{\Phi_r \qquad \Pi_s}{\Gamma_r + \Delta_s \Vdash^{(b_r+b_s, m_r+m_s, d_r+d_s)} \mathrm{upd}_l(r, s) : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0}\,(\mathrm{upd})$$

Moreover, we can build Φ' as follows:

$$\dfrac{\Phi_p \qquad \Pi_q}{\Gamma_p + (\Gamma_r + \Delta_s) \Vdash^{(b_p+b_r+b_s, m_p+m_r+m_s, d_p+d_r+d_s)} (p, \mathrm{upd}_l(r, s)) : \kappa}\,(\mathtt{conf})$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

where $b' = b_p + b_r + b_s$, $m' = m_p + m_r + m_s$, and $d' = d_p + d_r + d_s$. Thus, we can conclude with $\Gamma = (\Gamma_r + \Gamma_p) + \Delta_s$, $b = b_r + b_p + b_s = b'$, $m - 1 = (1 + m_r + m_p + m_s) - 1 = m_r + m_p + m_s = m'$, and $d = d_r + d_p + d_s = d'$.

- Case $(t, s) = (pr, s) \rightsquigarrow_{\mathtt{n}} (p'r, q) = (u, q)$, such that $(p, s) \rightsquigarrow_{\mathtt{n}} (p', q)$. Then, Φ has two premises $\Phi_t \triangleright \Gamma_t \Vdash_{\mathtt{n}}^{(b_t, m_t, d_t)} t : \mathscr{T} \Rightarrow \kappa$ and $\Pi_s \triangleright \Delta_s \Vdash_{\mathtt{n}}^{(b_s, m_s, d_s)} s : \mathscr{T}$, such that $\Gamma = \Gamma_t + \Delta_s$, $b = b_t + b_s$, $m = m_t + m_s$, and $d = d_t + d_s$. Moreover, Φ_t must end with rule (@) or ($@_{\mathtt{p}}$):
  - Case Φ_t ends with (@). Then, Φ_t must be of the following form:

$$\frac{\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{n}}^{(b_p, m_p, d_p)} p : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}') \qquad \Phi_r \triangleright \Gamma_r \Vdash_{\mathtt{n}}^{(b_r, m_r, d_r)} r : \mathscr{M}}{\Gamma_p + \Gamma_r \Vdash^{(1 + b_p + b_r, m_p + m_r, d_p + d_r)} pr : \mathscr{T} \Rightarrow \kappa} (@)$$

    where $\Gamma_t = \Gamma_p + \Gamma_r$, $b_t = 1 + b_p + b_r$, $m_t = m_p + m_r$, and $d_t = d_p + d_r$. Therefore, we can build the following derivation for (p, s):

$$\frac{\Phi_p \qquad \Pi_s}{\Gamma_p + \Delta_s \Vdash^{(b_p + b_s, m_p + m_s, d_p + d_s)} (p, s) : (\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'} (\mathtt{conf})$$

    Moreover, Γ_p and Δ_s are tight, and $(p, s) \rightsquigarrow_{\mathtt{n}} (p', q)$ implies that $\neg \mathtt{abs}(p)$. By applying the *i.h.* over $\Phi_{(p,s)}$, there exists derivation $\Phi_{(p',q)}$ that must be of the following form:

$$\frac{\begin{array}{c}\Phi_p \triangleright \Gamma_{p'} \Vdash_{\mathtt{n}}^{(b_{p'}, m_{p'}, d_{p'})} p' : \mathscr{T}'' \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}') \\ \Pi_q \triangleright \Delta_q \Vdash_{\mathtt{n}}^{(b_q, m_q, d_q)} q : \mathscr{T}''\end{array}}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'} + b_q, m_{p'} + m_q, d_{p'} + d_q)} (p', q) : (\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'} (\mathtt{conf})$$

    such that $\Gamma_p + \Delta_s = \Gamma_{p'} + \Delta_q$ and $(b_{p'} + b_q, m_{p'} + m_q, d_{p'} + d_q)$ and $(b_p + b_s, m_p + m_s, d_p + d_s)$ are properly related. Therefore, we can build Φ' as follows:

$$\frac{\dfrac{\Phi_{p'} \qquad \Phi_r}{\Gamma_{p'} + \Gamma_r \Vdash^{(1 + b_{p'} + b_r, m_{p'} + m_r, d_{p'} + d_r)} p'r : \mathscr{T}'' \Rightarrow \kappa} (@) \qquad \Pi_q \triangleright \Delta_q \Vdash_{\mathtt{n}}^{(b_q, m_q, d_q)} q : \mathscr{T}''}{\Gamma_{p'} + \Gamma_r + \Delta_q \Vdash^{(1 + b_{p'} + b_r + b_q, m_{p'} + m_r + m_q, d_{p'} + d_r + d_q)} (p'r, q) : \kappa} (\mathtt{conf})$$

    where $b' = 1 + b_{p'} + b_r + b_q$, $m' = m_{p'} + m_r + m_q$, and $d' = d_{p'} + d_r + d_q$. Thus, we can conclude with $\Gamma = \Gamma_t + \Delta_s = (\Gamma_p + \Gamma_r) + \Delta_s = (\Gamma_{p'} + \Gamma_r) + \Delta_q$, and $(1 + b_{p'} + b_r + b_q, m_{p'} + m_r + m_q, d_{p'} + d_r + d_q)$ and $(1 + b_p + b_r + b_s, m_p + m_r + m_s, d_p + d_r + d_s)$ are related properly.
  - Case Φ_{pr} ends with ($@_{\mathtt{p}}$). Then, Φ must be of the following form:

$$\frac{\dfrac{\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{n}}^{(b_p, m_p, d_p)} p : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}{\Gamma_p \Vdash^{(b_p, m_p, 1 + d_p)} pr : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')} (@_{\mathtt{p}}) \qquad \Pi_s \triangleright \Delta \Vdash_{\mathtt{n}}^{(b_s, m_s, d_s)} s : \mathscr{T}}{\Gamma_p + \Delta_s \Vdash^{(b_p + b_s, m_p + m_s, 1 + d_p + d_s)} (pr, s) : \mathtt{n} \times \mathscr{T}'} (\mathtt{conf})$$

    where $\Gamma = \Gamma_p + \Delta_s$, $b = b_p + b_s$, $m = m_p + m_s$, and $s = d_p + d_s$. Therefore, we can build the following derivation $\Phi_{(p,s)}$:

$$\frac{\Phi_p \qquad \Pi_s}{\Gamma_p + \Delta_s \Vdash^{(b_p + b_s, m_p + m_s, d_p + d_s)} (p, s) : (\mathtt{n} \times \mathscr{T}')} (\mathtt{conf})$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Moreover, Γ_p and Δ_s are tight, and $(p,s) \rightsquigarrow_{\mathtt{n}} (p',q)$ implies that $\neg\mathtt{abs}(p)$. By applying the *i.h.* over $\Phi_{(p,s)}$, there exists a derivation for (p',q) that must be of the following form:

$$\frac{\Phi_{p'} \rhd \Gamma_{p'} \Vdash_{\mathtt{n}}^{(b_{p'},m_{p'},d_{p'})} p' : \mathscr{T}'' \Rightarrow (\mathtt{n} \times \mathscr{T}') \qquad \Pi_q \rhd \Delta_q \Vdash_{\mathtt{n}}^{(b_q,m_q,d_q)} q : \mathscr{T}''}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'}+b_q,m_{p'}+m_q,d_{p'}+d_q)} (p',q) : (\mathtt{n} \times \mathscr{T}')}(\mathtt{conf})$$

where $\Gamma_{p'} + \Delta_q = \Gamma_p + \Delta_s$, and $(b_{p'} + b_q, m_{p'} + m_q, d_{p'} + d_q)$ and $(b_p + b_s, m_p + m_s, d_p + d_s)$ are related properly. Therefore, we can build derivation Φ' as follows:

$$\frac{\dfrac{\Phi_{p'}}{\Gamma_{p'} \Vdash^{(b_{p'},m_{p'},d_{p'})} p'r : \mathscr{T}'' \Rightarrow (\mathtt{n} \times \mathscr{T}')}(\mathbb{C}_{\mathtt{p}}) \qquad \Pi_q \rhd \Delta_q \Vdash_{\mathtt{n}}^{(b_q,m_q,d_q)} q : \mathscr{T}''}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'}+b_q,m_{p'}+m_q,1+d_{p'}+d_q)} (p'r,q) : (\mathtt{n} \times \mathscr{T}')}(\mathtt{conf})$$

where $\Gamma_{p'} + \Delta_q = \Gamma_p + \Delta_s$, $b' = b_{p'} + b_q$, $m' = m_{p'} + m_q$, and $d' = 1 + d_{p'} + d_q$. Thus, we can conclude $\Gamma = \Gamma_p + \Delta_s = \Gamma_{p'} + \Delta_q$, and $(b_{p'} + b_q, m_{p'} + m_q, 1 + d_{p'} + d_q)$ and $(b_p + b_s, m_p + m_s, 1 + d_p + d_s)$ are properly related. □

Theorem 3 (Quantitative soundness). *If* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d)} (t,s) : \kappa$ *a tight derivation, then there exists* (u,q) *such that* $u \in \mathtt{no}_{\mathtt{n}}$ *and* $(t,s) \rightsquigarrow_{\mathtt{n}}^{(b,m)} (u,q)$ *with* b $\beta_{\mathtt{n}}$*-steps,* m $\mathtt{g}/\mathtt{s}$*-steps, and* $|(u,q)|_{\mathtt{n}} = d$.

Proof. We start by noting that Φ has two premises $\Phi_t \rhd \Gamma_t \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$ and $\Pi_s \rhd \Delta_s \Vdash_{\mathtt{n}}^{(b_s,m_s,d_s)} s : \mathscr{T}$, where $\Gamma = \Gamma_t + \Delta_s$ and $\kappa = \tau \times \mathscr{T}'$, such that τ and $\mathscr{T}'$ are tight. Now, the proof follows by induction on $b + m$:

- Case $b + m = 0$. Then, $b = m = 0$, and thus, $b_t = m_t = b_s = m_s = 0$. Therefore, (t,s) is a configuration in normal form, by Lemma 10, and $|(t,s)|_{\mathtt{n}} = d$, by Lemma 9. Thus, we can conclude by taking $u = t$ and $q = s$.
- Case $b + m > 0$. Then, $b > 0$ or $m > 0$, and thus, $b_t + b_s > 0$ or $m_t + m_s > 0$. Note that it cannot be that $b_t = m_t = 0$, because this would imply that $b = m = 0$ by Lemma 9. Therefore, it must be the case that $b_t + m_t > 0$, and thus, (t,s) is not a configuration in normal form, by Lemma 10. Moreover, note that (t,s) cannot be a blocked configuration, by Proposition 4. Therefore, by Proposition 3, there exists (t',s') such that $(t,s) \rightsquigarrow_{\mathtt{n}} (t',s')$. By Lemma 13, there exists $\Phi' \rhd \Gamma \Vdash_{\mathtt{n}}^{(b',m',d)} (t',s') : \kappa$, such that $b' + m' = b + m - 1$. By the *i.h.*, there exists (u,q), such that $u \in \mathtt{no}_{\mathtt{n}}$, $(t',s') \rightsquigarrow_{\mathtt{n}}^{(b',m')} (u,q)$ and $d = |(u,q)|_{\mathtt{n}}$. So we can conclude with $(t,s) \rightsquigarrow_{\mathtt{n}} (t',s') \rightsquigarrow_{\mathtt{n}}^{(b',m')} (u,q)$, which means that $(t,s) \rightsquigarrow_{\mathtt{n}}^{(b,m)} (u,q)$, as expected. □

Lemma 39 (Anti-substitution). *If* $\Phi_{t\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}} \Vdash_{\mathtt{n}}^{(b,m,d)} t\{x \setminus u\} : \mu$, *then* $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mu$ *and* $\Phi_u \rhd \Gamma_u \Vdash^{(b_u,m_u,d_u)} u : \mathscr{M}$, *such that* $\Gamma_{t\{x\setminus u\}} = \Gamma_t + \Gamma_u$, $b = b_t + b_u$, $m = m_t + m_u$, *and* $d = d_t + d_u$.

Proof. We generalize the original statement by allowing Φ_t to conclude with either a monadic type μ or a multi-type $\mathscr{M}_0$. Let $\mathscr{G} \in \{\mu, \mathscr{M}_0\}$. Then the statement is as follows:

If $\Phi_{t\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}} \Vdash_{v}^{(b,m,d)} t\{x \setminus u\} : \mathscr{G}$, then there exist $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_t,m_t,d_t)} t : \mathscr{G}$ and $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}$, such that $\Gamma_{t\{x\setminus u\}} = \Gamma_t + \Gamma_u$, $b = b_t + b_u$, $m = m_t + m_u$, and $d = d_t + d_u$.

The proof follows by induction over $\Phi_{t\{x\setminus u\}}$, according to the last rule of $\Phi_{t\{x\setminus u\}}$:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case $\mathscr{G} = \mu$. Let us assume $t = x$, such that $t\{x \setminus u\} = u$. Then, $\Phi_{t\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}} \Vdash_{\mathtt{n}}^{(b,m,d)} u : \mu$, and we can build $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : [\mu]$, where $\Gamma_u = \Gamma_{t\{x\setminus u\}}$, $b_u = b$, $m_u = m$, and $d_u = d$, by applying rule (m) to $\Phi_{t\{x\setminus u\}}$, and $\Phi_t \rhd x : [\mu] \Vdash_{\mathtt{n}}^{(0,0,0)} x : \mu$, where $\Gamma_t = \emptyset$, and $b_t = m_t = d_t = 0$, by using rule (ax). From now on, we are going to assume $t \neq x$ and consider the last rule used in $\Phi_{t\{x\setminus u\}}$, which must be (ax), (λ), ($\lambda_{\mathtt{p}}$), (@), (get), (set), or ($@_{\mathtt{p}}$):
  - Rule (ax). Then, $t = y$, $y\{x \setminus u\} = y$, and $\mathscr{M} = [\,]$. So we can pick $\Phi_t = \Phi_{t\{x\setminus u\}}$ and $\Phi_u \rhd \emptyset \Vdash_{\mathtt{n}}^{(0,0,0)} u : [\,]$, where $\Gamma_u = \emptyset$, and $b_u = m_u = d_u = 0$. And we can conclude with $\Gamma_{t\{x\setminus u\}} = \Gamma_t = \Gamma_t + \Gamma_u$, $b = b_t = b_t + b_u$, $m = m_t = m_t + m_u$, and $d = d_t = d_t + d_u$.
  - Rule (λ). Then, $t\{x \setminus u\} = \lambda y.(p\{x \setminus u\})$, $\mu = \mathscr{T} \Rightarrow ((\mathscr{M}' \to \mu') \times \mathscr{T})$, and $\Phi_{t\{x\setminus u\}}$ has one premise $\Phi_{p\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}}; y : \mathscr{M}' \Vdash_{\mathtt{n}}^{(b_{p\{x\setminus u\}}, m_{p\{x\setminus u\}}, d_{p\{x\setminus u\}})} p\{x \setminus u\} : \mu'$, $b = b_{p\{x\setminus u\}}$, $m = m_{p\{x\setminus u\}}$, and $d = d_{p\{x\setminus u\}}$. By applying the *i.h.* to $\Phi_{p\{x\setminus u\}}$, there exist $\Phi_p \rhd (\Gamma_p; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mu'$ and $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}$, where $\Gamma_{t\{x\setminus u\}} = \Gamma_p + \Gamma_u$, $b_{p\{x\setminus u\}} = b_p + b_u$, $m_{p\{x\setminus u\}} = m_p + m_u$, and $d_{p\{x\setminus u\}} = d_p + d_u$. Therefore, we can build Φ_t as follows:

$$\frac{\Phi_p}{\Gamma_p \Vdash^{(b_p,m_p,d_p)} \lambda y.p : \mathscr{T} \Rightarrow ((\mathscr{M}' \to \mu') \times \mathscr{T})}(\lambda)$$

    where $b_t = b_p$, $m_t = m_p$, and $d_t = d_p$. Thus, we can conclude.
  - Rule ($\lambda_{\mathtt{p}}$). Then, $t\{x \setminus u\} = \lambda y.(p\{x \setminus u\})$, $\mu = \mathscr{T} \Rightarrow (\mathtt{a} \times \mathscr{T})$, and $\Phi_{t\{x\setminus u\}} \rhd \emptyset \Vdash_{\mathtt{n}}^{(0,0,0)} \lambda y.(p\{x \setminus u\}) : \mathtt{a}$, such that $\Gamma_{t\{x\setminus u\}} = \emptyset$ and $b = m = d = 0$. Therefore, we can build $\Phi_u \rhd \emptyset \Vdash_{\mathtt{n}}^{(0,0,0)} u : [\,]$, where $\Gamma_u = \emptyset$, $\mathscr{M} = [\,]$, and $b_u = m_u = d_u = 0$. And we can build Φ_t as follows:

$$\frac{}{\emptyset; x : \mathscr{M} \Vdash^{(0,0,0)} \lambda y.p : \mathscr{T} \Rightarrow (\mathtt{a} \times \mathscr{T})}(\lambda_{\mathtt{p}})$$

    where $\Gamma_t = \emptyset$, $b_t = m_t = d_t = 0$. Thus, we can conclude $\Gamma_{t\{x\setminus u\}} = \Gamma_t + \Gamma_u = \emptyset$, $b = b_t + b_u = m = m_t + m_u = d = d_t + d_u = 0$.
  - Rule (@). Then, $t = rp$, $\mu = \mathscr{T} \Rightarrow \kappa$, and $t\{x \setminus u\} = (r\{x \setminus u\})(p\{x \setminus u\})$. Therefore, $\Phi_{t\{x\setminus u\}}$ has two premises $\Phi_{r\{x\setminus u\}} \rhd \Gamma_{r\{x\setminus u\}}; x : \mathscr{M}_1 \Vdash_{\mathtt{n}}^{(b_{r\{x\setminus u\}}, m_{r\{x\setminus u\}}, d_{r\{x\setminus u\}})} r\{x \setminus u\} : \mathscr{T} \Rightarrow ((\mathscr{M}' \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}')$ and $\Phi_{p\{x\setminus u\}} \rhd \Gamma_{p\{x\setminus u\}}; x : \mathscr{M}_2 \Vdash_{\mathtt{n}}^{(b_{p\{x\setminus u\}}, m_{p\{x\setminus u\}}, d_{p\{x\setminus u\}})} p\{x \setminus u\} : \mathscr{M}'$, where $\Gamma_{t\{x\setminus u\}} = \Gamma_{r\{x\setminus u\}} + \Gamma_{p\{x\setminus u\}}$, $b = 1 + b_{r\{x\setminus u\}} + b_{p\{x\setminus u\}}$, $m = m_{r\{x\setminus u\}} + m_{p\{x\setminus u\}}$, and $d = d_{r\{x\setminus u\}} + d_{p\{x\setminus u\}}$. By applying the *i.h.* to $\Phi_{r\{x\setminus u\}}$, there exist $\Phi_r \rhd \Gamma_r; x : \mathscr{M}_1 \Vdash_{\mathtt{n}}^{(b_r,m_r,d_r)} r : \mathscr{T} \Rightarrow ((\mathscr{M}' \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}')$ and $\Phi_u^1 \rhd \Gamma_u^1 \Vdash_{\mathtt{n}}^{(b_u^1,m_u^1,d_u^1)} u : \mathscr{M}_1$, where $\Gamma_{r\{x\setminus u\}} = \Gamma_r + \Gamma_u^1$, $b_{r\{x\setminus u\}} = b_r + b_u^1$, $m_{r\{x\setminus u\}} = m_r + m_u^1$, and $d_{r\{x\setminus u\}} = d_r + d_u^1$. By applying the *i.h.* to $\Phi_{p\{x\setminus u\}}$, there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{M}'$ and $\Phi_u^2 \rhd \Gamma_u^2 \Vdash_{\mathtt{n}}^{(b_u^2,m_u^2,d_u^2)} u : \mathscr{M}_2$, where $\Gamma_{p\{x\setminus u\}} = \Gamma_p + \Gamma_u^2$, $b_{p\{x\setminus u\}} = b_p + b_u^2$, $m_{p\{x\setminus u\}} = m_p + m_u^2$, and $d_{p\{x\setminus u\}} = d_p + d_u^2$. By Lemma 36, there exists $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}$, where $\mathscr{M} = \mathscr{M}_1 \sqcup \mathscr{M}_2$, $b_u = b_u^1 + b_u^2$, $m_u = m_u^1 + m_u^2$, and $d_u = d_u^1 + d_u^2$. Moreover, we can build Φ_t as follows:

$$\frac{\Phi_r \qquad \Phi_p}{(\Gamma_r + \Gamma_p); x : \mathscr{M} \Vdash^{(1+b_r+b_p, m_r+m_p, d_r+d_p)} rp : \mathscr{T} \Rightarrow \kappa}(@)$$

    where $\Gamma_t = \Gamma_r + \Gamma_p$, $b_t = 1 + b_r + b_p$, $m_t = m_r + m_p$, and $d_t = d_r + d_p$. Thus, we can conclude with $\Gamma_{t\{x\setminus u\}} = \Gamma_{r\{x\setminus u\}} + \Gamma_{p\{x\setminus u\}} = (\Gamma_r + \Gamma_u^1) + (\Gamma_p + \Gamma_u^2) = \Gamma_t + \Gamma_u$, $b = 1 + b_{r\{x\setminus u\}} + b_{p\{x\setminus u\}} = 1 + (b_r + b_u^1) + (b_p + b_u^2) = b_t + b_u$, $m = m_{r\{x\setminus u\}} + m_{p\{x\setminus u\}} =$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$(m_r + m_u^1) + (m_p + m_u^2) = m_t + m_u$, and $d = d_{r\{x\setminus u\}} + d_{p\{x\setminus u\}} = (d_r + d_u^1) + (d_p + d_u^2) = d_t + d_u$.

- Rule (get). Then $t = \mathtt{get}_l(\lambda y.p)$, $\mu = \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and $t\{x \setminus u\} = \mathtt{get}_l(\lambda y.(p\{x \setminus u\}))$. Therefore, $\Phi_{t\{x\setminus u\}}$ has one premise $\Phi_{p\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}}; y : \mathscr{M}' \vdash_{\mathtt{n}}^{(b_{p\{x\setminus u\}}, m_{p\{x\setminus u\}}, d_{p\{x\setminus u\}})} p\{x \setminus u\} : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, where $b = b_{p\{x\setminus u\}}$, $m = 1 + m_{p\{x\setminus u\}}$, and $d = d_{p\{x\setminus u\}}$. By applying the *i.h.* to $\Phi_{p\{x\setminus u\}}$, there exists $\Phi_p \rhd (\Gamma_p; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_p, m_p, d_p)} p : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$ and $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u, m_u, d_u)} u : \mathscr{M}$, where $\Gamma_{t\{x\setminus u\}}; y : \mathscr{M}' = (\Gamma_p; y : \mathscr{M}') + \Gamma_u$, $b_{p\{x\setminus u\}} = b_p + b_u$, $m_{p\{x\setminus u\}} = m_p + m_u$, and $d_{p\{x\setminus u\}} = d_p + d_u$. Therefore, $\Gamma_{t\{x\setminus u\}} = \Gamma_p + \Gamma_u$. Moreover, we can build Φ_t as follows:

$$\frac{\Phi_p}{\Gamma_p; x : \mathscr{M} \Vdash^{(b_p, 1+m_p, d_p)} \mathtt{get}_l(\lambda y.p) : \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{get})$$

where $\Gamma_t = \Gamma_p$, $b_t = b_p$, $m_t = m_p$, and $d_t = d_p$. Thus, we can conclude with $\Gamma_{t\{x\setminus u\}} = \Gamma_p + \Gamma_u$, $b = b_{p\{x\setminus u\}} = b_p + b_u$, $m = 1 + m_{p\{x\setminus u\}} = 1 + m_p + m_u$, and $d = d_{p\{x\setminus u\}} = d_p + d_u$.

- Rule (set). Then, $t = \mathtt{set}_l(p, r)$, $\mu = \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and $t\{x \setminus u\} = \mathtt{set}_l((p\{x \setminus u\}), (r\{x \setminus u\}))$. Therefore, $\Phi_{t\{x\setminus u\}}$ has two premises $\Phi_{p\{x\setminus u\}} \rhd \Gamma_{p\{x\setminus u\}} \Vdash_{\mathtt{n}}^{(b_{p\{x\setminus u\}}, m_{p\{x\setminus u\}}, d_{p\{x\setminus u\}})} p\{x \setminus u\} : \mathscr{M}'$ and $\Phi_{r\{x\setminus u\}} \rhd \Gamma_{r\{x\setminus u\}} \Vdash_{\mathtt{n}}^{(b_{r\{x\setminus u\}}, m_{r\{x\setminus u\}}, d_{r\{x\setminus u\}})} r\{x \setminus u\} : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, where $\Gamma_{t\{x\setminus u\}} = \Gamma_{p\{x\setminus u\}} + \Gamma_{r\{x\setminus u\}}$, $b = b_{p\{x\setminus u\}} + b_{r\{x\setminus u\}}$, $m = 1 + m_{p\{x\setminus u\}} + m_{r\{x\setminus u\}}$, and $d = d_{p\{x\setminus u\}} + d_{r\{x\setminus u\}}$. By applying the *i.h.* to $\Phi_{p\{x\setminus u\}}$, there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_1 \Vdash_{\mathtt{n}}^{(b_p, m_p, d_p)} p : \mathscr{M}'$ and $\Phi_u^1 \rhd \Gamma_u^1 \Vdash_{\mathtt{n}}^{(b_u^1, m_u^1, d_u^1)} u : \mathscr{M}_1$, where $\Gamma_{p\{x\setminus u\}} = \Gamma_p + \Gamma_u^1$, $b_{p\{x\setminus u\}} = b_p + b_u^1$, $m_{p\{x\setminus u\}} = m_p + m_u^1$, $d_{p\{x\setminus u\}} = d_p + d_u^1$. By applying the *i.h.* to $\Phi_{r\{x\setminus u\}}$, there exist $\Phi_r \rhd \Gamma_r; x : \mathscr{M}_2 \Vdash_{\mathtt{n}}^{(b_r, m_r, d_r)} r : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$ and $\Phi_u^2 \rhd \Gamma_u^2 \Vdash_{\mathtt{n}}^{(b_u^2, m_u^2, d_u^2)} u : \mathscr{M}_2$, where $\Gamma_{r\{x\setminus u\}} = \Gamma_r + \Gamma_u^2$, $b_{r\{x\setminus u\}} = b_r + b_u^2$, $m_{r\{x\setminus u\}} = m_r + m_u^2$, and $d_{r\{x\setminus u\}} = d_r + d_u^2$. By Lemma 36, there exists $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u, m_u, d_u)} u : \mathscr{M}$, where $b_u = b_u^1 + b_u^2$, $m_u = m_u^1 + m_u^2$, and $d_u = d_u^1 + d_u^2$. Moreover, we can build Φ_t as follows:

$$\frac{\Phi_p \qquad \Phi_r}{(\Gamma_p + \Gamma_r); x : \mathscr{M} \Vdash^{(b_p + b_r, 1 + m_p + m_r, d_p + d_r)} \mathtt{set}_l(p, r) : \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{set})$$

where $\Gamma_t = \Gamma_p + \Gamma_r$, $b_t = b_p + b_r$, $m_t = 1 + m_p + m_r$, and $d_t = d_p + d_r$. Thus, we can conclude with $\Gamma_{t\{x\setminus u\}} = \Gamma_{p\{x\setminus u\}} + \Gamma_{r\{x\setminus u\}} = (\Gamma_r + \Gamma_u^1) + (\Gamma_p + \Gamma_u^2) = \Gamma_t + \Gamma_u$, $b = b_{r\{x\setminus u\}} + b_{p\{x\setminus u\}} = (b_r + b_u^1) + (b_p + b_u^2) = b_t + b_u$, $m = 1 + m_{r\{x\setminus u\}} + m_{p\{x\setminus u\}} = 1 + (m_r + m_u^1) + (m_p + m_u^2) = m_t + m_u$, and $d = d_{r\{x\setminus u\}} + d_{p\{x\setminus u\}} = (d_r + d_u^1) + (d_p + d_u^2) = d_t + d_u$.

- Rule ($@_{\mathtt{p}}$). Then, $t = rp$, $\mu = \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, and $t\{x \setminus u\} = (r\{x \setminus u\})(p\{x \setminus u\})$. Therefore, $\Phi_{t\{x\setminus u\}}$ has one premise $\Phi_{r\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}} \Vdash_{\mathtt{n}}^{(b_{r\{x\setminus u\}}, m_{r\{x\setminus u\}}, d_{r\{x\setminus u\}})} r\{x \setminus u\} : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, $b = b_{r\{x\setminus u\}}$, $m = m_{r\{x\setminus u\}}$, and $d = 1 + d_{r\{x\setminus u\}}$. By applying the *i.h.* to $\Phi_{r\{x\setminus u\}}$, there exist $\Phi_r \rhd \Gamma_r; x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_r, m_r, d_r)} r : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$ and $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u, m_u, d_u)} u : \mathscr{M}$, where $\Gamma_{t\{x\setminus u\}} = \Gamma_r + \Gamma_u$, $b_{r\{x\setminus u\}} = b_r + b_u$, $m_{r\{x\setminus u\}} = m_r + m_r$, and $d_{r\{x\setminus u\}} = d_r + d_u$. Therefore, we can build Φ_t as follows:

$$\frac{\Phi_r}{\Gamma_r; x : \mathscr{M} \Vdash^{(b_r, m_r, 1+d_r)} r : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}(@_{\mathtt{p}})$$

where $b_t = b_r$, $m_t = m_r$, and $d_t = 1 + d_r$. Thus, we can conclude with $\Gamma_{t\{x\setminus u\}} = \Gamma_r + \Gamma_u$, $b = b_{r\{x\setminus u\}} = b_r + b_u = b_r + b_u$, $m = m_{r\{x\setminus u\}} = m_r + m_u = m_t + m_u$, and $d = 1 + d_{r\{x\setminus u\}} = 1 + d_r + d_u = d_t + d_u$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case $\mathscr{G} = \mathscr{M}_0 = [\delta_i]_{i\in I}$. Let us assume $t = x$, such that $t\{x \setminus u\} = u$. Then, $\Phi_{t\{x\setminus u\}} \rhd \Gamma_{t\{x\setminus u\}} \Vdash_{\mathtt{n}}^{(b,m,d)} u : [\delta_i]_{i\in I}$ and we pick $\Phi_u = \Phi_{t\{x\setminus u\}}$, where $\Gamma_u = \Gamma_{t\{x\setminus u\}}$, $b_u = b$ and $m_u = m$, $d_u = d$, and build Φ_t as follows:

$$\frac{\left(\dfrac{}{x : [\delta_i] \Vdash^{(0,0,0)} x : \delta_i}\,(\mathtt{ax})\right)_{i\in I}}{x : [\delta_i]_{i\in I} \Vdash^{(0,0,0)} x : [\delta_i]_{i\in I}}\,(\mathtt{m})$$

  where $\Gamma_t = \emptyset$, and $b_t = m_t = d_t = 0$. In both cases, we can conclude with $\Gamma_{t\{x\setminus u\}} = \Gamma_u = \Gamma_t + \Gamma_u$, $b = b_u = b_t + b_u$, $m = m_u = m_t + m_u$, and $d = d_u = d_t + d_u$. From now on, we are going to assume $t \neq x$ and consider the last rule used in $\Phi_{t\{x\setminus u\}}$, which must be (m):

  – Rule (m). Then, $\mathscr{M}_0 = [\delta_i]_{i\in I}$ and $\Phi_{t\{x\setminus u\}}$ has premises $(\Phi^i_{t\{x\setminus u\}} \rhd \Gamma^i_{t\{x\setminus u\}} \Vdash_{\mathtt{n}}^{(b_i,m_i,d_i)} t\{x \setminus u\} : \delta_i)_{i\in I}$, where $\Gamma_{t\{x\setminus u\}} = +_{i\in I}\Gamma^i_{t\{x\setminus u\}}$, $b = +_{i\in I} b_i$, $m = +_{i\in I} m_i$, and $d = +_{i\in I} d_i$. By applying the *i.h.* to each $\Phi^i_{t\{x\setminus u\}}$, there exist $(\Phi^i_t \rhd \Gamma^i_t; x : \mathscr{M}_i \Vdash_{\mathtt{n}}^{(b^i_t,m^i_t,d^i_t)} t : \delta_i)_{i\in I}$ and $(\Phi^i_u \rhd \Gamma^i_u \Vdash_{\mathtt{n}}^{(b^i_u,m^i_u,d^i_u)} u : \mathscr{M}_i)_{i\in I}$, where $\Gamma^i_{t\{x\setminus u\}} = \Gamma^i_t + \Gamma^i_u$, $b_i = b^i_t + b^i_u$, $m_i = m^i_t + m^i_u$, and $d_i = d^i_t + d^i_u$, for each $i \in I$. By Lemma 36, there exists $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{n}}^{(b_u,m_u,d_u)} u : \mathscr{M}$, where $\Gamma_u = +_{i\in I}\Gamma^i_u$, $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b_u = +_{i\in I} b^i_u$, $m_u = +_{i\in I} m^i_u$, and $d_u = +_{i\in I} d^i_u$. Moreover, we can build Φ_t as follows:

$$\frac{(\Phi^i_t \rhd \Gamma^i_t; x : \mathscr{M}_i \Vdash_{\mathtt{n}}^{(b^i_t,m^i_t,d^i_t)} t : \delta_i)_{i\in I}}{+_{i\in I}\Gamma^i_t; x : \sqcup_{i\in I}\mathscr{M}_i \Vdash^{(+_{i\in I}b_i,+_{i\in I}m_i,+_{i\in I}d_i)} t : [\delta_i]_{i\in I}}\,(\mathtt{m})$$

  where $\Gamma_t = +_{i\in I}\Gamma^i_t$, $b_t = +_{i\in I} b^i_t$, $m_t = +_{i\in I} m^i_t$, and $d_t = +_{i\in I} d^i_t$. Thus, we can conclude with $\Gamma_{t\{x\setminus u\}} = +_{i\in I}\Gamma^i_{t\{x\setminus u\}} = +_{i\in I}(\Gamma^i_t + \Gamma^i_u) = +_{i\in I}\Gamma^i_t +_{i\in I} \Gamma^i_u = \Gamma_t + \Gamma_u$, $b = +_{i\in I} b_i = +_{i\in I}(b^i_t + b^i_u) = +_{i\in I} b^i_t +_{i\in I} b^i_u = b_t + b_u$, $m = +_{i\in I} m_i = +_{i\in I}(m^i_t + m^i_u) = +_{i\in I} m^i_t +_{i\in I} m^i_u = m_t + m_u$, $d = +_{i\in I} d_i = +_{i\in I}(d^i_t + d^i_u) = +_{i\in I} d^i_t +_{i\in I} d^i_u = d_t + d_u$. □

Lemma 14 (Quantitative subject expansion). *Let* $(t, s) \rightsquigarrow_{\mathtt{r}} (u, q)$, *where* $\mathtt{r} \in \{\beta_{\mathtt{n}}, \mathtt{g}, \mathtt{s}\}$. *If* $\Phi' \rhd \Gamma \Vdash_{\mathtt{n}}^{(b',m',d')} (u, q) : \kappa$ *is a tight derivation, then* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d')} (t, s) : \kappa$, *where* $\mathtt{r} = \beta_{\mathtt{n}}$ *implies* $b' = b - 1$ *and* $m' = m$, *while* $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ *implies* $b' = b$ *and* $m' = m - 1$.

Proof. We show a stronger statement:

Let $(t, s) \rightsquigarrow_{\mathtt{r}} (u, q)$, where $\mathtt{r} \in \{\beta_{\mathtt{n}}, \mathtt{g}, \mathtt{s}\}$. If $\Phi' \rhd \Gamma \Vdash_{\mathtt{n}}^{(b',m',d')} (u, q) : \kappa$, Γ is tight, and κ is tight or $\neg\mathtt{abs}(t)$, then $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,d')} (t, s) : \kappa$, where $\mathtt{r} = \beta_{\mathtt{n}}$ implies $b' = b - 1$ and $m' = m$, while $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ implies $b' = b$ and $m' = m - 1$.

We proceed by induction on $(t, s) \rightsquigarrow_{\mathtt{n}} (u, q)$:

- Case $(t, s) = ((\lambda x.p)r, s) \rightsquigarrow_{\beta_{\mathtt{n}}} (p\{x \setminus r\}, s) = (u, q)$. Then Φ' has two premises $\Phi_{p\{x\setminus r\}} \rhd \Gamma_{p\{x\setminus r\}} \Vdash_{\mathtt{n}}^{(b_{p\{x\setminus r\}},m_{p\{x\setminus u\}},d_{p\{x\setminus r\}})} p\{x \setminus r\} : \mathscr{T} \Rightarrow \kappa$ and $\Pi_s \rhd \Delta_s \Vdash_{\mathtt{n}}^{(b_s,m_s,d_s)} s : \mathscr{T}$, such that $\Gamma = \Gamma_{p\{x\setminus r\}} + \Delta_s$, $b' = b_{p\{x\setminus r\}} + b_s$, $m' = m_{p\{x\setminus r\}} + m_s$, and $d' = d_{p\{x\setminus r\}} + d_s$. By Lemma 39, there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M} \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \mathscr{T} \Rightarrow \kappa$ and $\Phi_r \rhd \Gamma_r \Vdash_{\mathtt{n}}^{(b_r,m_r,d_r)} r : \mathscr{M}$, such that $\Gamma_{p\{x\setminus r\}} = \Gamma_p + \Gamma_r$, $b_{p\{\setminus r\}} = b_p + b_r$, $m_{p\{x\setminus u\}} = m_p + m_r$, and $d_{p\{x\setminus u\}} = d_p + d_r$. Therefore, we can build Φ as follows:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\dfrac{\dfrac{\Phi_p}{\Gamma_p \Vdash^{(b_p,m_p,d_p)} \lambda x.p : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T} \Rightarrow \kappa)) \times \mathscr{T})}\,(\lambda) \qquad \Phi_r}{\dfrac{\Gamma_p + \Gamma_r \Vdash^{(1+b_p+b_r,m_p+m_r,d_p+d_r)} (\lambda x.p)r : \mathscr{T} \Rightarrow \kappa}{(\Gamma_p + \Gamma_r) + \Delta_s \Vdash^{(1+b_p+b_r+b_s,m_p+m_r+m_s,d_p+d_r+d_s)} ((\lambda x.p)r, s) : \kappa}\,(@) \qquad \Phi_s}\,(\texttt{conf})$$

such that $b = 1 + b_p + b_r + b_s$, $m = m_p + m_r + m_s$, and $d = d_p + d_r + d_s$. Thus, we can conclude with $\Gamma = \Gamma_{p\{x\backslash r\}} + \Delta_s = (\Gamma_p + \Gamma_r) + \Delta_s$, $b' = b_{p\{x\backslash r\}} + b_s = b_p + b_r + b_s = (1 + b_p + b_r + b_s) - 1 = b - 1$, $m' = m_{p\{x\backslash u\}} + m_s = (m_p + m_r) + m_s = m$, and $d' = d_{p\{x\backslash u\}} + d_s = (d_p + d_r) + d_s = d$.

- Case $(t, s) = (\texttt{get}_l(\lambda x.p), s) \rightsquigarrow_{\mathtt{n}} (p\{x \setminus r\}, s) = (u, q)$, such that $q \equiv \texttt{upd}_l(r, s') \equiv s$. Then, Φ' has two premises $\Phi_u \rhd \Gamma_{p\{x\backslash r\}} \Vdash_{\mathtt{n}}^{(b_{p\{x\backslash r\}},m_{p\{x\backslash r\}},d_{p\{x\backslash r\}})} p\{x \setminus r\} : \{l : \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$ and Π_q, which must be of the following form:

$$\dfrac{\Phi_r^2 \rhd \Gamma_r^2 \Vdash_{\mathtt{n}}^{(b_r^2,m_r^2,d_r^2)} r : \mathscr{M}_2 \qquad \Pi_{s'} \rhd \Delta_{s'} \Vdash_{\mathtt{n}}^{(b_{s'},m_{s'},d_{s'})} s' : \{l : \mathscr{L}\}; \mathscr{T}}{\Gamma_r^2 + \Delta_{s'} \Vdash^{(b_r^2+b_{s'},m_r^2+m_{s'},d_r^2+d_{s'})} \texttt{upd}_l(r, s') : \{l : \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T}}\,(\texttt{upd})$$

where $\Gamma = \Gamma_{p\{x\backslash r\}} + (\Gamma_r^2 + \Delta_{s'})$, $b' = b_{p\{x\backslash r\}} + b_r^2 + b_{s'}$, $m' = m_{p\{x\backslash r\}} + b_r^2 + b_{s'}$, and $d' = d_{p\{x\backslash r\}} + d_r^2 + d_{s'}$. By Lemma 39, there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_1 \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \{l : \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$ and $\Phi_r^1 \rhd \Gamma_r^1 \Vdash_{\mathtt{n}}^{(b_r^1,m_r^1,d_r^1)} r : \mathscr{M}_1$, such that $\Gamma_{p\{x\backslash r\}} = \Gamma_p + \Gamma_r^1$, $b_{p\{x\backslash r\}} = b_p + b_r^1$, $m_{p\{x\backslash r\}} = m_p + m_r^1$, and $d_{p\{x\backslash u\}} = d_p + d_r^1$. Therefore, we can build Φ_t as follows:

$$\dfrac{\Phi_p \rhd \Gamma_p; x : \mathscr{M}_1 \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \{l : \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}{\Gamma_p \Vdash^{(b_p,1+m_p,d_p)} \texttt{get}_l(\lambda x.p) : \{l : \mathscr{M}_1 \sqcup \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}\,(\texttt{get})$$

By Lemma 36, there exists $\Phi_r \rhd \Gamma_r^1 + \Gamma_r^2 \Vdash_{\mathtt{n}}^{(b_r^1+b_r^2,m_r^1+m_r^2,d_r^1+d_r^2)} r : \mathscr{M}_1 \sqcup \mathscr{M}_2$. Therefore, we can build an alternative derivation Π'_q for q as follows:

$$\dfrac{\Phi_r \qquad \Pi_{s'}}{(\Gamma_r^1 + \Gamma_r^2) + \Delta_{s'} \Vdash^{(b_r^1+b_r^2+b_{s'},m_r^1+m_r^2+m_{s'},d_r^1+d_r^2+d_{s'})} \texttt{upd}_l(r, s') : \{l : \mathscr{M}_1 \sqcup \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T}}\,(\texttt{upd})$$

Then, we can pick $\Pi_s = \Pi'_q$, and build Φ as follows:

$$\dfrac{\Phi_t \qquad \Pi_s}{\Gamma_p + (\Gamma_r^1 + \Gamma_r^2) + \Delta_{s'} \Vdash^{(b_p+b_r^1+b_r^2+b_{s'},1+m_p+m_r^1+m_r^2+m_{s'},d_p+d_r^1+d_r^2+d_{s'})} (\texttt{get}_l(\lambda x.p), s) : \kappa}\,(\texttt{conf})$$

where $b = b_p + b_r^1 + b_r^2 + b_{s'}$, $m = 1 + m_p + m_r^1 + m_r^2 + m_{s'}$, and $d = d_p + d_r^1 + d_r^2 + d_{s'}$. Thus, we can conclude with $\Gamma = \Gamma_{p\{x\backslash r\}} + (\Gamma_r^2 + \Delta_{s'}) = \Gamma_p + \Gamma_r^1 + \Gamma_r^2 + \Delta_{s'}$, $b' = b_{p\{x\backslash r\}} + b_r^2 + b_{s'} = b_p + b_r^1 + b_r^2 + b_{s'} = b$, and $m' = m_{p\{x\backslash r\}} + m_r^2 + m_{s'} = m_p + m_r^1 + m_r^2 + m_{s'} = (1 + m_p + m_r^1 + m_r^2 + m_{s'}) - 1 = m - 1$, $d' = d_{p\{x\backslash r\}} + d_r^2 + d_{s'} = d_p + d_r^1 + d_r^2 + d_{s'} = d$.

- Case $(t, s) = (\texttt{set}_l(r, p), s) \rightsquigarrow_{\mathtt{n}} (p, \texttt{upd}_l(r, s)) = (u, q)$. Then, Φ' has two premises $\Phi_p \rhd \Gamma_p \Vdash_{\mathtt{n}}^{(b_p,m_p,d_p)} p : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$ and Π_q, which must be of the following form:

$$\dfrac{\Phi_r \rhd \Gamma_r \Vdash_{\mathtt{n}}^{(b_r,m_r,d_r)} r : \mathscr{M} \qquad \Pi_s \rhd \Delta_s \Vdash_{\mathtt{n}}^{(b_s,m_s,d_s)} s : \{l : \mathscr{L}\}; \mathscr{T}}{\Gamma_r + \Delta_s \Vdash^{(b_r+b_s,m_r+m_s,d_r+d_s)} \texttt{upd}_l(r, s) : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}}\,(\texttt{upd})$$

where $\Gamma = \Gamma_p + (\Gamma_r + \Delta_s)$, $b' = b_p + b_r + b_s$, $m' = m_p + m_r + m_s$, and $d' = d_p + d_r + d_s$. Therefore, we can build Φ as follows:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$$\dfrac{\dfrac{\Phi_r \qquad \Phi_p}{\Gamma_r + \Gamma_p \Vdash^{(b_r+b_p,1+m_r+m_p,d_r+d_p)} \mathtt{set}_l(r,p) : \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}\,(\mathtt{set}) \qquad \Pi_s}{(\Gamma_r + \Gamma_p) + \Delta_s \Vdash^{(b_r+b_p+b_s,1+m_r+m_p+m_s,d_r+d_p+d_s)} (\mathtt{set}_l(r,p), s) : \kappa}\,(\mathtt{conf})$$

where $b = b_r + b_p + b_s$, $m = 1 + m_r + m_p + m_s$, and $d = d_r + d_p + d_s$. Thus, we can conclude with $\Gamma = \Gamma_p + (\Gamma_r + \Delta_s)$, $b' = b_r + b_p + b_s = b$, $m' + 1 = (m_p + m_r + m_s) + 1 = m$, and $d' = d_r + d_p + d_s = d$.

- Case $(t, s) = (pr, s) \rightsquigarrow_{\mathtt{n}} (p'r, q) = (u, q)$, such that $(p, s) \rightsquigarrow_{\mathtt{n}} (p', q)$. Then, Φ' has two premises $\Phi_u \triangleright \Gamma_u \Vdash^{(b_u,m_u,d_u)}_{\mathtt{n}} u : \mathscr{T}' \Rightarrow \kappa$ and $\Pi_q \triangleright \Delta_q \Vdash^{(b_q,m_q,d_q)}_{\mathtt{n}} q : \mathscr{T}'$, such that $\Gamma = \Gamma_u + \Delta_q$, $b' = b_u + b_q$, $m' = m_u + m_q$, and $d' = d_u + d_q$. Moreover, Φ_u must end with rule ($@$) or ($@_{\mathtt{p}}$):
  - Case Φ_u ends with ($@$). Then, Φ_u must be of the following form:

$$\dfrac{\begin{array}{c}\Phi_{p'} \triangleright \Gamma_{p'} \Vdash^{(b_{p'},m_{p'},d_{p'})}_{\mathtt{n}} p' : \mathscr{T}' \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}'') \\ \Phi_r \triangleright \Gamma_r \Vdash^{(b_r,m_r,d_r)}_{\mathtt{n}} r : \mathscr{M}\end{array}}{\Gamma_{p'} + \Gamma_r \Vdash^{(1+b_{p'}+b_r,m_{p'}+m_r,d_{p'}+d_r)} p'r : \mathscr{T}' \Rightarrow \kappa}\,(@)$$

    where $\Gamma_u = \Gamma_{p'} + \Gamma_r$, $b_u = 1 + b_{p'} + b_r$, $m_u = m_{p'} + m_r$, and $d_u = d_{p'} + d_r$. Therefore, we can build derivation $\Phi_{(p',q)}$ as follows:

$$\dfrac{\Phi_{p'} \qquad \Pi_q}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'}+b_q,m_{p'}+m_q,d_{p'}+d_q)} (p', q) : (\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}''}\,(\mathtt{conf})$$

    Notice that $\Gamma_{p'}$ and Δ_q are tight, and $(p, s) \rightsquigarrow_{\mathtt{n}} (p', q)$ implies $\neg\mathtt{abs}(p)$. By applying the *i.h.* to $\Phi_{(p',q)}$, there exists $\Phi_{(p,s)}$ that must be of the following form:

$$\dfrac{\begin{array}{c}\Phi_p \triangleright \Gamma_p \Vdash^{(b_p,m_p,d_p)}_{\mathtt{n}} p : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}'') \\ \Pi_s \triangleright \Delta_s \Vdash^{(b_s,m_s,d_s)}_{\mathtt{n}} s : \mathscr{T}\end{array}}{\Gamma_p + \Delta_s \Vdash^{(b_p+b_s,m_p+m_s,d_p+d_s)} (p, s) : (\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}''}\,(\mathtt{conf})$$

    where $\Gamma_p + \Delta_s = \Gamma_{p'} + \Delta_q$, and $(b_p + b_s, m_p + m_s, d_p + d_s)$ and $(b_{p'} + b_q, m_{p'} + m_q, d_{p'} + d_q)$ are properly related. Therefore, we can build Φ as follows:

$$\dfrac{\dfrac{\Phi_p \qquad \Phi_r}{\Gamma_r + \Gamma_p \Vdash^{(1+b_p+b_r,m_p+m_r,d_p+d_r)} pr : \mathscr{T} \Rightarrow \kappa}\,(@) \qquad \Pi_s}{(\Gamma_r + \Gamma_p) + \Delta_s \Vdash^{(1+b_p+b_r+b_s,m_p+m_r+m_s,d_p+d_r+d_s)} (pr, s) : \kappa}\,(\mathtt{conf})$$

    where $\Gamma = \Gamma_{p'} + \Gamma_r + \Delta_q = \Gamma_p + \Gamma_r + \Delta_s$, and $(1 + b_p + b_r + b_s, m_p + m_r + m_s, d_p + d_r + d_s)$ and $(1 + b_{p'} + b_r + b_q, m_{p'} + m_r + m_q, d_{p'} + d_r + d_q)$ are properly related.
  - Case $\Phi_{p'r}$ ends with ($@_{\mathtt{p}}$). Then, Φ' must be of the following form:

$$\dfrac{\dfrac{\Phi_{p'} \triangleright \Gamma_{p'} \Vdash^{(b_{p'},m_{p'},d_{p'})}_{\mathtt{n}} p' : \mathscr{T}' \Rightarrow (\mathtt{n} \times \mathscr{T}'')}{\Gamma_{p'} \Vdash^{(b_{p'},m_{p'},1+d_{p'})} p'r : \mathscr{T}' \Rightarrow (\mathtt{n} \times \mathscr{T}'')}\,(@_{\mathtt{p}}) \qquad \Pi_q \triangleright \Delta_q \Vdash^{(b_q,m_q,d_q)}_{\mathtt{n}} q : \mathscr{T}'}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'}+b_q,m_{p'}+m_q,1+d_{p'}+d_q)} (p'r, q) : \mathtt{n} \times \mathscr{T}''}\,(\mathtt{conf})$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

where $\Gamma = \Gamma_{p'} + \Delta_q$ tight, $b' = b_{p'} + b_q$, $m' = m_{p'} + m_q$, and $d' = 1 + d_{p'} + d_q$. Then, we can build derivation $\Phi_{(p',q)}$ as follows:

$$\frac{\Phi_{p'} \qquad \Pi_q}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'}+b_q, m_{p'}+m_q, 1+d_{p'}+d_q)} (p', q) : \mathtt{n} \times \mathscr{T}''}(\mathtt{conf})$$

Moreover, $\Gamma_{p'}$ and Δ_q are tight, and $(p, s) \rightsquigarrow_{\mathtt{n}} (p', q)$ implies $\neg\mathtt{abs}(p)$. By applying the *i.h.* to $\Phi_{(p',q)}$, there exists a derivation for $\Phi_{(p,s)}$ for (p, s) that must be of the following form:

$$\frac{\Phi_p \rhd \Gamma_p \Vdash_{\mathtt{n}}^{(b_p, m_p, d_p)} p : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}'') \qquad \Pi_s \rhd \Delta_s \Vdash_{\mathtt{n}}^{(b_s, m_s, d_s)} s : \mathscr{T}}{\Gamma_p + \Delta_s \Vdash^{(b_p+b_s, m_p+m_s, d_p+d_s)} (p, s) : \mathtt{n} \times \mathscr{T}''}(\mathtt{conf})$$

where $\Gamma_p + \Delta_s = \Gamma_{p'} + \Delta_q$, $(b_p + b_s, m_p + m_s, d_p + d_s)$ and $(b_{p'} + b_q, m_{p'} + m_q, 1 + d_{p'} + d_q)$ are properly related. Therefore, we can build Φ as follows:

$$\dfrac{\dfrac{\Phi_p}{\Gamma_p \Vdash^{(b_p, m_p, 1+d_p)} pr : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}'')}(@_{\mathtt{p}}) \qquad \Pi_s}{\Gamma_p + \Delta_s \Vdash^{(b_p+b_s, m_p+m_s, 1+d_p+d_s)} (pr, s) : \mathtt{n} \times \mathscr{T}''}(\mathtt{conf})$$

where $b = b_p + b_s$, $m = m_p + m_s$, and $d = 1 + d_p + d_s$. Thus, we can conclude with $\Gamma = \Gamma_{p'} + \Delta_q = \Gamma_p + \Delta_s$, and $(b_p + b_s, m_p + m_s, 1 + d_p + d_s)$ and $(b_{p'} + b_q, m_{p'} + m_q, d_{p'} + d_q)$ are properly related. □

Theorem 4 (Quantitative completeness). *If* $(t, s) \rightsquigarrow_{\mathtt{n}}^{(b,m)} (u, q)$ *and* $u \in \mathtt{no}_{\mathtt{n}}$, *then there exists a tight derivation* $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b,m,|(u,q)|_{\mathtt{n}})} (t, s) : \kappa$.

Proof. By induction on $b + m$:

- Case $b + m = 0$. Then, $b = m = 0$, and thus, $(t, s) = (u, q)$. Therefore, $t \in \mathtt{no}$. Thus, we can conclude by Lemma 12.
- Case $b + m > 0$. Then, there exists (t', s'), such that $(t, s) \rightsquigarrow_{\mathtt{n}}^{(1,0)} (t', s') \rightsquigarrow_{\mathtt{n}}^{(b-1,m)} (u, q)$ or $(t, s) \rightsquigarrow_{\mathtt{n}}^{(0,1)} (t', s') \rightsquigarrow_{\mathtt{n}}^{(b,m-1)} (u, q)$. By the *i.h.*, there exists $\Phi' \rhd \Gamma \Vdash_{\mathtt{n}}^{(b', m', |(u,q)|_{\mathtt{n}})} (t', s') : \kappa$ tight, such that $b' + m' = b + m - 1$. By Lemma 14, we have $\Phi \rhd \Gamma \Vdash_{\mathtt{n}}^{(b'', m'', |(u,q)|_{\mathtt{n}})} (t, s) : \kappa$ tight, such that $b'' + m'' = 1 + b' + m'$. Therefore, $b'' + m'' = b + m$, since the fact that $b'' = b$, and $m'' = m$ can be easily checked by a simple case analysis. □

A.3 Weak Open CBV with Global State

Proposition 5. *Let* (t, s) *be a configuration. Then* (t, s) *is normal iff* $(t, s) \not\rightsquigarrow_{\mathtt{v}}$.

Proof.

⇒) Let (t, s) be normal. Then, either (t, s) is blocked or $t \in \mathtt{no}_{\mathtt{v}}$:

- Case (t, s) is blocked. We reason by induction on (t, s).
  - Case $t = \mathtt{get}_l(\lambda x.u)$, such that $l \notin \mathtt{labels}(s)$. Then $(t, s) \not\rightsquigarrow_{\mathtt{v}}$ by definition.
  - Case $(t, s) = (up, s)$, such that (u, s) is blocked. Then, by applying the *i.h.* to (u, s), we get $(u, s) \not\rightsquigarrow_{\mathtt{v}}$. Moreover, since (u, s) is blocked, then necessarily $\neg\mathtt{val}(u)$. Therefore,

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

rule $(\beta_{\mathtt{v}})$ does not apply since $\neg\mathtt{val}(u)$, rule (appL) does not apply since $(u,s) \not\rightsquigarrow_{\mathtt{v}}$, rule (appR) does not apply since (u,s) is blocked, and rules (get) and (set) do not apply since t is an application. Thus, $(up,s) \not\rightsquigarrow_{\mathtt{v}}$.

* Case $(t,s) = (up,s)$, such that $(u,s) \not\rightsquigarrow_{\mathtt{v}}$, (u,s) is unblocked, and (p,s) is blocked. Then, by applying the *i.h.* to (p,s), we get $(p,s) \not\rightsquigarrow_{\mathtt{v}}$. Moreover, since (p,s) is blocked, then necessarily $p \neq v$. Therefore, rule $(\beta_{\mathtt{v}})$ does not apply since $p \neq v$, rule (appL) does not apply since $(u,s) \not\rightsquigarrow_{\mathtt{v}}$, rule (appR) does not apply since $(p,s) \not\rightsquigarrow_{\mathtt{v}}$, and rules (get) and (set) do not apply since t is an application. Thus, no rule applies to (up,s), that is, $(up,s) \not\rightsquigarrow_{\mathtt{v}}$.

– Case $t \in \mathtt{no}_{\mathtt{v}}$. Then, (t,s) is unblocked by definition. We reason by induction on $t \in \mathtt{no}_{\mathtt{v}}$.

* Case $\mathtt{val}(t)$. Then $(t,s) \not\rightsquigarrow_{\mathtt{v}}$ by definition.
* Case $\neg\mathtt{val}(t)$. Then necessarily $t \in \mathtt{ne}_{\mathtt{v}}$. There are three cases:

· Case $t = xp$, such that $p \in \mathtt{no}_{\mathtt{v}}$. By applying the *i.h.* to (p,s), we get $(p,s) \not\rightsquigarrow_{\mathtt{v}}$. Therefore, no rule of $\rightsquigarrow_{\mathtt{v}}$ applies to (xp,s), that is, $(xp,s) \not\rightsquigarrow_{\mathtt{v}}$.
· Case $t = up$, such that $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}} \subseteq \mathtt{no}_{\mathtt{v}}$. Moreover, $p \neq v$, and (u,s) and (p,s) are unblocked. By applying the *i.h.* to (u,s) and (p,s), we get $(u,s) \not\rightsquigarrow_{\mathtt{v}}$ and $(p,s) \not\rightsquigarrow_{\mathtt{v}}$, respectively. Therefore, rule $(\beta_{\mathtt{v}})$ does not apply since $p \neq v$, rule (appL) does not apply since $(u,s) \not\rightsquigarrow_{\mathtt{v}}$, rule (appR) does not apply since $(p,s) \not\rightsquigarrow_{\mathtt{v}}$, and rules (get) and (set) do not apply since t is an application. Thus, $(up,s) \not\rightsquigarrow_{\mathtt{v}}$.
· Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{v}}$ and $p \in \mathtt{no}_{\mathtt{v}}$. Moreover, $\neg\mathtt{val}(u)$ and (u,s) and (p,s) are unblocked. By applying the *i.h.* to (u,s) and (p,s), we get $(u,s) \not\rightsquigarrow_{\mathtt{v}}$ and $(p,s) \not\rightsquigarrow_{\mathtt{v}}$, respectively. Therefore, rule $(\beta_{\mathtt{v}})$ does not apply since $\neg\mathtt{val}(u)$, rule (appL) does not apply since $(u,s) \not\rightsquigarrow_{\mathtt{v}}$, rule (appR) does not apply since $(p,s) \not\rightsquigarrow_{\mathtt{v}}$, and rules (get) and (set) do not apply since t is an application. Thus, $(up,s) \not\rightsquigarrow_{\mathtt{v}}$.

$\Leftarrow$) Let $(t,s) \not\rightsquigarrow_{\mathtt{v}}$. We reason by induction on t:

– Case $\mathtt{val}(t)$. Then, $t \in \mathtt{no}_{\mathtt{v}}$ by definition. Therefore, (t,s) is normal.
– Case $t = up$:

* Case $u = \lambda x.r$ and $\mathtt{val}(p)$. Then, rule $(\beta_{\mathtt{v}})$ is applicable to (up,s), which leads to a contradiction with the hypothesis. Therefore, this case does not apply.
* Case $u = \lambda x.r$ and $p \neq v$. Since $(\lambda x.r,s) \not\rightsquigarrow_{\mathtt{v}}$ and $(\lambda x.r,s)$ is unblocked by definition, therefore, $(p,s) \not\rightsquigarrow_{\mathtt{v}}$; otherwise, rule (appR) would be applicable to (up,s). We can then apply the *i.h.* to p, which implies that (p,s) is normal. Therefore, either (p,s) is blocked, or $p \in \mathtt{no}_{\mathtt{v}}$. Let us first assume (p,s) is blocked. Then, (up,s) is blocked, and thus, it is normal. Let us now assume $p \in \mathtt{no}_{\mathtt{v}}$. Since $p \neq v$ then $p \in \mathtt{ne}_{\mathtt{v}}$, and $\mathtt{val}(u)$, and thus, $u \subseteq \mathtt{no}_{\mathtt{v}}$. Therefore, $up \in \mathtt{ne}_{\mathtt{v}} \subseteq \mathtt{no}_{\mathtt{v}}$. Thus, (up,s) is normal.
* Case $\neg\mathtt{abs}(u)$. Then, $(u,s) \not\rightsquigarrow_{\mathtt{v}}$, otherwise rule (appL) would be applicable to (up,s). We can then apply the *i.h.* to u, which implies that (u,s) is normal. Therefore, either (u,s) is blocked, or $u \in \mathtt{no}_{\mathtt{v}}$. Let us first assume (u,s) is blocked. Then, (up,s) is blocked by definition, and thus, it is normal. Let us now assume $u \in \mathtt{no}_{\mathtt{v}}$. Therefore, (u,s) is unblocked. As a consequence, $(p,s) \not\rightsquigarrow_{\mathtt{v}}$, otherwise rule (appR) would be applicable to (up,s). We can then apply the *i.h.* to p, which implies that (p,s) is normal. Therefore, either (p,s) is blocked, or $p \in \mathtt{no}_{\mathtt{v}}$. In the former case, (up,s) is blocked by definition, and thus, it is normal. In the latter case, $up \in \mathtt{ne}_{\mathtt{v}} \subseteq \mathtt{no}_{\mathtt{v}}$. Thus, (up,s) is normal. □

Lemma 40 (Relevance). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mu$ *(resp.* $\Pi \triangleright \Delta \Vdash_{\mathtt{v}}^{(b',m',d')} s : \mathscr{T}$*). Then* $\mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(t)$ *(resp.* $\mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(s)$*).*

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Proof. By a simple induction over Φ (resp. Π). □

Lemma 41 (Tight spreading). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, *such that* Γ *is tight. If* $t \in \mathtt{ne}_{\mathtt{v}}$, *then* $\tau = \mathtt{n}$.

Proof. We proceed by induction on $t \in \mathtt{ne}_{\mathtt{v}}$:

(1) Case $t = xu$, with $u \in \mathtt{no}_{\mathtt{v}}$.
(2) Case $t = up$, with $u \in \mathtt{ne}_{\mathtt{v}}$ and $p \in \mathtt{no}_{\mathtt{v}}$.
(3) Case or $t = up$, with $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}}$.

In all the above cases, Φ must end with rule ($@$), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):

- Case Φ ends with rule ($@$). Then, Φ has two premises $\Phi_l \triangleright \Gamma_l \vdash^{(b_l,m_l,d_l)} t_l : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}''' \Rightarrow \kappa)) \times \mathscr{T}'')$ and $\Phi_r \triangleright \Gamma_r \vdash^{(b_r,m_r,d_r)} t_r : \mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}''')$, such that $\Gamma = \Gamma_l + \Gamma_r$ is tight, $\kappa = \tau \times \mathscr{T}'$, $b = 1 + b_l + b_r$, $m = m_l + m_r$, and $d = d_l + d_r$:
  (1) $t_l = x$, $t_r = u$, and $\Gamma_l = (x : [\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)])$. However, $\mathscr{M} \to (\mathscr{T}''' \Rightarrow \kappa)$ is not tight, which leads to a contradiction with Γ tight. Therefore, this case does not apply.
  (2) $t_l = u$ and $t_r = p$. By *i.h.* on $u \in \mathtt{ne}_{\mathtt{v}}$ we get $\mathscr{M} \to (\mathscr{T}''' \Rightarrow \kappa) = \mathtt{n}$, which is a contradiction. Therefore, this case does not apply either.
  (3) $t_l = u$ and $t_r = p$. By *i.h.* on $p \in \mathtt{ne}_{\mathtt{v}}$, we get $\mathscr{M} = \mathtt{n}$, which is a contradiction. Therefore, this case does not apply either.
- Case Φ ends with rule ($@^1_{\mathtt{p}}$) or ($@^2_{\mathtt{p}}$). Then $\tau = \mathtt{n}$ by definition.

□

Lemma 42. *Let* $\Pi \triangleright \Delta \Vdash_{\mathtt{v}}^{(b,m,d)} s : \mathscr{T}$. *If* $l \in \mathtt{dom}(\mathscr{T})$, *then* $l \in \mathtt{labels}(s)$.

Proof. We proceed by proving the following stronger version of the statement:
Let $\Pi \triangleright \Delta \Vdash_{\mathtt{v}}^{(b,m,d)} s : \mathscr{T}$. If $l \in \mathtt{dom}(\mathscr{T})$, then $s \equiv \mathtt{upd}_l(v, q)$, for some value v and store q.
The proof follows by induction on Π:

- Case Π ends with ($\mathtt{emp}$). Then $\mathscr{T} = \emptyset$ which contradicts $l \in \mathtt{dom}(\mathscr{T})$.
- Case Π ends with ($\mathtt{upd}$). Then Π is of the following form:

$$\frac{\Phi_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M} \qquad \Pi_q \triangleright \Delta_q \Vdash_{\mathtt{v}}^{(b_q,m_q,d_q)} q : \{l' : \mathscr{L}'\}; \mathscr{T}_q}{\Gamma_v + \Delta_q \Vdash^{(b_v+b_q,m_v+m_q,d_v+d_q)} \mathtt{upd}_{l'}(v, q) : \{l' : \mathscr{M} \cdot \mathscr{L}'\}; \mathscr{T}_q}(\mathtt{upd})$$

  where $\Delta = \Gamma_v + \Delta_q$, $s = \mathtt{upd}_{l'}(v, q)$, $\mathscr{T} = \{l' : \mathscr{M} \cdot \mathscr{L}'\}; \mathscr{T}_q$, $b = b_v + b_q$, $m = m_v + m_q$, and $d = d_v + d_q$. We consider two cases:
  - Case $l = l'$. Then we are done.
  - Case $l \neq l'$.The hypothesis $l \in \mathtt{dom}(\mathscr{T})$ implies $l \in \mathtt{dom}(\mathscr{T}_q)$, and thus, $l \in \mathtt{dom}(\{l' : \mathscr{L}'\}; \mathscr{T}_q)$. By the *i.h.* on Π_q, we have $q \equiv \mathtt{upd}_l(w, q')$, for some value w and store q'. Therefore, $s \equiv \mathtt{upd}_{l'}(v, \mathtt{upd}_l(w, q')) \equiv \mathtt{upd}_l(w, \mathtt{upd}_{l'}(v, q'))$, and we thus conclude.

The correctness of the original statement now follows easily from the fact that $s \equiv \mathtt{upd}_l(v, q)$ implies $l \in \mathtt{labels}(s)$, by the definition of the domain of a state in Weak Open CBV with Global State (*c.f.* Subsection 3.2). □

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Lemma 15 (Normal forms have pure types). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, *such that* Γ *is tight. If* $t \in \mathtt{no}_{\mathtt{v}}$, *then* (1) $\mathscr{T} = \mathscr{T}'$ *and,* (2) *for any* $\mathscr{T}_0$, *there exists* $\Phi' \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mathscr{T}_0 \Rightarrow (\tau \times \mathscr{T}_0)$.

Proof. The proof follows by induction on t:

- Case $\mathtt{val}(t)$. Then, Φ can only end with rule ($\uparrow$). Therefore, (1) $\mathscr{T} = \mathscr{T}'$, by definition. And (2) Φ' can be built by taking the premise Φ and simply choosing $\mathscr{T}_0$ when applying rule ($\uparrow$) again.
- Case $\neg\mathtt{val}(t)$. Then, $t \in \mathtt{ne}_{\mathtt{v}}$:
  - Case $t = xu$, such that $u \in \mathtt{no}_{\mathtt{v}}$. Then, Φ can only end with rule ($@$), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
    - Case Φ ends with rule ($@$). Then, Φ has two premises $\Phi_x \triangleright x : [\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)] \Vdash_{\mathtt{v}}^{(0,0,0)} x : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T})$ and $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b',m',d')} u : \mathscr{T} \Rightarrow (\mathscr{M} \times \mathscr{T}')$, where $\Gamma = x : [\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)] + \Gamma_u$. Note that a derivation assigning a monadic type to x, has to necessarily finish with ($\uparrow$). However, $\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)$ is not tight, which gives a contradiction with Γ tight, so that this case cannot happen.
    - Case Φ ends with rule ($@^1_{\mathtt{p}}$). Then, Φ has two premises $\Phi_x \triangleright x : [\overline{\mathtt{a}}] \Vdash_{\mathtt{v}}^{(0,0,0)} x : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T})$ and $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b,m,d_u)} u : \mathscr{T} \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')$, such that $\Gamma = (x : [\overline{\mathtt{a}}]) + \Gamma_u$, $\tau = \mathtt{n}$, and $d = 1 + d_u$. By applying the *i.h.* to Φ_u, we get (1) $\mathscr{T} = \mathscr{T}'$, and (2) for any $\mathscr{T}_0$, we have $\Phi'_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \mathscr{T}_0 \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}_0)$. Furthermore, we can trivially derive $\Phi'_x \triangleright x : [\overline{\mathtt{a}}] \Vdash_{\mathtt{v}}^{(0,0,0)} x : \mathscr{T}_0 \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}_0)$ by picking the same $\mathscr{T}_0$. Therefore, (2) we can build Φ' as follows:
      $$\frac{\Phi'_x \qquad \Phi'_u}{(x : [\overline{\mathtt{a}}]) + \Gamma_u \Vdash^{(b,m,1+d_u)} xu : \mathscr{T}_0 \Rightarrow (\mathtt{n} \times \mathscr{T}_0)}(@^1_{\mathtt{p}})$$
      Thus, we can conclude.
    - Case Φ ends with rule ($@^2_{\mathtt{p}}$). This case is very similar to the previous case.
  - Case $t = up$, such that $u \in \mathtt{ne}_{\mathtt{v}}$ and $p \in \mathtt{no}_{\mathtt{v}}$. Then, Φ must end with rule ($@$), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
    - Case Φ ends with rule ($@$). Then, Φ has two premises $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,s_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'')$ and $\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}')$, where $\Gamma = \Gamma_u + \Gamma_p$ is tight. By applying Lemma 41 to Φ_u we get $\mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'') = \mathtt{n}$, which is a contradiction. Therefore, this case cannot hold.
    - Case Φ ends with rule ($@^1_{\mathtt{p}}$) or ($@^2_{\mathtt{p}}$). These cases follow easily from the *i.h.*
  - Case $t = up$, such that $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}}$. Then, Φ must end with rule ($@$), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
    - Case Φ ends with rule ($@$). Then, Φ has two premises $\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,s_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'')$ and $\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}')$, where $\Gamma = \Gamma_u + \Gamma_p$ is tight. By applying Lemma 41 to Φ_p, we get $\mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}') = \mathtt{n}$, which is a contradiction. Therefore, this case cannot hold.
    - Case Φ ends with rule ($@^1_{\mathtt{p}}$) or ($@^2_{\mathtt{p}}$). These cases follow easily from the *i.h.*

□

Proposition 6 (Typed configurations are unblocked). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t, s) : \kappa$. *Then* $\mathtt{tight}(\Gamma)$ *implies* (t, s) *is unblocked.*

Proof. The proof follows by contraposition. We assume that (t, s) is blocked and show that Γ is not tight. We proceed by induction on (t, s) blocked:

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case $t = \mathtt{get}_l(\lambda x.u)$, such that $l \notin \mathtt{labels}(s)$. Then, Φ must be of the following form:

$$\dfrac{\dfrac{\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}{\Gamma_u \setminus\!\setminus x \Vdash^{(b_u,1+m_u,d_u)} \mathtt{get}_l(\lambda x.u) : \{l : \Gamma_u(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}\,(\mathtt{get}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \{l : \Gamma_u(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}}{(\Gamma_u \setminus\!\setminus x) + \Delta_s \Vdash^{(b_u+b_s,1+m_u+m_s,d_u+d_s)} (\mathtt{get}_l(\lambda x.u), s) : \kappa}\,(\mathtt{conf})$$

  such that $\Gamma = (\Gamma_u \setminus\!\setminus x) + \Delta_s$, $b = b_u + b_s$, $m = 1 + m_u + m_s$, and $d = d_u + d_s$. Therefore, by applying Lemma 42 to Π_s, we get $l \in \mathtt{labels}(s)$, which leads to a contradiction. Therefore, this case cannot apply.
- Case $t = up$, such that (u, s) is blocked. We have three cases:
  - Case Φ ends with rule ($\mathtt{@}$). Then, Φ must be of the following form:

$$\dfrac{\dfrac{\begin{array}{c}\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}') \\ \Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}' \Rightarrow (\mathscr{M} \times \mathscr{T}'')\end{array}}{\Gamma_u + \Gamma_p \Vdash^{(1+b_u+b_p,m_u+m_p,d_u+d_p)} up : \mathscr{T} \Rightarrow \kappa}\,(\mathtt{@}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{(\Gamma_u + \Gamma_p) + \Delta_s \Vdash^{(1+b_u+b_p+b_s,m_u+m_p+m_s,d_u+d_p+d_s)} (up, s) : \kappa}\,(\mathtt{conf})$$

    such that $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$, $b = 1 + b_u + b_p + b_s$, $m = m_u + m_p + m_s$, and $d_u + d_p + d_s$. Therefore, we can build the following derivation for (u, s):

$$\dfrac{\Phi_u \qquad \Pi_s}{\Gamma_u + \Delta_s \Vdash^{(b_u+b_s,m_u+m_s,d_u+d_s)} (u, s) : (\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa) \times \mathscr{T}')}\,(\mathtt{conf})$$

    such that $\Gamma_u + \Delta_s$ is tight. However, by applying the *i.h.* to (u, s), we get that $\Gamma_u + \Delta_s$ is not tight. Thus, Γ cannot be tight, since $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$.
  - Case Φ ends with rule ($\mathtt{@}^1_{\mathtt{p}}$). Then, Φ must be of the following form:

$$\dfrac{\dfrac{\begin{array}{c}\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}'') \\ \Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}'' \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')\end{array}}{\Gamma_u + \Gamma_p \Vdash^{(b_u+b_p,m_u+m_p,1+d_u+d_p)} up : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}\,(\mathtt{@}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{(\Gamma_u + \Gamma_p) + \Delta_s \Vdash^{(b_u+b_p+b_s,m_u+m_p+m_s,1+d_u+d_p+d_s)} (up, s) : \kappa}\,(\mathtt{conf})$$

    such that $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$, $b = b_u + b_p + b_s$, $m = m_u + m_p + m_s$, and $1 + d_u + d_p + d_s$. Therefore, we can build the following derivation for (u, s):

$$\dfrac{\Phi_u \qquad \Pi_s}{\Gamma_u + \Delta_s \Vdash^{(b_u+b_s,m_u+m_s,d_u+d_s)} (u, s) : (\overline{\mathtt{a}} \times \mathscr{T}')}\,(\mathtt{conf})$$

    such that $\Gamma_u + \Delta_s$ is tight. However, by applying the *i.h.* to (u, s), we get that $\Gamma_u + \Delta_s$ is not tight. Thus, Γ cannot be tight, since $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$.
  - Case Φ ends with rule ($\mathtt{@}^2_{\mathtt{p}}$). This case is very similar to the previous case.
- Case $t = up$, such that $(u, s) \not\rightarrow_{\mathtt{v}}$, (u, s) is unblocked, and (p, s) is blocked. Moreover, Φ must end in rule ($\mathtt{@}$), ($\mathtt{@}^1_{\mathtt{p}}$), or ($\mathtt{@}^2_{\mathtt{p}}$):

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

– Case Φ ends with rule ($@$). Then, Φ must be of the following form:

$$\dfrac{\dfrac{\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}') \quad \Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}' \Rightarrow (\mathscr{M} \times \mathscr{T}'')}{\Gamma_u + \Gamma_p \Vdash^{(1+b_u+b_p,m_u+m_p,d_u+d_p)} up : \mathscr{T} \Rightarrow \kappa}\,(@) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{(\Gamma_u + \Gamma_p) + \Delta_s \Vdash^{(1+b_u+b_p+b_s,m_u+m_p+m_s,d_u+d_p+d_s)} (up, s) : \kappa}\,(\mathtt{conf})$$

such that $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$, $b = 1 + b_u + b_p + b_s$, $m = m_u + m_p + m_s$, and $d = d_u + d_p + d_s$. Since $(u, s) \not\rightarrow_{\mathtt{v}}$, then Proposition 5 implies that (u, s) is normal, that is $u \in \mathtt{no}_{\mathtt{v}}$ or (u, s) is blocked. But we know that (u, s) is unblocked; therefore, $u \in \mathtt{no}_{\mathtt{v}}$, which implies $\mathscr{T} = \mathscr{T}'$, according to Lemma 15.1. Therefore, we can build the following derivation for (p, s):

$$\dfrac{\Phi_p \qquad \Pi_s}{\Gamma_p + \Delta_s \Vdash^{(b_p+b_s,m_p+m_s,d_p+d_s)} (p, s) : \mathscr{M} \times \mathscr{T}''}(\mathtt{conf})$$

such that $\Gamma_p + \Delta_s$ is tight. However, by applying the *i.h.* to (p, s), we get $\Gamma_p + \Delta_s$ is not tight. Thus, Γ cannot be tight, since $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$.

– Case Φ ends with rule ($@^1_{\mathtt{p}}$). Then, Φ must be of the following form:

$$\dfrac{\dfrac{\Phi_u \triangleright \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}'') \quad \Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}'' \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')}{\Gamma_u + \Gamma_p \Vdash^{(b_u+b_p,m_u+m_p,1+d_u+d_p)} up : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')}\,(@) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{(\Gamma_u + \Gamma_p) + \Delta_s \Vdash^{(b_u+b_p+b_s,m_u+m_p+m_s,1+d_u+d_p+d_s)} (up, s) : \kappa}\,(\mathtt{conf})$$

such that $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$, $b = b_u + b_p + b_s$, $m = m_u + m_p + m_s$, and $d = 1 + d_u + d_p + d_s$. Since $(u, s) \not\rightarrow_{\mathtt{v}}$, then Proposition 5 implies that (u, s) is normal, that is $u \in \mathtt{no}_{\mathtt{v}}$ or (u, s) is blocked. But we know that (u, s) is unblocked; therefore, $u \in \mathtt{no}_{\mathtt{v}}$, which implies $\mathscr{T} = \mathscr{T}''$ according to Lemma 15.1. Therefore, we can build the following derivation for (p, s):

$$\dfrac{\Phi_p \qquad \Pi_s}{\Gamma_p + \Delta_s \Vdash^{(b_p+b_s,m_p+m_s,d_p+d_s)} (p, s) : \mathtt{tt}_{\mathtt{v}} \times \mathscr{T}'}(\mathtt{conf})$$

such that $\Gamma_p + \Delta_s$ is tight. However, by applying the *i.h.* to (p, s), we get that $\Gamma_p + \Delta_s$ is not tight. Thus, we can conclude that Γ cannot be tight, since $\Gamma = (\Gamma_u + \Gamma_p) + \Delta_s$.

– Case Φ ends with rule ($@^2_{\mathtt{p}}$). This case is very similar to the previous case. □

Lemma 43. *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, *such that* Γ *is tight. If* $\tau \in \{\mathtt{v}, \mathtt{n}\} = \overline{\mathtt{a}}$, *then* $\neg\mathtt{abs}(t)$.

Proof. By induction over Φ:

- Case Φ ends with rule ($\mathtt{ax}$), (λ), ($\lambda_{\mathtt{p}}$), or ($\mathtt{m}$). Then, Φ does not conclude with a monadic type. Therefore, these cases hold vacuously.
- Case Φ ends with rule ($@$), ($\mathtt{get}$), ($\mathtt{set}$), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$). Then, $\neg\mathtt{abs}(t)$ by definition. Thus, we can conclude.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case Φ ends with rule ($\uparrow$). Let us assume $t = \lambda x.u$. Then, $\tau = \mathscr{M} \to \mu$, $\tau = \mathscr{M}$, or $\tau = \mathtt{a}$, depending on whether the premise(s) of Φ end with rule(s) (λ), ($\mathtt{m}$), or ($\lambda_{\mathtt{p}}$), respectively. Thus, $\tau \notin \overline{\mathtt{a}}$, so we are done. Now, let us assume $t = x$. Then $\neg\mathtt{abs}(t)$ holds. □

Lemma 44 (Terms in normal form and tight states have correct size).

(1) *Let* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(0,0,d)} t : \mu$ *be a tight derivation. Then,* (*i*) $t \in \mathtt{no}_{\mathtt{v}}$ *and* (*ii*) $d = |t|_{\mathtt{v}}$.

(2) *Let* $\Pi \rhd \Delta \Vdash_{\mathtt{v}}^{(b,m,d)} s : \mathscr{T}$ *be a tight derivation. Then* $b = m = d = 0$.

Proof.

(1) We split statement (1.i) into two parts, but keep the same statement for (1.ii):

(a) If $\mu = \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, then $t \in \mathtt{ne}_{\mathtt{v}}$.
(b) If $\mu = \mathscr{T} \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')$, then $t \in \mathtt{no}_{\mathtt{v}}$.
(c) $d = |t|_{\mathtt{v}}$.

We reason by induction on Φ:

- Case Φ ends with ($\mathtt{ax}$), (λ), ($\mathtt{m}$), or ($\lambda_{\mathtt{p}}$). Then the type of the conclusion is not a monadic type. Therefore, this case holds vacuously.
- Case Φ ends with ($@$). Then the first counter must be greater than zero. Therefore, this case holds vacuously.
- Case Φ ends with ($\uparrow$). Then, $\mathtt{val}(t)$, and thus, $t \subseteq \mathtt{no}_{\mathtt{v}}$ (thus satisfying 1b), $\mu = \mathscr{T} \Rightarrow (\pi \times \mathscr{T}')$, where $\mathscr{T} = \mathscr{T}'$, and Φ has one premise $\Phi' \rhd \Gamma \Vdash_{\mathtt{v}}^{(0,0,d)} t : \pi$, where $\pi = \sigma$ or $\pi = \mathscr{M}$. But, $\mathscr{M}$ is not a tight type, so we just need to check if 1c is satisfied when Φ concludes with type $\pi \in \{\mathtt{v}, \mathtt{a}\}$:
  - Case $\pi = \mathtt{v}$. Then Φ has one premise that must end with rule ($\mathtt{ax}$) and $t = x$. Moreover, $d = 0 = |x|_{\mathtt{v}}$, and thus, 1c is satisfied.
  - Case $\pi = \mathtt{a}$. Then Φ has one premise that must end either with ($\mathtt{ax}$) or ($\lambda_{\mathtt{p}}$), and thus, $t = x$ or $t = \lambda x.u$, respectively. Moreover, $d = 0 = |x|_{\mathtt{v}} = |\lambda x.u|_{\mathtt{v}}$, and thus, 1c is satisfied.
- Case Φ ends with ($\mathtt{get}$) or ($\mathtt{set}$). Then, the second counter must be greater than zero. Therefore, this case holds vacuously.
- Case Φ ends with ($@^1_{\mathtt{p}}$). Then, $t = up$, and Φ has two premises $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{v}}^{(0,0,d_u)} u : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}'')$ and $\Phi_p \rhd \Gamma_p \Vdash_{\mathtt{v}}^{(0,0,d_p)} p : \mathscr{T}'' \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')$, where $\Gamma = \Gamma_u + \Gamma_p$, $\mu = \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, and $d = 1 + d_u + d_p$. By applying the *i.h.* to Φ_u and Φ_p, we get $u, p \in \mathtt{no}_{\mathtt{v}}$, and $d_u = |u|_{\mathtt{v}}$ and $d_p = |p|_{\mathtt{v}}$, respectively. Moreover, $\neg\mathtt{abs}(u)$, according to Lemma 43. Therefore, $u = x$ or $u \in \mathtt{ne}_{\mathtt{v}}$. Therefore, $up \in \mathtt{ne}_{\mathtt{v}}$ (satisfying 1a) Thus, we can conclude with $d = 1 + d_u + d_p = |up|_{\mathtt{v}}$ (satisfying 1c).
- Case Φ ends with ($@^2_{\mathtt{p}}$). This case is very similar to the previous case.

(2) We reason by induction over Π:

- Case Π ends with ($\mathtt{emp}$). Then $\Pi \rhd \emptyset \Vdash_{\mathtt{v}}^{(0,0,0)} \varepsilon : \emptyset$, where $d = 0$.
- Case Π ends with ($\mathtt{upd}$). Then $s = \mathtt{upd}_l(v, q)$ for some v and some q, and Π has two premises $\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$ and $\Pi_q \rhd \Delta_q \Vdash_{\mathtt{v}}^{(b_q,m_q,d_q)} q : \{l : \mathscr{L}\}; \mathscr{T}_q$, where $\Delta = \Gamma_v + \Delta_q$, $\mathscr{T} = \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_q$, $b = b_v + b_q$, $m = m_v + m_q$, and $d = d_v + d_q$. However, since we are assuming that Π is tight, then $\mathscr{T}$ is tight, which means in particular $\mathscr{M} = [\,]$, $\mathscr{L}$ tight and $\mathscr{T}_q$ tight. Therefore, $b_v = m_v = d_v = 0$. Moreover, by applying the *i.h.* to Π_q,

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

we get $b_q = m_q = d_q = 0$. Thus, we can conclude with $b = b_v + b_q = m = m_v + m_q = d = d_v + d_q = 0$. □

Lemma 45 (Terms in normal form take zero steps). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mu$ *be a tight derivation. Then,* $b = m = 0$ *iff* $t \in \mathtt{no}_{\mathtt{v}}$.

Proof.

⇒) By Lemma 44.1.i.

⇐) By induction on $t \in \mathtt{no}_{\mathtt{v}}$:

- Case $t = v$. Then, Φ must end with (ax), (λ), (m), ($\uparrow$), or ($\lambda_{\mathtt{p}}$):
  - Case Φ ends with (ax), (λ), (m), or ($\lambda_{\mathtt{p}}$). Then, Φ does not conclude with a monadic type. Therefore, this case holds vacuously.
  - Φ ends with ($\uparrow$). Then, $\mu = \mathscr{T} \Rightarrow (\pi \times \mathscr{T}')$ and $\pi = \mathtt{v}$ or $\pi = \mathtt{a}$, depending on whether the premise(s) of Φ end with rule(s) (ax) or ($\lambda_{\mathtt{p}}$). In both cases $b = m = 0$, by definition of rules (ax) and ($\lambda_{\mathtt{p}}$).
- Case $t \neq v$. Then, $t \in \mathtt{ne}_{\mathtt{v}}$ by definition. Therefore, (1) $t = xu$, where $u \in \mathtt{no}_{\mathtt{v}}$, (2) $t = up$, where $u \in \mathtt{ne}_{\mathtt{v}}$ and $p \in \mathtt{no}_{\mathtt{v}}$, (3) or $t = up$, where $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}}$. In all cases, Φ must end with rule (@), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
  - Case Φ ends with rule (@). Then, Φ has two premises $\Phi_l \triangleright \Gamma_l \Vdash_{\mathtt{v}}^{(b_l,m_l,d_l)} t_l : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'')$ and $\Phi_r \triangleright \Gamma_r \Vdash_{\mathtt{v}}^{(b_r,m_r,d_r)} t_r : \mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}')$, such that $\Gamma = \Gamma_l + \Gamma_r$ tight, $\mu = \mathscr{T} \Rightarrow \kappa$, $b = 1 + b_l + b_r$, $m = m_l + m_r$, and $d = d_l + d_r$:
    (1) $t_l = x$, $t_r = u$, and $\Gamma_l = (x : [\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)])$. However, $\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)$ is not tight, which leads to a contradiction with Γ tight. Therefore, this case does not apply.
    (2) $t_l = u \in \mathtt{ne}_{\mathtt{v}}$ and $t_r = p$. By Lemma 41, $\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)$ should be equal to n, which leads to a contradiction. Therefore, this case does not apply.
    (3) $t_l = u$ and $t_r = p \in \mathtt{ne}_{\mathtt{v}}$. By Lemma 41, $\mathscr{M}$ should be equal to n, which leads to a contradiction. Therefore, this case does not apply.
  - Case Φ ends with rule ($@^1_{\mathtt{p}}$) or ($@^2_{\mathtt{p}}$). These cases follow easily from the *i.h.* □

Lemma 16 (Tight configurations have correct size). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t, s) : \kappa$ *be a tight derivation of the following form:*

$$\frac{\Phi_t \triangleright \Gamma_t \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}') \quad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{\Gamma \Vdash^{(b,m,d)} (t, s) : \kappa}(\mathtt{conf})$$

where $\Gamma = \Gamma_t + \Delta_s$, $\kappa = (\tau \times \mathscr{T}')$, $b = b_t + b_s$, $m = m_t + m_s$, *and* $d = d_t + d_s$. *If* $b_t = m_t = 0$, *then* (*i*) $b = m = 0$ *and* (*ii*) $d = |(t, s)|_{\mathtt{v}}$.

Proof. Since Γ is tight, then so are Γ_t and Δ_s. Since $b_t = m_t = 0$, then $t \in \mathtt{no}_{\mathtt{v}}$ by Lemma 44.1.i and $d_t = |t|_{\mathtt{v}}$ by Lemma 64.1.ii. Therefore, $\mathscr{T} = \mathscr{T}'$ by Lemma 15.1. Therefore, $\mathscr{T}$ is tight and $b_s = m_s = d_s = 0$ by Lemma 44.2. Thus, we can conclude with $d = d_t + d_s = d_t = |t|_{\mathtt{v}} = |(t, s)|_{\mathtt{v}}$ and $b = b_t + b_s = m = m_t + m_s = 0$. □

Lemma 17 (Unblocked normal configurations take zero steps). *Let* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t, s) : \kappa$ *be a tight derivation. Then,* $b = m = 0$ *iff* (t, s) *is an unblocked normal configuration.*

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Proof. Derivation Φ must be of the following form:

$$\frac{\Phi_t \rhd \Gamma_t \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}') \qquad \Pi_s \rhd \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{\Gamma_t + \Delta_s \Vdash^{(b_t+b_s,m_t+m_s,d_t+d_s)} (t,s) : \tau \times \mathscr{T}'}(\mathtt{conf})$$

where $\Gamma = \Gamma_t + \Delta_s$, $\kappa = (\tau \times \mathscr{T}')$, $b = b_t + b_s$, $m = m_t + m_s$, and $d = d_t + d_s$.

- $\Rightarrow$) Let $b = m = 0$. Then, $b_t = m_t = 0$, and thus, $t \in \mathtt{no}_{\mathtt{n}}$, by Lemma 45. And we can conclude that (t, s) is in normal form.
- $\Leftarrow$) Let (t, s) be in normal form. Then, $t \in \mathtt{no}_{\mathtt{v}}$, by definition. Thus, $b_t = m_t = 0$, by Lemma 16. And we can conclude that $b = m = 0$, by Lemma 45. □

Lemma 46 (Tight typability of states and normal forms).

(1) *Let s be a state. Then, there exists a tight derivation* $\Phi \rhd \emptyset \Vdash_{\mathtt{v}}^{(0,0,0)} s : \mathscr{T}$.

(2) *Let* $t \in \mathtt{no}_{\mathtt{v}}$. *Then, for any* $\mathscr{T}$, *there exists a tight derivation* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(0,0,d)} t : \mathscr{T} \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T})$ *s.t.* $d = |t|_{\mathtt{v}}$.

Proof.

(1) By induction over s:

- Case $s = \varepsilon$. Then, we can build $\Pi_s \rhd \emptyset \Vdash_{\mathtt{v}}^{(0,0,0)} \varepsilon : \emptyset$, by rule (emp), which is trivially tight.
- Case $s = \mathtt{upd}_l(v, q)$. By applying the *i.h.* to q, there exists a tight derivation $\Pi_q \rhd \emptyset \Vdash_{\mathtt{v}}^{(0,0,0)} q : \mathscr{T}_q$. Moreover, $\mathscr{T}_q$ can be written as $\{l : \mathscr{L}\}; \mathscr{T}'_q$, where $\mathscr{L}$ is tight and $\mathscr{T}'_q$ is tight. Therefore, we can build Π_s as follows:

$$\frac{\dfrac{}{\emptyset \Vdash^{(0,0,0)} v : [\,]}(\mathtt{m}) \qquad \emptyset \Vdash_{\mathtt{v}}^{(0,0,0)} q : \{l : \mathscr{L}\}; \mathscr{T}'_q}{\emptyset \Vdash^{(0,0,0)} \mathtt{upd}_l(v, q) : \{l : [\,] \cdot \mathscr{L}\}; \mathscr{T}'_q}(\mathtt{upd})$$

  We conclude since $\{l : [\,] \cdot \mathscr{L}\}; \mathscr{T}'_q$ is tight.

(2) We reason by induction over $t \in \mathtt{no}_{\mathtt{v}}$:

- Case $\mathtt{val}(t)$. Then, (a) $t = x$ or (b) $t = \lambda x.u$. Therefore, we can build Φ by (a) rule (ax), or rule $(\lambda_{\mathtt{p}})$, followed by rule $(\uparrow)$. Thus, we can conclude.
- Case $\neg\mathtt{val}(t)$. Then, $t \in \mathtt{ne}_{\mathtt{v}}$:
  - Case $t = xu$, such that $u \in \mathtt{no}_{\mathtt{v}}$. By applying the *i.h.* to u, for any $\mathscr{T}'$, there exists a tight derivation $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{v}}^{(0,0,|u|_{\mathtt{v}})} u : \mathscr{T}' \Rightarrow (\tau' \times \mathscr{T}')$, such that $\tau' \in \mathtt{tt}_{\mathtt{v}}$. Moreover, we can build $\Phi_x \rhd x : [\overline{\mathtt{a}}] \Vdash_{\mathtt{v}}^{(0,0,0)} x : \mathscr{T}' \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}')$. Therefore, we can build Φ as follows:
$$\frac{\Phi_x \qquad \Phi_u}{(x : [\overline{\mathtt{a}}]) + \Gamma_u \Vdash^{(0,0,1+|u|_{\mathtt{v}})} xu : \mathscr{T}' \Rightarrow (\mathtt{n} \times \mathscr{T}')}(@^1_{\mathtt{p}})$$
    such that $\Gamma = (x : [\overline{\mathtt{a}}]) + \Gamma_u$ tight, and $1 + |u|_{\mathtt{v}} = |xu|_{\mathtt{v}}$.
  - Case $t = up$, such that $u \in \mathtt{no}_{\mathtt{v}}$ and $p \in \mathtt{ne}_{\mathtt{v}}$. By the *i.h.*, for any $\mathscr{T}'$, there exist $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{v}}^{(0,0,|u|_{\mathtt{v}})} u : = \mathscr{T}' \Rightarrow (\tau' \times \mathscr{T}')$ tight, such that $\tau' \in \mathtt{tt}_{\mathtt{v}}$, and $\Phi_p \rhd \Gamma_p \Vdash_{\mathtt{v}}^{(0,0,|p|_{\mathtt{v}})} u : \mathscr{T}' \Rightarrow (\tau'' \times \mathscr{T}')$, such that $\tau'' \in \mathtt{tt}_{\mathtt{v}}$. Moreover, since Γ_p is tight, and $p \in \mathtt{ne}_{\mathtt{v}}$, we know

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$\tau'' = \mathtt{n}$, according to Lemma 41. Therefore, we can build Φ as follows:

$$\frac{\Phi_u \qquad \Phi_p}{\Gamma_u + \Gamma_p \Vdash^{(0,0,1+|u|_\mathtt{v}+|p|_\mathtt{v})} up : \mathscr{T}' \Rightarrow (\mathtt{n} \times \mathscr{T}')}(@^1_\mathtt{p})$$

such that $\Gamma = \Gamma_u + \Gamma_p$ tight, and $1 + |u|_\mathtt{v} + |p|_\mathtt{v} = |up|_\mathtt{v}$.

– Case $t = up$, such that $u \in \mathtt{ne}_\mathtt{v}$ and $p \in \mathtt{no}_\mathtt{v}$. This case is very similar to the previous one.

□

Lemma 18 (All configurations in normal form are tightly typable). *If (t, s) is an unblocked normal configuration, then there exists a tight typing derivation $\Phi \triangleright \Gamma \Vdash^{(b,m,d)}_\mathtt{v} (t, s) : \kappa$, such that $d = |(t, s)|_\mathtt{v}$.*

Proof. By Lemma 46. □

Lemma 47 (Multi-types can split and merge). *Let $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b = +_{i\in I} b_i$, $m = +_{i\in I} m_i$, and $d = +_{i\in I}$. Then, $\Phi_t \triangleright \Gamma \Vdash^{(b,m,d)}_\mathtt{v} v : \mathscr{M}$ iff $(\Phi^i_t \triangleright \Gamma_i \Vdash^{(b_i,m_i,d_i)}_\mathtt{v} v :: \mathscr{M}_i)_{i\in I}$.*

Proof. We omit this proof given its similarity with the proof for Weak Open CBV. □

Lemma 48 (States can split). *Let $\Pi_s \triangleright \Delta \Vdash^{(b,m,d)}_\mathtt{v} s : \mathscr{T}$, such that $l \in \mathtt{dom}(\mathscr{T})$. Then, $s \equiv \mathtt{upd}_l(v, q)$, $\mathscr{T}(l) = \mathscr{M} \cdot \mathscr{L}$, $\Phi_v \triangleright \Gamma_v \Vdash^{(b_v,m_v,d_v)}_\mathtt{v} v : \mathscr{M}$ and $\Pi_q \triangleright \Delta_q \Vdash^{(b_q,m_q,d_q)}_\mathtt{v} \exists q : \{l : \mathscr{L}\}; \mathscr{T}'$, such that $\Delta = \Gamma_v + \Delta_q$, $\mathscr{T} = \{l : \mathscr{T}(l)\}; \mathscr{T}'$, $b = b_v + b_q$, $m = m_v + m_q$, and $d = d_v + d_q$.*

Proof. The proof follows by induction on the structure of s:

- Case $s = \varepsilon$. Then the statement is vacuously true.
- Case $s = \mathtt{upd}_{l'}(w, q')$. Then Φ_s is of the form:

$$\frac{\Phi_w \triangleright \Gamma_w \Vdash^{(b_w,m_w,d_w)}_\mathtt{v} w : \mathscr{M}' \qquad \Pi_{q'} \triangleright \Delta_{q'} \Vdash^{(b_{q'},m_{q'},d_{q'})}_\mathtt{v} q' : \{l' : \mathscr{L}'\}; \mathscr{T}_{q'}}{\Gamma_w + \Delta_{q'} \Vdash^{(b_w+b_{q'},m_w+m_{q'},d_w+d_{q'})} \mathtt{upd}_{l'}(w, q') : \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \mathscr{T}_{q'}}(\mathtt{upd})$$

  where $\Delta = \Gamma_w + \Delta_{q'}$, $\mathscr{T} = \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \mathscr{T}_{q'}$, $b = b_w + b_{q'}$, $m = m_w + m_{q'}$, and $d = d_w + d_{q'}$. We consider two cases:

  – Case $l' = l$. Then we simply take $v = w$ and $q = q'$, and we are done.
  – Case $l' \neq l$. Since $l \in \mathtt{dom}(\{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \mathscr{T}_{q'})$ and $l' \neq l$, then $l \in \mathtt{dom}(\mathscr{T}_{q'})$. By applying the *i.h.* to $\Pi_{q'}$, we have that $q' \equiv \mathtt{upd}_l(w', q'')$, $(\{l' : \mathscr{L}'\}; \mathscr{T}_{q'})(l) = \mathscr{T}_{q'}(l) = \mathscr{M}'' \cdot \mathscr{L}''$, $\Phi_{w'} \triangleright \Gamma_{w'} \Vdash^{(b_{w'},m_{w'},d_{w'})}_\mathtt{v} w' : \mathscr{M}''$ and $\Pi_{q''} \triangleright \Delta_{q''} \Vdash^{(b_{q''},m_{q''},d_{q''})}_\mathtt{v} q'' : \{l : \mathscr{L}''\}; \{l' : \mathscr{L}'\}; \mathscr{T}_{q''}$, such that $\Delta_{q'} = \Gamma_{w'} + \Delta_{q''}$, $\{l' : \mathscr{L}'\}; \mathscr{T}_{q'} = \{l : \mathscr{M}'' \cdot \mathscr{L}''\}; \{l' : \mathscr{L}'\}; \mathscr{T}_{q''}$, $b_{q'} = b_{w'} + b_{q''}$, $m_{q'} = m_{w'} + m_{q''}$, and $d_{q'} = d_{w'} + d_{q''}$. But $s = \mathtt{upd}_{l'}(w, \mathtt{upd}_l(w', q'')) \equiv \mathtt{upd}_l(w', \mathtt{upd}_{l'}(w, q''))$, so we can take $v = w'$, $q = \mathtt{upd}_{l'}(w, q'')$, $\mathscr{M} = \mathscr{M}''$, $\mathscr{L} = \mathscr{L}''$, and $\mathscr{T}' = \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \mathscr{T}_{q''}$, and consider Φ_q to be the following derivation:

$$\frac{\Phi_w \triangleright \Gamma_w \Vdash^{(b_w,m_w,d_w)}_\mathtt{v} w : \mathscr{M} \qquad \Pi_{q''} \triangleright \Delta_{q''} \Vdash^{(b_{q''},m_{q''},d_{q''})}_\mathtt{v} q'' : \{l' : \mathscr{L}'\}; \{l : \mathscr{L}''\}; \mathscr{T}_{q''}}{\Gamma_w + \Delta_{q''} \Vdash^{(b_w+b_{q''},m_w+m_{q''},d_w+d_{q''})} \mathtt{upd}_{l'}(w, q'') : \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \{l : \mathscr{L}''\}; \mathscr{T}_{q''}}(\mathtt{upd})$$

  where $\Delta_q = \Gamma_w + \Delta_{q''}$ and $\mathscr{T}_q = \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \{l : \mathscr{L}''\}; \mathscr{T}_{q''}$. We can then conclude with the following observations:

  $* \ \Gamma_v + \Delta_q = \Gamma_{w'} + \Gamma_w + \Delta_{q''} = \Gamma_w + \Delta_{q'} = \Delta$,

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

∗ Since $\mathscr{T} = \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \mathscr{T}_{q'}$ and $l' \neq l$, then $\mathscr{T}(l) = \mathscr{T}_{q'}(l)$ and

$$\begin{aligned}\mathscr{T} = \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \mathscr{T}_{q'} &= \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \{l : \mathscr{M}'' \cdot \mathscr{L}''\}; \mathscr{T}_{q''} \\ &= \{l : \mathscr{T}_{q'}(l)\}; \{l' : \mathscr{M}' \cdot \mathscr{L}'\}; \mathscr{T}_{q''} \\ &= \{l : \mathscr{T}(l)\}; \mathscr{T}'\end{aligned}$$

∗ $b_v + b_q = b_{w'} + b_w + b_{q''} = b_w + b_{q'} = b, m_v + m_q = m_{w'} + m_w + m_{q''} = m_w + m_{q'} = b$ and $d_v + d_q = d_{w'} + d_w + d_{q''} = d_w + d_{q'} = d$. □

Lemma 49. *Let* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$. *If* t *is a value, then* $\tau \neq \mathtt{n}$.

Proof. By a simple inspection of the rules. □

Lemma 50 (Substitution). *If* $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mu$ *and* $\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, *there exists* $\Phi_{t\{x\setminus v\}} \rhd \Gamma_t + \Gamma_v \Vdash_{\mathtt{v}}^{(b_t+b_v,m_t+m_v,d_t+d_v)} t\{x \setminus v\} : \mu$.

Proof. We generalize the original statement by allowing Φ_t to conclude with either a monadic type μ or a type τ. Let $\mathscr{G} \in \{\mu, \tau\}$. Then the statement is as follows:

If $\Phi_t \rhd \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mathscr{G}$ and $\Phi_u \rhd \Gamma_u \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \mathscr{M}$, then $\Phi_{t\{x\setminus u\}} \rhd \Gamma_t + \Gamma_u \Vdash_{\mathtt{v}}^{(b_t+b_u,m_t+m_u,d_t+d_u)} t\{x \setminus u\} : \mathscr{G}$.

The proof is by induction over Φ_t, according to the last rule used of Φ_t:

- Case $\mathscr{G} = \mu$. Then, Φ_t must end with rule (`@`), (↑), (`get`), (`set`), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
  - Rule (`@`). Then $t = up$, $\mu = \mathscr{T} \Rightarrow \kappa$, and Φ_t has two premises $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_1 \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'')$ and $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}')$, where $\Gamma = \Gamma_u + \Gamma_p$, $\mathscr{M} = \mathscr{M}_1 \sqcup \mathscr{M}_2$, $b_t = 1 + b_u + b_p$, $m_t = m_u + m_p$, and $d_t = d_u + d_p$. By Lemma 47, there exist $\Phi^1_v \rhd \Gamma^1_v \Vdash_{\mathtt{v}}^{(b^1_v,m^1_v,d^1_v)} v\mathscr{M}_1$ and $\Phi^2_v \rhd \Gamma^2_v \Vdash_{\mathtt{v}}^{(b^2_v,m^2_v,d^2_v)} v : \mathscr{M}_2$, where $\Gamma_v = \Gamma^1_v + \Gamma^1_v$, $b_v = b^1_v + b^2_v$, $m_v = m^1_v + m^2_v$, and $d_v = d^1_v + d^2_v$. By applying the *i.h.* to Φ_u and Φ_p, there exists $\Phi_{u\{x\setminus v\}} \rhd \Gamma_u + \Gamma^1_v \Vdash_{\mathtt{v}}^{(b_u+b^1_v,m_u+m^1_v,d_u+d^1_u)} u\{x \setminus v\} : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'')$ and $\Phi_{p\{x\setminus v\}} \rhd \Gamma_p + \Gamma^2_v \Vdash_{\mathtt{v}}^{(b_p+b^2_v,m_p+m^2_v,d_p+d^2_v)} p\{x \setminus v\} : \mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}')$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{\Phi_{u\{x\setminus v\}} \qquad \Phi_{p\{x\setminus v\}}}{(\Gamma_u + \Gamma_p) + (\Gamma^1_v + \Gamma^2_v) \Vdash^{(1+b_u+b_p+b^1_v+b^2_v,m_u+m_p+m^1_v+m^2_v,d_u+d_p+d^1_v+d^2_v)} (up)\{x \setminus v\} : \mathscr{T} \Rightarrow \kappa}(@)$$

And we can conclude with $\Gamma_t + \Gamma_v = (\Gamma_u + \Gamma_p) + (\Gamma^1_v + \Gamma^2_v)$, $b_t + b_v = 1 + b_u + b_p + b^1_v + b^2_v$, $m_t + m_v = m_u + m_p + m^1_v + m^2_v$, and $d_t + d_v = d_u + d_p + d^1_v + d^2_v$.
  - Rule (↑). Then $t = w$, $\mu = \mathscr{T} \Rightarrow (\pi \times \mathscr{T})$, and Φ_t has one premise $\Phi_w \rhd \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} w : \pi$. By applying *i.h.* to Φ_w, there exists $\Phi_{w\{x\setminus v\}} \rhd \Gamma_w + \Gamma_v \Vdash_{\mathtt{v}}^{(b_t+b_v,m_t+m_v,d_t+d_v)} w\{x \setminus v\} : \pi$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{\Phi_{w\{x\setminus v\}} \rhd \Gamma_t + \Gamma_v \Vdash_{\mathtt{v}}^{(b_t+b_v,m_t+m_v,d_t+d_v)} w\{x \setminus v\} : \pi}{\Gamma_t + \Gamma_v \Vdash^{(b_t+b_v,m_t+m_v,d_t+d_v)} w\{x \setminus v\} : \mathscr{T} \Rightarrow (\pi \times \mathscr{T})}(\uparrow)$$

And we can conclude.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Rule (get). Then $t = \mathtt{get}_l(\lambda y.u)$, $\mu = \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and Φ_t has one premise $\Phi_u \rhd (\Gamma_u; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t, m_u, d_t)} u : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, where $\Gamma_t = \Gamma_u$ (since we can assume that $y \neq x$ by α-conversion), and $m_t = 1 + m_u$. By applying the *i.h.* to Φ_u, there exists $\Phi_{u\{x\setminus v\}} \rhd (\Gamma_u; y : \mathscr{M}') + \Gamma_v \Vdash_{\mathtt{v}}^{(b_t+b_v, 1+m_u+m_v, d_t+d_v)} u\{x \setminus v\} : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$. Moreover, by α-conversion, we can assume since $y \notin \mathtt{dom}(\Gamma_v)$, so that $(\Gamma_u; y : \mathscr{M}') + \Gamma_v = \Gamma_u + \Gamma_v; y : \mathscr{M}' = \Gamma_t + \Gamma_v; y : \mathscr{M}'$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:
- Rule (get). Then $t = \mathtt{get}_l(\lambda y.u)$, $\mu = \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and Φ_t has one premise $\Phi_u \rhd (\Gamma_u; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t, m_u, d_t)} u : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, where $\Gamma_t = \Gamma_u$ (since we can assume that $y \neq x$ by α-conversion), and $m_t = 1 + m_u$. By applying the *i.h.* to Φ_u, there exists $\Phi_{u\{x\setminus v\}} \rhd (\Gamma_u; y : \mathscr{M}') + \Gamma_v \Vdash_{\mathtt{v}}^{(b_t+b_v, 1+m_u+m_v, d_t+d_v)} u\{x \setminus v\} : \{l : \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$. Moreover, by α-conversion, we can assume since $y \notin \mathtt{dom}(\Gamma_v)$, so that $(\Gamma_u; y : \mathscr{M}') + \Gamma_v = \Gamma_u + \Gamma_v; y : \mathscr{M}' = \Gamma_t + \Gamma_v; y : \mathscr{M}'$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{\Phi_{u\{x\setminus v\}}}{\Gamma_t + \Gamma_v \Vdash^{(b_t+b_v, 1+m_u+m_v, d_t+d_v)} \mathtt{get}_l(\lambda y.u\{x \setminus v\}) : \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{get})$$

  And we can conclude with $m_t = 1 + m_u$.
- Rule (set). Then $t = \mathtt{set}_l(w, u)$, $\mu = \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and Φ_t has two premises $\Phi_w \rhd \Gamma_w; x : \mathscr{M}_1 \Vdash_{\mathtt{v}}^{(b_w, m_w, d_w)} w : \mathscr{M}'$ and $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_2 \Vdash_{\mathtt{v}}^{(b_u, m_u, d_u)} u : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, where $\Gamma_t = \Gamma_w + \Gamma_u$, $b_t = b_w + b_u$, $m_t = 1 + m_w + m_u$, and $d_t = d_w + d_u$. By Lemma 47, there exist $\Phi_v^1 \rhd \Gamma_v^1 \Vdash_{\mathtt{v}}^{(b_v^1, m_v^1, d_v^1)} v : \mathscr{M}_1$ and $\Phi_v^2 \rhd \Gamma_v^2 \Vdash_{\mathtt{v}}^{(b_v^2, m_v^2, d_v^2)} v : \mathscr{M}_2$, where $\Gamma_v = \Gamma_v^1 + \Gamma_v^2$, $b_v = b_v^1 + b_v^2$, $m_v = m_v^1 + m_v^2$, and $d_v = d_v^1 + d_v^2$. By applying *i.h.* to Φ_w and the *i.h.* to Φ_u, there exist $\Phi_{w\{x\setminus v\}} \rhd \Gamma_w + \Gamma_v^1 \Vdash_{\mathtt{v}}^{(b_w+b_v^1, m_w+m_v^1, d_w+d_v^1)} w\{x \setminus v\} : \mathscr{M}'$ and $\Phi_{u\{x\setminus v\}} \rhd \Gamma_u + \Gamma_v^2 \Vdash_{\mathtt{v}}^{(b_u+b_v^2, m_u+m_u^2, d_u+d_v^2)} u\{x \setminus v\} : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:
- Rule (set). Then $t = \mathtt{set}_l(w, u)$, $\mu = \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and Φ_t has two premises $\Phi_w \rhd \Gamma_w; x : \mathscr{M}_1 \Vdash_{\mathtt{v}}^{(b_w, m_w, d_w)} w : \mathscr{M}'$ and $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_2 \Vdash_{\mathtt{v}}^{(b_u, m_u, d_u)} u : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, where $\Gamma_t = \Gamma_w + \Gamma_u$, $b_t = b_w + b_u$, $m_t = 1 + m_w + m_u$, and $d_t = d_w + d_u$. By Lemma 47, there exist $\Phi_v^1 \rhd \Gamma_v^1 \Vdash_{\mathtt{v}}^{(b_v^1, m_v^1, d_v^1)} v : \mathscr{M}_1$ and $\Phi_v^2 \rhd \Gamma_v^2 \Vdash_{\mathtt{v}}^{(b_v^2, m_v^2, d_v^2)} v : \mathscr{M}_2$, where $\Gamma_v = \Gamma_v^1 + \Gamma_v^2$, $b_v = b_v^1 + b_v^2$, $m_v = m_v^1 + m_v^2$, and $d_v = d_v^1 + d_v^2$. By applying *i.h.* to Φ_w and the *i.h.* to Φ_u, there exist $\Phi_{w\{x\setminus v\}} \rhd \Gamma_w + \Gamma_v^1 \Vdash_{\mathtt{v}}^{(b_w+b_v^1, m_w+m_v^1, d_w+d_v^1)} w\{x \setminus v\} : \mathscr{M}'$ and $\Phi_{u\{x\setminus v\}} \rhd \Gamma_u + \Gamma_v^2 \Vdash_{\mathtt{v}}^{(b_u+b_v^2, m_u+m_u^2, d_u+d_v^2)} u\{x \setminus v\} : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{\Phi_{w\{x\setminus v\}} \qquad \Phi_{u\{x\setminus v\}}}{(\Gamma_w + \Gamma_u) + (\Gamma_v^1 + \Gamma_v^2) \Vdash^{(b_w+b_u+b_v^1+b_v^2, 1+m_w+m_u+m_v^1+m_v^2, d_w+d_u+d_v^1+d_v^2)} \mathtt{set}_l(w, u)\{x \setminus v\} : \{l : \mathscr{L}\} \Rightarrow \kappa}(\mathtt{set})$$

  And we can conclude with $\Gamma_t + \Gamma_v = (\Gamma_w + \Gamma_u) + (\Gamma_v^1 + \Gamma_v^2)$, $b_t + b_v = b_w + b_u + b_v^1 + b_v^2$, $m_t + m_v = 1 + m_w + m_u + m_v^1 + m_v^2$, and $d_t + d_v = d_w + d_u + d_v^1 + d_v^2$.
- Rule $(@_{\mathtt{p}}^1)$. Then $t = up$, $\mu = \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, and Φ_t has two premises $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_1 \Vdash_{\mathtt{v}}^{(b_u, m_u, d_u)} u : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}'')$ and $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \Vdash_{\mathtt{v}}^{(b_p, m_p, d_p)} p : \mathscr{T}'' \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')$, where $\Gamma = \Gamma_u + \Gamma_p$, $\mathscr{M} = \mathscr{M}_1 \sqcup \mathscr{M}_2$, $b_t = b_u + b_p$, $m_t = m_u + m_p$, and $d_t = 1 + d_u + d_p$. By Lemma 47, there exist $\Phi_v^1 \rhd \Gamma_v^1 \Vdash_{\mathtt{v}}^{(b_v^1, m_v^1, d_v^1)} v : \mathscr{M}_1$ and $\Phi_v^2 \rhd \Gamma_v^2 \Vdash_{\mathtt{v}}^{(b_v^2, m_v^2, d_v^2)} v : \mathscr{M}_2$, where $\Gamma_v = \Gamma_v^1 + \Gamma_v^2$, $b_v = b_v^1 + b_v^2$, $m_v = m_v^1 + m_v^2$, and $d_v = d_v^1 + d_v^2$. By applying the *i.h.* to Φ_u and Φ_p, there

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

exist $\Phi_{u\{x\setminus v\}} \rhd \Gamma_u + \Gamma_v^1 \Vdash_{\mathtt{v}}^{(b_u+b_v^1, m_u+m_v^1, d_u+d_u^1)} u\{x \setminus v\} : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}'')$ and $\Phi_{p\{x\setminus v\}} \rhd \Gamma_p + \Gamma_v^2 \Vdash_{\mathtt{v}}^{(b_p+b_v^2, m_p+m_v^2, d_p+d_v^2)} p\{x \setminus v\} : \mathscr{T}'' \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{\Phi_{u\{x\setminus v\}} \qquad \Phi_{p\{x\setminus v\}}}{(\Gamma_u + \Gamma_p) + (\Gamma_v^1 + \Gamma_v^2) \Vdash^{(b_u+b_p+b_v^1+b_v^2, m_u+m_p+m_v^1+m_v^2, 1+d_u+d_p+d_v^1+d_v^2)} (up)\{x \setminus v\} : \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')} (@_{\mathtt{p}}^1)$$

And we can conclude with $\Gamma_t + \Gamma_v = (\Gamma_u + \Gamma_p) + (\Gamma_v^1 + \Gamma_v^2)$, $b_t + b_v = b_u + b_p + b_v^1 + b_v^2$, $m_t + m_v = m_u + m_p + m_v^1 + m_v^2$, and $d_t + d_v = 1 + d_u + d_p + d_v^1 + d_v^2$.

– Rule $(@_{\mathtt{p}}^2)$. This case is very similar to the previous one.

- Case $\mathscr{G} = \tau$. Then, Φ_t must end with rule (ax), (λ), (m), or $(\lambda_{\mathtt{p}})$:
  - Rule (ax). Then, $t = y$. There are two cases:
    - $y = x$. Then $x\{x \setminus v\} = v$, $\tau = \sigma$, and $\Phi_t \rhd \emptyset; (x : [\sigma]) \Vdash_{\mathtt{v}}^{(0,0,0)} x : \sigma$, where $\Gamma_t = \emptyset$, $\mathscr{M} = [\sigma]$, and $b_t = m_t = d_t = 0$. Moreover, Φ_v must end with rule (m) with one premise $\Phi_v' \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v, m_v, d_v)} v : \sigma$. So we can take $\Phi_{t\{x\setminus v\}} = \Phi_v'$, and conclude with $\Gamma_t + \Gamma_v = \Gamma_v$, $b_t + b_v = b_v$, $m_t + m_v = m_v$, and $d_t + d_v = d_v$.
    - $y \neq x$. Then $y\{x \setminus v\} = y$, $\tau = \sigma$, and $\Phi_t \rhd y : [\sigma] \Vdash_{\mathtt{v}}^{(0,0,0)} y : \sigma$, where $\Gamma_t = (y : [\sigma])$, $\mathscr{M} = [\,]$, and $b_t = m_t = d_t = 0$. Moreover, Φ_v ends with rule (m) with no premises. Therefore, $\Gamma_v = \emptyset$, $b_v = m_v = d_v = 0$. So we can pick $\Phi_{t\{x\setminus v\}} = \Phi_t$, and conclude with $\Gamma_t + \Gamma_v = \Gamma_t$, $b_t + b_v = b_t$, $m_t + m_v = m_t$, and $d_t + d_v = d_t$.
  - Rule (λ). Then $t = \lambda y.u$, $\tau = \mathscr{M}' \to \mu$, and Φ_t has one premise $\Phi_u \rhd (\Gamma_u; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t, m_t, d_t)} u : \mu$, where $\Gamma_t = \Gamma_u$ (since we can assume that $y \neq x$ by α-conversion). By applying the *i.h.* to Φ_u, there exists $\Phi_{u\{x\setminus v\}} \rhd (\Gamma_u; y : \mathscr{M}') + \Gamma_v \Vdash_{\mathtt{v}}^{(b_t+b_v, m_t+m_v, d_t+d_v)} u\{x \setminus v\} : \mu$. Moreover, by α-conversion, we can assume since $y \notin \mathtt{dom}(\Gamma_v)$, so that $(\Gamma_u; y : \mathscr{M}') + \Gamma_v = \Gamma_u + \Gamma_v; y : \mathscr{M}'$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{\Phi_{u\{x\setminus v\}}}{\Gamma_u + \Gamma_v \Vdash^{(b_t+b_v, m_t+m_v, d_t+d_v)} \lambda y.(u\{x \setminus v\}) : \mathscr{M}' \to \mu} (\lambda)$$

And we can conclude since $\Gamma_t + \Gamma_v = \Gamma_u + \Gamma_v$ and $t\{x \setminus v\} = \lambda y.(u\{x \setminus v\})$.

  - Rule (m). Then $t = w$, $\tau = [\sigma_i]_{i\in I}$, and Φ_t has premises $(\Phi_w^i \rhd \Gamma_w^i; x : \mathscr{M}_i \Vdash_{\mathtt{v}}^{(b_w^i, m_w^i, d_w^i)} w : \sigma_i)_{i\in I}$, where $\Gamma_t = +_{i\in I}\Gamma_w^i$, $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b_t = +_{i\in I} b_w^i$, $m_t = +_{i\in I} m_w^i$, and $d_t = +_{i\in I} d_w^i$. By Lemma 47, there exist $(\Phi_v^i \rhd \Gamma_v^i \Vdash_{\mathtt{v}}^{(b_v^i, m_v^i, d_v^i)} v : \mathscr{M}_i)_{i\in I}$, where $\Gamma_v = +_{i\in I}\Gamma_v^i$, $b_v = +_{i\in I} b_v^i$, $m_v = +_{i\in I} m_v^i$, and $d_v = +_{i\in I} d_v^i$. By applying the *i.h.* to each Φ_w^i, there exists $(\Phi_{w\{x\setminus v\}}^i \rhd \Gamma_w^i + \Gamma_v^i \Vdash_{\mathtt{v}}^{(b_w^i+b_v^i, m_w^i+m_v^i, d_w^i+d_v^i)} w\{x \setminus v\} : \sigma_i)_{i\in I}$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{(\Phi_{w\{x\setminus v\}}^i)_{i\in I}}{+_{i\in I}(\Gamma_v^i + \Gamma_w^i) \Vdash^{(+_{i\in I}(b_w^i+b_v^i), +_{i\in I}(m_w^i+m_v^i), +_{i\in I}(d_w^i+d_v^i))} w\{x \setminus v\} : [\tau_i]_{i\in I}} (\mathtt{m})(\mathtt{m})$$

And we can conclude with $\Gamma_t + \Gamma_v = +_{i\in I}\Gamma_w^i +_{i\in I} \Gamma_v^i = +_{i\in I}(\Gamma_w^i + \Gamma_v^i)$, $b_t + b_v = +_{i\in I} b_w^i +_{i\in I} b_v^i = +_{i\in I}(b_w^i + b_v^i)$, $m_t + m_v = +_{i\in I} m_w^i +_{i\in I} m_v^i = +_{i\in I}(m_w^i + m_v^i)$, and $d_t + d_v = +_{i\in I} d_w^i +_{i\in I} d_v^i = +_{i\in I}(d_w^i + d_v^i)$.

  - Rule $(\lambda_{\mathtt{p}})$. Then $t = \lambda y.u$, $\tau = \mathtt{a}$, and $\Phi_t \rhd \emptyset; x : [\,] \Vdash_{\mathtt{v}}^{(0,0,0)} \lambda y.u : \mathtt{a}$, where $\Gamma_t = \emptyset$, $\mathscr{M} = [\,]$, and $b_t = m_t = d_t = 0$. Also, notice that $t\{x \setminus v\} = \lambda y.(u\{x \setminus v\})$. Therefore, Φ_v ends with

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

rule (m) with no premises, and $\Gamma_v = \emptyset$. Therefore, we can build $\Phi_{t\{x\setminus v\}}$ as follows:

$$\frac{}{\emptyset \Vdash_{\mathtt{v}}^{(0,0,0)} (\lambda y.u)\{x \setminus v\} : \mathtt{a}}(\lambda_{\mathtt{p}})$$

And we can conclude with $\Gamma_t + \Gamma_v = \emptyset$, $b_t + b_v = m_t + m_v = d_t + d_v = 0$. □

Lemma 19 (Quantitative subject reduction). *Let* $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, *where* $\mathtt{r} \in \{\beta_{\mathtt{v}}, \mathtt{g}, \mathtt{s}\}$. *If* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t,s) : \kappa$ *is a tight derivation, then* $\Phi' \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b',m',d)} (u,q) : \kappa$, *where* $\mathtt{r} = \beta_{\mathtt{v}}$ *implies* $b' = b-1$ *and* $m' = m$, *while* $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ *implies* $b' = b$ *and* $m' = m-1$.

Proof. We show a stronger statement of the form:

Let $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, where $\mathtt{r} \in \{\beta_{\mathtt{v}}, \mathtt{g}, \mathtt{s}\}$. If $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t,s) : \kappa$, Γ is tight, and (κ is tight or $\neg\mathtt{val}(t)$), then $\Phi' \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b',m',d)} (u,q) : \kappa$, where $\mathtt{r} = \beta_{\mathtt{v}}$ implies $b' = b-1$ and $m' = m$, while $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ implies $b' = b$ and $m' = m-1$.

We proceed by induction on $(t,s) \rightsquigarrow_{\mathtt{v}} (u,q)$:

- Case $(t,s) = ((\lambda x.p)v, s) \rightsquigarrow_{\beta_{\mathtt{v}}} (p\{x \setminus v\}, s) = (u,q)$. Then, Φ has two premises $\Phi_{(\lambda x.p)v} \triangleright \Gamma_{(\lambda x.p)v} \Vdash_{\mathtt{v}}^{(b_{(\lambda x.p)v}, m_{(\lambda x.p)v}, d_{(\lambda x.p)v})} (\lambda x.p)v : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$ and $\Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}$, such that $\Gamma = \Gamma_{(\lambda x.p)v} + \Delta_s$, $b = b_{(\lambda x.p)v} + b_s$, $m = m_{(\lambda x.p)v} + m_s$, and $d = d_{(\lambda x.p)v} + d_s$. Then, $\Phi_{(\lambda x.p)v}$ must end with rule (@), ($@_{\mathtt{p}}^1$), or ($@_{\mathtt{p}}^2$). Therefore, $\Phi_{(\lambda x.p)v}$ has two premises $\Phi_{\lambda x.p} \triangleright \Gamma_{\lambda x.p} \Vdash_{\mathtt{v}}^{(b_{\lambda x.p}, m_{\lambda x.p}, d_{\lambda x.p})} \lambda x.p : \mathscr{T} \Rightarrow (\tau_1 \times \mathscr{T}'')$ and $\Phi_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{T}'' \Rightarrow (\tau_2 \times \mathscr{T}')$, such that $\Gamma_{(\lambda x.p)v} = \Gamma_{\lambda x.p} + \Gamma_v$, $b_{(\lambda x.p)v} = b_{\lambda x.p} + b_v$, $m_{(\lambda x.p)v} = m_{\lambda x.p} + m_v$, and $d_{(\lambda x.p)v} = 1 + d_{\lambda x.p} + d_v$. Let us first assume $\Phi_{(\lambda x.p)v}$ ends with ($@_{\mathtt{p}}^1$). Then, $\tau_1 = \overline{\mathtt{a}}$. However, according to Lemma 43 $\neg\mathtt{abs}(\lambda x.p)$ should hold. This gives a contradiction, meaning that this case does not apply. Now, let us assume $\Phi_{(\lambda x.p)v}$ ends with ($@_{\mathtt{p}}^2$). Then, $\tau_2 = \mathtt{n}$. However, according to Lemma 49 $\tau_2 \neq \mathtt{n}$ should holds. This gives a contradiction, meaning that this case does not apply. Finally, let us assume $\Phi_{(\lambda x.p)v}$ ends with rule (@). Then, $\Phi_{(\lambda x.p)v}$ must be of the following form:

$$\dfrac{\dfrac{\dfrac{\Phi_p \triangleright \Gamma_p; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_{\lambda x.p}, m_{\lambda x.p}, d_{\lambda x.p})} p : \mathscr{T} \Rightarrow \kappa}{\Gamma_{\lambda x.p} \Vdash^{(b_{\lambda x.p}, m_{\lambda x.p}, d_{\lambda x.p})} \lambda x.p : \mathscr{M} \to (\mathscr{T} \Rightarrow \kappa)}(\lambda)}{\Gamma_{\lambda x.p} \Vdash^{(b_{\lambda x.p}, m_{\lambda x.p}, d_{\lambda x.p})} \lambda x.p : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T} \Rightarrow \kappa)) \times \mathscr{T})}(\uparrow) \quad \dfrac{\Phi'_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}}{\Phi_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{T} \Rightarrow (\mathscr{M} \times \mathscr{T})}(\uparrow)}{\Gamma_p + \Gamma_v \Vdash^{(1+b_{\lambda x.p}+b_v, m_{\lambda x.p}+m_v, d_{\lambda x.p}+d_v)} (\lambda x.p)v : \mathscr{T} \Rightarrow \kappa}(@)$$

where $\Gamma_{(\lambda x.p)v} = \Gamma_{\lambda x.p} + \Gamma_v$, $\mathscr{T} = \mathscr{T}'' = \mathscr{T}'$ by Lemma 15, $\tau_1 = \mathscr{M} \to (\mathscr{T} \Rightarrow \kappa)$, $\tau_2 = \mathscr{M}$, $b = 1 + b_{\lambda x.p} + b_v + b_s$, $m = m_{\lambda x.p} + m_v + m_s$, and $d = d_{\lambda x.p} + d_v + d_s$. By Lemma 50, there exists $\Phi_{p\{x\setminus v\}} \triangleright \Gamma_{\lambda x.p} + \Gamma_v \Vdash_{\mathtt{v}}^{(b_p+b_v, m_p+m_v, d_p+d_v)} p\{x \setminus v\} : \mathscr{T} \Rightarrow \kappa$. Therefore, we can build $\Phi_{(p\{x\setminus v\},s)}$ as follows:

$$\frac{\Phi_{p\{x\setminus v\}} \triangleright \Gamma_{\lambda x.p} + \Gamma_v \Vdash_{\mathtt{v}}^{(b_v+b_p, m_v+m_p, d_v+d_p)} p\{x \setminus v\} : \mathscr{T} \Rightarrow \kappa \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}}{(\Gamma_{\lambda x.p} + \Gamma_v) + \Delta_s \Vdash^{(b_v+b_p+b_s, m_v+m_p+m_s, d_v+d_p+d_s)} (p\{x \setminus v\}, s) : \kappa}(\mathtt{conf})$$

where $b' = b_v + b_{\lambda x.p} + b_s$, $m' = m_v + m_{\lambda x.p} + m_s$, and $d' = d_v + d_{\lambda x.p} + d_s$. Thus, we can conclude with $\Gamma = \Gamma_{(\lambda x.p)v} + \Delta_s = (\Gamma_{\lambda x.p} + \Gamma_v) + \Delta_s$, $b = b_{(\lambda x.p)v} + b_s = 1 + b_{\lambda x.p} + b_v + b_s = b' - 1$, $m = m_{(\lambda x.p)v} + m_s = m_{\lambda x.p} + m_v + m_s = m'$, and $d = d_{(\lambda x.p)v} + d_s = d_{\lambda x.p} + d_v + d_s = d'$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case $(t, s) = (\mathtt{get}_l(\lambda x.p), s) \rightsquigarrow_{\mathtt{v}} (p\{x \setminus v\}, s) = (u, q)$, such that $s \equiv \mathrm{upd}_l(v, s')$. Then Φ must be of the following form, where $\mathscr{T} = \{l : \Gamma_p(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0$:

$$\dfrac{\dfrac{\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p, m_p, d_p)} p : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0 \Rightarrow \kappa}{\Gamma_p \backslash\!\backslash x \Vdash^{(b_p, 1+m_p, d_p)} \mathtt{get}_l(\lambda x.p) : \mathscr{T} \Rightarrow \kappa}\,(\mathtt{get}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s, m_s, d_s)} s : \mathscr{T}}{\Gamma_p \backslash\!\backslash x + \Delta_s \Vdash^{(b_p+b_s, 1+m_p+m_s, d_p+d_s)} (\mathtt{get}_l(\lambda x.p), s) : \kappa}\,(\mathtt{conf})$$

  where $\Gamma = \Gamma_p \backslash\!\backslash x + \Delta_s$, $b = b_p + b_s$, $m = 1 + m_p + m_s$, and $d = d_p + d_s$. Since $\Pi_s \triangleright \mathrm{v}\Delta_s \Vdash_{\mathtt{v}}^{(b_s, m_s, d_s)} s : \{l : \Gamma_p(x) \sqcup \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0$, we know, by Lemma 48, that $s \equiv \mathrm{upd}_l(v, s')$, $\Phi_v \triangleright \Delta_v \Vdash_{\mathtt{v}}^{(b_v, m_v, d_v)} v : \Gamma_p(x) \sqcup \mathscr{M}$ and $\Pi_{s'} \triangleright \Delta_{s'} \Vdash_{\mathtt{v}}^{(b_{s'}, m_{s'}, d_{s'})} s' : \{l : \mathscr{L}\}; \mathscr{T}_0$, such that $\Delta_s = \Gamma_v + \Delta_{s'}$, $b_s = b_v + b_{s'}$, $m_s = m_v + m_{s'}$, and $d_s = d_v + d_{s'}$. Therefore, by Lemma 47, there exist $\Phi_v^1 \triangleright \Gamma_v^1 \Vdash_{\mathtt{v}}^{(b_v^1, m_v^1, d_v^1)} v : \Gamma_p(x)$ and $\Phi_v^2 \triangleright \Gamma_v^2 \Vdash_{\mathtt{v}}^{(b_v^2, m_v^2, d_v^2)} v : \mathscr{M}$, such that $\Gamma_v = \Gamma_v^1 + \Gamma_v^2$, $b_v = b_v^1 + b_v^2$, $m_v = m_v^1 + m_v^2$, and $d_v = d_v^1 + d_v^2$. By applying Lemma 50 to Φ_p and Φ_v^1, there exists $\Phi_{p\{x\setminus v\}} \triangleright \Gamma_p \backslash\!\backslash x + \Delta_v^1 \Vdash_{\mathtt{v}}^{(b_p+b_v^1, m_p+m_v^1, d_p+d_v^1)} p\{x \setminus v\} : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0 \Rightarrow \kappa$. Moreover, we can construct an alternative type derivation Π_s' for s as follows:

$$\dfrac{\Phi_v^2 \qquad \Pi_{s'}}{\Gamma_v^2 + \Delta_{s'} \Vdash^{(b_v^2+b_{s'}, m_v^2+m_{s'}, d_v^2+d_{s'})} \mathrm{upd}_l(v, s') : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0}\,(\mathrm{upd})$$

  Let $\Pi_q = \Pi_s'$, such that $q = \mathrm{upd}_l(v, s') = s$ and $\Delta_q = \Gamma_v^2 + \Delta_{s'}$. Therefore, we can build Φ' as follows:

$$\dfrac{\Phi_{p\{x\setminus v\}} \qquad \Pi_q}{(\Gamma_p \backslash\!\backslash x + \Gamma_v^1) + (\Gamma_v^2 + \Delta_{s'}) \Vdash^{(b_p+b_v^1+b_v^2+b_{s'}, m_p+m_v^1+m_v^2+m_{s'}, d_p+d_v^1+d_v^2+d_{s'})} (p\{x \setminus v\}, s) : \kappa}\,(\mathtt{conf})$$

  where $b' = b_p + b_v^1 + b_v^2 + b_{s'}$, $m' = m_p + m_v^1 + m_v^2 + m_{s'}$, and $d' = d_p + d_v^1 + d_v^2 + d_{s'}$. Thus, we can conclude with $\Gamma = \Gamma_p \backslash\!\backslash x + \Delta_s = \Gamma_p \backslash\!\backslash x + (\Gamma_v + \Delta_{s'}) = \Gamma_p \backslash\!\backslash x + (\Gamma_v^1 + \Gamma_v^2) + \Delta_{s'}$, $b = b_p + b_s = b_p + (b_v + b_{s'}) = b_p + (b_v^1 + b_v^2) + b_{s'} = b'$, $m - 1 = (1 + m_p + m_s) - 1 = (1 + m_p + m_v + m_{s'}) - 1 = (1 + m_p + m_v^1 + m_v^2 + m_{s'}) - 1 = m'$, $d = d_p + d_s = d_p + (d_v + d_{s'}) = d_p + (d_v^1 + d_v^2) + d_{s'} = d'$.
- Case $(t, s) = (\mathtt{set}_l(v, p), s) \rightsquigarrow_{\mathtt{v}} (p, \mathrm{upd}_l(v, s)) = (u, q)$. Then, Φ must be of the following form, where $\mathscr{T} = \{l : \mathscr{L}\}; \mathscr{T}_0$:

$$\dfrac{\dfrac{\Phi_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v, m_v, d_v)} v : \mathscr{M} \qquad \Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p, m_p, d_p)} p : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0 \Rightarrow \kappa}{\Gamma_v + \Gamma_p \Vdash^{(b_v+b_p, 1+m_v+m_p, s_v+s_p)} \mathtt{set}_l(v, p) : \mathscr{T} \Rightarrow \kappa}\,(\mathtt{set}) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s, m_s, d_s)} s : \mathscr{T}}{(\Gamma_v + \Gamma_p) + \Delta_s \Vdash^{(b_v+b_p+b_s, 1+m_v+m_p+m_s, d_v+d_p+d_s)} (\mathtt{set}_l(v, p), s) : \kappa}\,(\mathtt{conf})$$

  such that $\Gamma = (\Gamma_v + \Gamma_p) + \Delta_s$, $b = b_v + b_p + b_s$, $m = 1 + m_v + m_p + m_s$ and $d = d_v + d_p + d_s$. Therefore, we can build derivation Π_q for $\mathrm{upd}_l(v, s)$ as follows:

$$\dfrac{\Phi_v \qquad \Pi_s}{\Gamma_v + \Delta_s \Vdash^{(b_v+b_s, m_v+m_s, d_v+d_s)} \mathrm{upd}_l(v, s) : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}_0}\,(\mathrm{upd})$$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

Moreover, we can build Φ' as follows:

$$\frac{\Phi_p \qquad \Pi_q}{\Gamma_p + (\Gamma_v + \Delta_s) \Vdash^{(b_p+b_v+b_s, m_p+m_v+m_s, d_p+d_v+d_s)} (p, \mathtt{upd}_l(v,s)) : \kappa}(\mathtt{conf})$$

such that $b' = b_p + b_v + b_s$, $m' = m_p + m_v + m_s$, and $d' = d_p + d_v + d_s$. Thus, we can conclude with $\Gamma = (\Gamma_v + \Gamma_p) + \Delta_s$, $b = b_v + b_p + b_s = b'$, $m - 1 = (1 + m_v + m_p + m_s) - 1 = m_v + m_p + m_s = m'$, and $d = d_v + d_p + d_s = d'$.

- Case $(t,s) = (pr, s) \rightsquigarrow_{\mathtt{v}} (p'r, q) = (u, q)$, such that $(p,s) \rightsquigarrow_{\mathtt{v}} (p', q)$. Then, Φ has two premises Φ_{pr} and $\Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s, m_s, d_s)} s : \mathscr{T}$, and Φ_{pr} must end with rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@^2_{\mathtt{p}})$:
  - Case Φ_{pr} ends with rule $(@)$. Then, Φ_{pr} must be of the following form:

$$\dfrac{\dfrac{\Phi_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p, m_p, d_p)} p : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}') \quad \Phi_r \triangleright \Gamma_r \Vdash_{\mathtt{v}}^{(b_r, m_r, d_r)} r : \mathscr{T}' \Rightarrow (\mathscr{M} \times \mathscr{T}'')}{\Gamma_p + \Gamma_r \Vdash^{(1+b_p+b_r, m_p+m_r, d_p+d_r)} pr : \mathscr{T} \Rightarrow \kappa}(@) \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s, m_s, d_s)} s : \mathscr{T}}{(\Gamma_p + \Gamma_r) + \Delta_s \Vdash^{(1+b_p+b_r+b_s, m_p+m_r+m_s, d_p+d_r+d_s)} (pr, s) : \kappa}(\mathtt{conf})$$

    where $\Gamma = (\Gamma_p + \Gamma_r) + \Delta_s$ is tight, $b = 1 + b_p + b_r + b_s$, $m = m_p + m_r + m_s$, and $d = d_p + d_r + d_s$. Therefore, we can build the following derivation $\Phi_{(p,s)}$:

$$\frac{\Phi_p \qquad \Pi_s}{\Gamma_p + \Delta_s \Vdash^{(b_p+b_s, m_p+m_s, d_p+d_s)} (p,s) : (\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}'}(\mathtt{conf})$$

    Since Γ is tight, then Γ_p and Δ_s are both tight. Moreover, $\neg\mathtt{val}(p)$, since $(p,s) \rightsquigarrow_{\mathtt{v}} (p', q)$. By applying the *i.h.* to $\Phi_{(p,s)}$, there exists $\Phi_{(p',q)}$ that must be of the following form:

$$\frac{\Phi_{p'} \triangleright \Gamma_{p'} \Vdash_{\mathtt{v}}^{(b_{p'}, m_{p'}, d_{p'})} p' : \mathscr{T}''' \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}') \quad \Pi_q \triangleright \Delta_q \Vdash_{\mathtt{v}}^{(b_q, m_q, d_q)} q : \mathscr{T}'''}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'}+b_q, m_{p'}+m_q, d_{p'}+d_q)} (p', q) : (\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}'}(\mathtt{conf})$$

    where $\Gamma_p + \Delta_s = \Gamma_{p'} + \Delta_q$, and the counters $(b_p + b_s, m_p + m_s, d_p + d_s)$ and $(b_{p'} + b_q, m_{p'} + m_q, d_{p'} + d_q)$ are related properly. Therefore, we can build Φ' as follows:

$$\dfrac{\dfrac{\Phi_{p'} \qquad \Phi_r}{\Gamma_{p'} + \Gamma_r \Vdash^{(1+b_{p'}+b_r, m_{p'}+m_r, d_{p'}+d_r)} p'r : \mathscr{T}''' \Rightarrow \kappa}(@) \qquad \Pi_q}{(\Gamma_{p'} + \Gamma_r) + \Delta_q \Vdash^{(1+b_{p'}+b_r+b_q, m_{p'}+m_r+m_q, d_{p'}+d_r+d_q)} (p'r, q) : \kappa}(\mathtt{conf})$$

    where $b' = 1 + b_{p'} + b_r + b_q$, $m' = m_{p'} + m_r + m_q$, and $d' = d_{p'} + d_r + d_q$. Thus, we can conclude with $\Gamma = (\Gamma_p + \Delta_s) + \Gamma_r = (\Gamma_{p'} + \Delta_q) + \Gamma_r$, and $(1 + b_p + b_r + b_s, m_p + m_r + m_s, d_p + d_r + d_s) = (b, m, d)$ and $(b', m', d') = (1 + b_{p'} + b_r + b_q, m_{p'} + m_r + m_q, d_{p'} + d_r + d_q)$ are properly related.
  - Case Φ_{pr} ends with $(@^1_{\mathtt{p}})$ or $(@^2_{\mathtt{p}})$. These cases are very similar to the previous case and follow easily from the *i.h.*.
- Case $(t,s) = (pr, s) \rightsquigarrow_{\mathtt{v}} (pr', q) = (u, q)$, such that $(p,s) \not\rightsquigarrow_{\mathtt{v}}$, (p,s) is unblocked, and $(r,s) \rightsquigarrow_{\mathtt{v}} (r', q)$. Then, Φ has two premises Φ_{pr} and $\Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s, m_s, d_s)} s : \mathscr{T}$, such that $\Gamma = \Gamma_{pr} + \Delta_s$,

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$b = 1 + b_{pr} + b_s$, $m = m_{pr} + m_s$, and $d = d_{pr} + d_s$. Then, Φ_{pr} must end with rule (@), ($@^1_{\mathrm{p}}$), or ($@^2_{\mathrm{p}}$):

– Case Φ_{pr} ends with (@). Then, Φ_{pr} must be of the following form:

$$\frac{\Phi_p \triangleright \Gamma_p \Vdash_{\mathrm{v}}^{(b_p,m_p,d_p)} p : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \times \kappa)) \times \mathscr{T}') \quad \Phi_r \triangleright \Gamma_r \Vdash_{\mathrm{v}}^{(b_r,m_r,d_r)} r : \mathscr{T}' \Rightarrow (\mathscr{M} \times \mathscr{T}'')}{\Gamma_p + \Gamma_r \Vdash^{(1+b_p+b_r,m_p+m_r,d_p+d_r)} pr : \mathscr{T} \Rightarrow \kappa}\ (@)$$

where $\Gamma_{pr} = \Gamma_p + \Gamma_r$, $b = 1 + b_p + b_r + b_s$, $m = m_p + m_r + m_s$, and $d = d_p + d_r + d_s$. Notice that, since $(p, s) \not\rightsquigarrow_{\mathrm{v}}$ and (p, s) is unblocked, then $p \in \mathtt{no}_{\mathrm{v}}$, by definition. Therefore, $\mathscr{T} = \mathscr{T}'$, according to Lemma 15. And, we can build derivation $\Phi_{(r,s)}$ as follows:

$$\frac{\Phi_r \qquad \Pi_s}{\Gamma_r + \Delta_s \Vdash^{(b_r+b_s,m_r+m_s,d_r+d_s)} (r, s) : (\mathscr{M} \times \mathscr{T}'')}(\mathtt{conf})$$

Since Γ is tight, then Γ_r and Δ_s are both tight. Moreover, $\neg\mathtt{val}(r)$, since $(r, s) \rightsquigarrow_{\mathrm{v}} (r', q)$. By applying the *i.h.* to $\Phi_{(r,s)}$, there exists $\Phi_{(r',q)}$ that must be of the following form:

$$\frac{\Phi_{r'} \triangleright \Gamma_{r'} \Vdash_{\mathrm{v}}^{(b_{r'},m_{r'},d_{r'})} r' : \mathscr{T}''' \Rightarrow (\mathscr{M} \times \mathscr{T}'') \quad \Pi_q \triangleright \Delta_q \Vdash_{\mathrm{v}}^{(b_q,m_q,d_q)} q : \mathscr{T}'''}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'}+b_q,m_{p'}+m_q,d_{p'}+d_q)} (r', q) : (\mathscr{M} \times \mathscr{T}'')}\ (\mathtt{conf})$$

where $\Gamma_r + \Delta_s = \Gamma_{r'} + \Delta_q$, and the counters $(b_r + b_s, m_r + m_s, d_r + d_s)$ and $(b_{r'} + b_q, m_{r'} + m_q, d_{r'} + d_q)$ are related properly. Moreover, there exists $\Phi'_p \triangleright \Gamma_p \Vdash_{\mathrm{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}''' \Rightarrow ((\mathscr{M} \rightsquigarrow_{\mathrm{v}} (\mathscr{T}'' \times \kappa)) \times \mathscr{T}''')$, by Lemma 15. Therefore, we can build Φ' as follows:

$$\frac{\dfrac{\Phi_p \qquad \Phi_{r'}}{\Gamma_p + \Gamma_{r'} \Vdash^{(1+b_p+b_{r'},m_p+m_{r'},d_p+d_{r'})} pr' : \mathscr{T}''' \Rightarrow \kappa}\ (@) \qquad \Pi_q}{(\Gamma_p + \Gamma_{r'}) + \Delta_q \Vdash^{(1+b_p+b_{r'}+b_q,m_p+m_{r'}+m_q,d_p+d_{r'}+d_q)} (pr', q) : \kappa}\ (\mathtt{conf})$$

where $b' = 1 + b_p + b_{r'} + b_q$, $m' = m_p + m_{r'} + m_q$, and $d' = d_p + d_{r'} + d_q$. Thus, we can conclude with $\Gamma = (\Gamma_p + \Delta_s) + \Gamma_r = (\Gamma_p + \Delta_q) + \Gamma_{r'}$, and $(1 + b_p + b_r + b_s, m_p + m_r + m_s, d_p + d_r + d_s) = (b, m, d)$ and $(b', m', d') = (1 + b_p + b_{r'} + b_q, m_p + m_{r'} + m_q, d_p + d_{r'} + d_q)$ are properly related.

– Case Φ_{pr} ends with ($@^1_{\mathrm{p}}$) or ($@^2_{\mathrm{p}}$). These two cases are very similar to the previous case. □

Theorem 5 (Quantitative soundness). *If* $\Phi \triangleright \Gamma \Vdash_{\mathrm{v}}^{(b,m,d)} (t, s) : \kappa$ *is a tight derivation, then there exists* (u, q) *such that* $u \in \mathtt{no}_{\mathrm{v}}$ *and* $(t, s) \rightsquigarrow_{\mathrm{v}}^{(b,m)} (u, q)$ *with* b β_{v}*-steps,* m $\mathtt{g}/\mathtt{s}$*-steps, and* $|(u, q)|_{\mathrm{v}} = d$.

Proof. We start by noting that Φ has two premises $\Phi_t \triangleright \Gamma_t \Vdash_{\mathrm{v}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$ and $\Pi_s \triangleright \Delta_s \Vdash_{\mathrm{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}$, where $\Gamma = \Gamma_t + \Delta_s$ and $\kappa = \tau \times \mathscr{T}'$, such that τ and $\mathscr{T}'$ are tight. Now, the proof follows by induction on $b + m$:

- Case $b + m = 0$. Then, $b = m = 0$, and thus, $b_t = m_t = b_s = m_s = 0$. Therefore, (t, s) is a configuration in normal form, by Lemma 17, and $|(t, s)|_{\mathrm{v}} = d$, by Lemma 16. Thus, we can conclude by taking $u = t$ and $q = s$.

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

- Case $b + m > 0$. Then, $b > 0$ or $m > 0$, and thus, $b_t + b_s > 0$ or $m_t + m_s > 0$. Note that it cannot be that $b_t = m_t = 0$, because this would imply that $b = m = 0$ by Lemma 16. Therefore, it must be the case that $b_t + m_t > 0$, and thus, $t \notin \mathtt{no}_{\mathtt{v}}$ by Lemma 45. Moreover, note that (t, s) cannot be a blocked configuration, by Proposition 6. Therefore, by Proposition 5, there exists (t', s') such that $(t, s) \rightsquigarrow_{\mathtt{v}} (t', s')$. By Lemma 19, there exists $\Phi' \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b',m',d)} (t', s') : \kappa$, such that $b' + m' = b + m - 1$. By the *i.h.*, there exists (u, q), such that $u \in \mathtt{no}_{\mathtt{v}}$, $(t', s') \rightsquigarrow_{\mathtt{v}}^{(b',m')} (u, q)$ and $d = |(u, q)|_{\mathtt{v}}$. So we can conclude with $(t, s) \rightsquigarrow_{\mathtt{v}} (t', s') \rightsquigarrow_{\mathtt{v}}^{(b',m')} (u, q)$, which means that $(t, s) \rightsquigarrow_{\mathtt{v}}^{(b,m)} (u, q)$, as expected. □

Lemma 51 (Anti-substitution). *If* $\Phi_{t\{x\setminus v\}} \triangleright \Gamma_{t\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b,m,d)} t\{x \setminus v\} : \mu$, *there exist* $\Phi_t \triangleright \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mu$ *and* $\Phi_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, *such that* $\Gamma_{t\{x\setminus v\}} = \Gamma_t + \Gamma_v$, $b = b_t + b_v$, $m = m_t + m_v$, *and* $d = d_t + d_v$.

Proof. We generalize the original statement by allowing $\Phi_{t\{x\setminus v\}}$ to conclude with either a monadic type μ or a type τ. Let $\mathscr{G} \in \{\mu, \tau\}$. Then the statement is as follows:

If $\Phi_{t\{x\setminus v\}} \triangleright \Gamma_{t\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b,m,d)} t\{x \setminus v\} : \mathscr{G}$, there exist $\Phi_t \triangleright \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mathscr{G}$ and $\Phi_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_{t\{x\setminus v\}} = \Gamma_t + \Gamma_v$, $b = b_t + b_v$, $m = m_t + m_v$, and $d = d_t + d_v$.

The proof is by induction over $\Phi_{t\{x\setminus v\}}$, according to the last rule used in $\Phi_{t\{x\setminus v\}}$:

- Case $\mathscr{G} = \mu$. Let us assume $t = x$, such that $t\{x \setminus v\} = v$. Then $\mu = \mathscr{T} \Rightarrow (\tau \times \mathscr{T}')$, such that $\mathscr{T} = \mathscr{T}'$ by the fact that $t\{x \setminus v\} = v$, and thus, $\Phi_{t\{x\setminus v\}}$ must finish with rule (↑), and $\tau \neq \mathtt{n}$ by Lemma 49. Moreover, $\Phi_{t\{x\setminus v\}}$ must necessarily end with rule (↑) with one premise $\Phi'_{t\{x\setminus v\}} \triangleright \Gamma_{t\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b,m,d)} t\{x \setminus v\} : \pi$, such that $\tau = \pi$. By applying the *i.h.* to $\Phi'_{t\{x\setminus v\}}$, there exist $\Phi'_t \triangleright \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \pi$ and $\Phi_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_{t\{x\setminus v\}} = \Gamma_t + \Gamma_v$, $b = b_t + b_v$, $m = m_t + m_v$, and $d = d_t + d_v$. Therefore, we can build $\Phi_t \triangleright \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\pi \times \mathscr{T})$, by applying rule (↑) to Φ'_t. We can conclude since rule (↑) leaves the typing environment and the counters of the premise unchanged. From now on, we are going to assume $t \neq x$ and consider the last rule used in $\Phi_{t\{x\setminus v\}}$, which must be (↑), (@), (get), (set), ($@^1_{\mathtt{p}}$), or ($@^2_{\mathtt{p}}$):
  - Rule (↑). Then $\mu = \mathscr{T} \Rightarrow (\tau \times \mathscr{T})$ and $\Phi_{t\{x\setminus v\}}$ has one premise $\Phi'_{t\{x\setminus v\}} \triangleright \Gamma_{t\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b,m,d)} t\{x \setminus v\} : \tau$. By applying the *i.h.* to $\Phi'_{t\{x\setminus v\}}$, there exist $\Phi'_t \triangleright \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \tau$ and $\Phi_v \triangleright \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_{t\{x\setminus v\}} = \Gamma_t + \Gamma_v$, $b = b_t + b_u$, $m = m_t + m_u$, and $d = d_t + d_u$. Therefore, we can build $\Phi_t \triangleright \Gamma_t; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_t,m_t,d_t)} t : \mathscr{T} \Rightarrow (\tau \times \mathscr{T})$ by taking Φ'_t and applying rule (↑). And we can conclude since rule (↑) leaves the typing environment and the counters of the premise unchanged.
  - Rule (@). Then $t = up$, $\mu = \mathscr{T} \Rightarrow \kappa$, and $t\{x \setminus v\} = (u\{x \setminus v\})(p\{x \setminus v\})$. Moreover, $\Phi_{t\{x\setminus v\}}$ has two premises $\Phi_{u\{x\setminus v\}} \triangleright \Gamma_{u\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u\{x \setminus v\} : \mathscr{T} \Rightarrow ((\mathscr{M}' \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'')$ and $\Phi_{p\{x\setminus v\}} \triangleright \Gamma_{p\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p\{x \setminus v\} : \mathscr{T}'' \Rightarrow (\mathscr{M}' \times \mathscr{T}')$, where $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} + \Gamma_{p\{x\setminus v\}}$, $b = 1 + b_u + b_p$, $m = m_u + m_p$, and $d = d_u + d_p$. By applying the *i.h.* to $\Phi_{u\{x\setminus v\}}$ and $\Phi_{p\{x\setminus v\}}$, there exist $\Phi_u \triangleright \Gamma_u; x : \mathscr{M}_1 \Vdash_{\mathtt{v}}^{(b'_u,m'_u,d'_u)} u : \mathscr{T} \Rightarrow ((\mathscr{M}' \to (\mathscr{T}' \Rightarrow \kappa)) \times \mathscr{T}'')$ and $\Phi^1_v \triangleright \Gamma^1_v \Vdash_{\mathtt{v}}^{(b^1_v,m^1_v,d^1_v)} v : \mathscr{M}_1$, such that $\Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma_v$, $b_u = b'_u + b^1_v$, $m_u = m'_u + m^1_v$, and $d_u = d'_u + d^1_v$, and $\Phi_p \triangleright \Gamma_p; x : \mathscr{M}_2 \Vdash_{\mathtt{v}}^{(b'_p,m'_p,d'_p)} p : \mathscr{T}'' \Rightarrow (\mathscr{M} \times \mathscr{T}')$ and $\Phi^2_v \triangleright \Gamma^2_v \Vdash_{\mathtt{v}}^{(b^2_v,m^2_v,d^2_v)} v : \mathscr{M}_2$, such that $\Gamma_{p\{x\setminus v\}} = \Gamma_p + \Gamma^2_v$, $b_p = b'_p + b^2_v$, $m_p = m'_p + m^2_v$, and $d_p =$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$d'_p + d^2_v$. By Lemma 47, there exists $\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_v = \Gamma^1_v + \Gamma^2_v$, $b_v = b^1_v + b^2_v$, $m_v = m^1_v + m^2_v$, and $d_v = d^1_v + d^2_v$. And we can build Φ_t as follows:

$$\frac{\Phi_u \qquad \Phi_p}{(\Gamma_u + \Gamma_p); x : \mathscr{M} \Vdash^{(1+b'_u+b'_p, m'_u+m'_p, d'_u+d'_p)} up : \mathscr{T} \Rightarrow \kappa}(@)$$

where $\Gamma_t = \Gamma_u + \Gamma_p$, $b_t = 1 + b_u + b_p$, $m_t = m_u + m_p$, and $d_t = d_u + d_p$. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} + \Gamma_{p\{x\setminus v\}} = (\Gamma_u + \Gamma^1_v) + (\Gamma_p + \Gamma^2_v) = \Gamma_t + \Gamma_v$, $b = 1 + b_u + b_p = 1 + (b'_u + b^1_v) + (b'_p + b^2_v) = 1 + b_u + b_p + b_v = b_t + b_v$, $m = m_u + m_p = (m'_u + m^1_v) + (m'_p + m^2_v) = m_u + m_p + m_v = m_t + m_v$, and $d = d_u + d_p = (d'_u + d^1_v) + (d'_p + d^2_v) = d_u + d_p + d_v = d_t + d_v$.

- Rule (get). Then, $t = \mathtt{get}_l(\lambda y.u)$, $\mu = \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}; \mathscr{T}\} \Rightarrow \kappa$, and $t\{x \setminus v\} = \mathtt{get}_l(\lambda y.(u\{x \setminus v\}))$. Moreover, $\Phi_{t\{x\setminus v\}}$ has one premise $\Phi_{u\{x\setminus v\}} \rhd \Gamma_{u\{x\setminus v\}}; y : \mathscr{M}' \Vdash_{\mathtt{v}}^{(b,m',d)} u\{x \setminus v\} : \{l : \mathscr{M}'' \cdot \mathscr{L}; \}\mathscr{T} \Rightarrow \kappa$, such that $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}}$ and $m = 1 + m'$. By applying the *i.h.* to $\Phi_{u\{x\setminus v\}}$, there exist $\Phi_u \rhd (\Gamma_u; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \{l : \mathscr{M}'' \cdot \mathscr{L}; \}\mathscr{T} \Rightarrow \kappa$ and $\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma_v$, $b = b_u + b_v$, $m' = m_u + m_v$, and $d = d_u + d_v$. Therefore, we can build Φ_t as follows:

$$\frac{\Phi_u \rhd \Gamma_u; y : \mathscr{M}'; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \{l : \mathscr{M}'' \cdot \mathscr{L}; \}\mathscr{T} \Rightarrow \kappa}{\Gamma_u;\ x : \mathscr{M} \Vdash^{(b_u,1+m_u,d_u)} \mathtt{get}_l(\lambda y.u) : \{l : \mathscr{M}' \sqcup \mathscr{M}'' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{get})$$

And we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma_v$, $b = b_u + b_v$, $m = 1 + m' = 1 + m_u + m_v = (1 + m_u) + m_v$, and $d = d_u + d_v$.

- Rule (set). Then, $t = \mathtt{set}_l(w, u)$, $\mu = \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, and $t\{x \setminus v\} = \mathtt{set}_l(w\{x \setminus v\}, u\{x \setminus v\})$. Moreover, $\Phi_{t\{x\setminus v\}}$ has two premises $\Phi_{w\{x\setminus v\}} \rhd \Gamma_{w\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b',m',d')} w\{x \setminus v\} : \mathscr{M}'$ and $\Phi_{u\{x\setminus v\}} \rhd \Gamma_{u\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b'',m'',d'')} u\{x \setminus v\} : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$, such that $\Gamma_{t\{x\setminus v\}} = \Gamma_{w\{x\setminus v\}} + \Gamma_{u\{x\setminus v\}}$, $b = b' + b''$, $m = 1 + m' + m''$, and $d = d' + d''$. By applying the *i.h.* to $\Phi_{w\{x\setminus v\}}$, there exist $\Phi_w \rhd \Gamma_w; x : \mathscr{M}_1 \Vdash_{\mathtt{v}}^{(b_w,m_w,d_w)} w : \mathscr{M}'$ and $\Phi^1_v \rhd \Gamma^1_v \Vdash_{\mathtt{v}}^{(b^1_v,m^1_v,d^1_v)} v : \mathscr{M}_1$, such that $\Gamma_{w\{x\setminus v\}} = \Gamma_w + \Gamma^1_v$, $b' = b_w + b^1_v$, $m' = m_w + m^1_v$, and $d' = d_w + d^1_v$. By applying the *i.h.* to $\Phi_{u\{x\setminus v\}}$, there exist $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_2 \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \{l : \mathscr{M}' \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$ and $\Phi^2_v \rhd \Gamma^2_v \Vdash_{\mathtt{v}}^{(b^2_v,m^2_v,d^2_v)} v : \mathscr{M}_2$, such that $\Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma^2_v$, $b'' = b_u + b^2_v$, $m'' = m_u + m^2_v$, and $d'' = d_u + d^2_v$. By Lemma 47, there exists $\Phi_v \rhd \Gamma^1_v + \Gamma^2_v \Vdash_{\mathtt{v}}^{(b^1_v+b^2_v, m^1_v+m^2_v, d^1_v+d^2_v)} v : \mathscr{M}$, such that $\Gamma_v = \Gamma^1_v + \Gamma^2_v$, $b_v = b^1_v + b^2_v$, $m_v = m^1_v + m^2_v$, and $d_v = d^1_v + d^2_v$. Therefore, we can build Φ_t as follows:

$$\frac{\Phi_w \qquad \Phi_u}{(\Gamma_w + \Gamma_u); x : \mathscr{M} \Vdash^{(b_w+b_u, 1+m_w+m_u, d_w+d_u)} \mathtt{set}_l(w, u) : \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{set})$$

where $\Gamma_t = \Gamma_w + \Gamma_u$, $b_t = b_w + b_u$, $m_t = 1 + m_w + m_u$, and $d_t = d_w + d_u$. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_{w\{x\setminus v\}} + \Gamma_{u\{x\setminus v\}} = (\Gamma_w + \Gamma^1_u) + (\Gamma_u + \Gamma^2_v) = (\Gamma_w + \Gamma_u) + (\Gamma^1_v + \Gamma^2_v) = \Gamma_t + \Gamma_v$, $b = b' + b'' = (b_w + b^1_v) + (b_u + b^2_v) = (b_w + b_u) + (b^1_v + b^2_v) = b_t + b_v$, $m = 1 + m' + m'' = 1 + (m_w + m^1_v) + (m_u + m^2_v) = (1 + m_w + m_u) + (m^1_v + m^2_v) = m_t + m_v$, and $d = d' + d'' = (d_w + d^1_v) + (d_u + d^2_v) = (d_w + d_u) + (d^1_v + d^2_v) = d_t + d_v$.

- Rule ($@^1_{\mathtt{p}}$). Then, $t = up$, $\mu = \mathscr{T} \Rightarrow (\mathtt{n} \times \mathscr{T}')$, and $t\{x \setminus v\} = (u\{x \setminus v\})(p\{x \setminus v\})$. Moreover, $\Phi_{t\{x\setminus v\}}$ has two premises $\Phi_{u\{x\setminus v\}} \rhd \Gamma_{u\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u\{x \setminus v\} : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}'')$ and $\Phi_{p\{x\setminus v\}} \rhd \Gamma_{p\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p\{x \setminus v\} : \mathscr{T}'' \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')$, such that $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} + \Gamma_{p\{x\setminus v\}}$, $b = b_u + b_p$, $m = m_u + m_p$, and $d = 1 + d_u + d_p$. By applying the *i.h.* to $\Phi_{u\{x\setminus v\}}$ and $\Phi_{p\{x\setminus v\}}$, there exist $\Phi_u \rhd \Gamma_u; x : \mathscr{M}_1 \Vdash_{\mathtt{v}}^{(b'_u,m'_u,d'_u)} u : \mathscr{T} \Rightarrow (\overline{\mathtt{a}} \times \mathscr{T}'')$ and $\Phi^1_v \rhd$

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

$\Gamma_v^1 \Vdash_{\mathtt{v}}^{(b_v^1,m_v^1,d_v^1)} v : \mathscr{M}_1$, such that $\Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma_v$, $b_u = b'_u + b_v^1$, $m_u = m'_u + m_v^1$, and $d_u = d'_u + d_v^1$, and $\Phi_p \rhd \Gamma_p; x : \mathscr{M}_2 \Vdash_{\mathtt{v}}^{(b'_p,m'_p,d'_p)} p : \mathscr{T}'' \Rightarrow (\mathtt{tt}_{\mathtt{v}} \times \mathscr{T}')$ and $\Phi_v^2 \rhd \Gamma_v^2 \Vdash_{\mathtt{v}}^{(b_v^2,m_v^2,d_v^2)} v : \mathscr{M}_2$, such that $\Gamma_{p\{x\setminus v\}} = \Gamma_p + \Gamma_v^2$, $b_p = b'_p + b_v^2$, $m_p = m'_p + m_v^2$, and $d_p = d'_p + d_v^2$. By Lemma 47, there exists $\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_v = \Gamma_v^1 + \Gamma_v^2$, $b_v = b_v^1 + b_v^2$, $m_v = m_v^1 + m_v^2$, and $d_v = d_v^1 + d_v^2$. Moreover, we can build Φ_t as follows:

$$\frac{\Phi_u \qquad \Phi_p}{(\Gamma_u + \Gamma_p); x : \mathscr{M} \Vdash^{(b'_u+b'_p,m'_u+m'_p,1+d'_u+d'_p)} up : \mathscr{T} \Rightarrow \kappa}\,(@_{\mathtt{p}}^1)$$

where $\Gamma_t = \Gamma_u + \Gamma_p$, $b_t = b_u + b_p$, $m_t = m_u + m_p$, and $d_t = 1 + d_u + d_p$. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} + \Gamma_{p\{x\setminus v\}} = (\Gamma_u + \Gamma_v^1) + (\Gamma_p + \Gamma_v^2) = \Gamma_t + \Gamma_v$, $b = b_u + b_p = (b'_u + b_v^1) + (b'_p + b_v^2) = b_u + b_p + b_v = b_t + b_v$, $m = m_u + m_p = (m'_u + m_v^1) + (m'_p + m_v^2) = m_u + m_p + m_v = m_t + m_v$, and $d = 1 + d_u + d_p = 1 + (d'_u + d_v^1) + (d'_p + d_v^2) = 1 + d_u + d_p + d_v = d_t + d_v$.

– Rule $(@_{\mathtt{p}}^2)$. This case is very similar to the previous case.

- Case $\mathscr{G} = \tau$. Let us assume $t = x$, such that $t\{x \setminus v\} = v$. Then we have to consider two cases: $\tau = [\sigma_i]_{i\in I}$ or τ is not a multi-type. If $\tau = [\sigma_i]_{i\in I}$, we can take $\Phi_v = \Phi_{t\{x\setminus v\}}$, such that $\Gamma_v = \Gamma_{t\{x\setminus v\}}$, $b_v = b$, $m_v = m$, and $d_v = d$, and build Φ_t as follows:

$$\frac{\left(\dfrac{}{x : [\sigma_i] \Vdash^{(0,0,0)} x : \sigma_i}\,(\mathtt{ax})\right)_{i\in I}}{x : [\sigma_i]_{i\in I} \Vdash^{(0,0,0)} x : [\sigma_i]_{i\in I}}\,(\mathtt{m})$$

such that $\Gamma_t = \emptyset$, $\mathscr{M} = [\sigma_i]_{i\in I}$, and $b_t = m_t = d_t = 0$. If τ is not a multi-type, then we can build $\Phi_v \rhd \Gamma_{t\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b,m,d)} v : [\tau]$, such that $\Gamma_v = \Gamma_{t\{x\setminus v\}}$, $b_v = b$, $m_v = m$, and $d_v = d$, from $\Phi_{t\{x\setminus v\}}$ followed by rule $(\mathtt{m})$, and $\Phi_t \rhd x : [\tau] \Vdash_{\mathtt{v}}^{(0,0,0)} x : \tau$, such that $\Gamma_t = \emptyset$, $b_t = m_t = d_t = 0$, by using rule $(\mathtt{ax})$. In both cases, we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_v = \Gamma_t + \Gamma_v$, $b = b_v = b_t + b_v$, $m = m_v = m_t + m_v$, and $d = d_v = d_t + d_v$. From now on, we are going to assume $t \neq x$ and consider the last rule used in $\Phi_{t\{x\setminus v\}}$, which must be $(\mathtt{ax})$, (λ), or $(\lambda_{\mathtt{p}})$:

  – Rule $(\mathtt{ax})$. Then $t = y$, $\tau = \sigma$, $y\{x \setminus v\} = y$, and $\mathscr{M} = [\,]$. Therefore, we can take $\Phi_t = \Phi_{t\{x\setminus v\}}$, where $\Gamma_t = \Gamma_{t\{x\setminus v\}}$, $b_t = b$, $m_t = m$, and $d_t = d$, and build $\Phi_v \rhd \emptyset \Vdash_{\mathtt{v}}^{(0,0,0)} v : [\,]$, such that $\Gamma_v = \emptyset$, and $b_v = m_v = d_v = 0$, by using rule $(\mathtt{m})$ with no premises. And we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_t = \Gamma_t + \Gamma_v$, $b = b_t = b_t + b_v$, $m = m_t = m_t + m_v$, and $d = d_t = d_t + d_v$.
  – Rule (λ). Then $t = \lambda y.u$, $\tau = \mathscr{M}' \to \mu$, and $t\{x \setminus v\} = \lambda y.(u\{x \setminus v\})$. Therefore, $\Phi_{t\{x\setminus v\}}$ has one premise $\Phi_{u\{x\setminus v\}} \rhd \Gamma_{u\{x\setminus v\}}; y : \mathscr{M}' \Vdash_{\mathtt{v}}^{(b,m,d)} u\{x \setminus v\} : \mu$, where $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}}$. By applying the *i.h.* to $\Phi_{u\{x\setminus v\}}$, there exist $\Phi_u \rhd (\Gamma_u; y : \mathscr{M}'); x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} u : \mu$ and $\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma_v$, since we may assume that $y \notin \mathtt{dom}(\Gamma_v)$ by α-equivalence, $b = b_u + b_v$, $m = m_u + m_v$, and $d = d_u + d_v$. Therefore, we can build $\Phi_t \rhd \Gamma_u; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_u,m_u,d_u)} \lambda y.u : \mathscr{M}' \to \mu$, such that $\Gamma_t = \Gamma_u$, $b_t = b_u$, $m_t = m_u$, and $d_t = d_u$, by taking Φ'_t and applying rule (λ). And we can conclude with $\Gamma_{t\{x\setminus v\}} = \Gamma_{u\{x\setminus v\}} = \Gamma_u + \Gamma_v$, $b = b_u + b_v$, $m = m_u + m_v$, and $d = d_u + d_v$.
  – Rule $(\mathtt{m})$. Then $\tau = [\sigma_i]_{i\in I}$ and $\Phi'_{t\{x\setminus v\}}$ has premises $(\Phi^i_{t\{x\setminus v\}} \rhd \Gamma_i \Vdash_{\mathtt{v}}^{(b_i,m_i,d_i)} t\{x \setminus v\} : \sigma_i)_{i\in I}$, where $\Gamma_{t\{x\setminus v\}} = +_{i\in I}\Gamma_i$, $b = +_{i\in I}b_i$, $m = +_{i\in I}m_i$, and $d = +_{i\in I}d_i$. By applying the *i.h.* to each $\Phi^i_{t\{x\setminus v\}}$, there exist $\Phi^i_t \rhd \Gamma^i_t; x : \mathscr{M}_i \Vdash_{\mathtt{v}}^{(b_t^i,m_t^i,d_t^i)} t : \sigma_i$ and $\Phi^i_v \rhd \Gamma^i_v \Vdash_{\mathtt{v}}^{(b_v^i,m_v^i,d_v^i)} v : = \mathscr{M}_i$,

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

such that $\Gamma_i = \Gamma_t^i + \Gamma_v^i$, $\mathscr{M} = \sqcup_{i\in I}\mathscr{M}_i$, $b_i = b_t^i + b_v^i$, $m_i = m_t^i + m_v^i$, and $d_i = d_t^i + d_v^i$, for each $i \in I$. By Lemma 47, there exists $\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_v = +_{i\in I}\Gamma_v^i$, $b_v = +_{i\in I}b_v^i$, $m_v = +_{i\in I}m_v^i$, and $d_v = +_{i\in I}d_v^i$. Therefore, we can build Φ_t as follows:

$$\frac{(\Phi_t^i)_{i\in I}}{+_{i\in I}\Gamma_t^i; x : \mathscr{M} \Vdash^{(+_{i\in I}b_t^i, +_{i\in I}m_t^i, +_{i\in I}d_t^i)} t : [\sigma_i]}\,(\mathtt{m})$$

And we can conclude $\Gamma_{t\{x\setminus v\}} = +_{i\in I}\Gamma_i = +_{i\in I}(\Gamma_t^i + \Gamma_v^i) = \Gamma_t + \Gamma_v$, $b = +_{i\in I}b_i = +_{i\in I}(b_t^i + b_v^i)$, $m = +_{i\in I}m_i = +_{i\in I}(m_t^i + m_v^i)$, and $d = +_{i\in I}d_i = +_{i\in I}(d_t^i + d_v^i)$.

- Rule $(\lambda_{\mathtt{p}})$. Then, $t = \lambda y.u$, $\tau = \mathtt{a}$, and $t\{x \setminus v\} = \lambda y.(u\{x \setminus v\})$. Moreover, $\Gamma_{t\{x\setminus v\}} = \emptyset$ and $b = m = d = 0$. Therefore, we can build $\Phi_t \rhd \emptyset \Vdash_{\mathtt{v}}^{(0,0,0)} \lambda y.u : \mathtt{a}$ using rule $(\lambda_{\mathtt{p}})$, such that $\Gamma_t = \emptyset$, $\mathscr{M} = [\,]$, and $b_t = m_t = d_t = 0$, and we can build $\Phi_v \rhd \Vdash_{\mathtt{v}}^{(0,0,0)} v : [\,]$, such that $\Gamma_v = \emptyset$ and $b_v = m_v = d_v = 0$, by using $(\mathtt{m})$ with no premises. Thus, we can conclude with $\Gamma_{t\{x\setminus v\}} = \emptyset = \Gamma_t + \Gamma_v$, and $b = 0 = b_t + b_v$, $m = 0 = m_t + m_v$, and $d = 0 = d_t + d_v$. □

Lemma 20 (Quantitative subject expansion). *Let* $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, *where* $\mathtt{r} \in \{\beta_{\mathtt{v}}, \mathtt{g}, \mathtt{s}\}$. *If* $\Phi' \rhd \Gamma \Vdash_{\mathtt{v}}^{(b',m',d)} (u,q) : \kappa$ *is a tight derivation, there exists* $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t,s) : \kappa$, *where* $\mathtt{r} = \beta_{\mathtt{v}}$ *implies* $b' = b - 1$ *and* $m' = m$, *while* $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ *implies* $b' = b$ *and* $m' = m - 1$.

Proof. We show a stronger statement of the form:

Let $(t,s) \rightsquigarrow_{\mathtt{r}} (u,q)$, where $\mathtt{r} \in \{\beta_{\mathtt{v}}, \mathtt{g}, \mathtt{s}\}$. If $\Phi' \rhd \Gamma \Vdash_{\mathtt{v}}^{(b',m',d)} (u,q) : \kappa$, Γ is tight, and (κ is tight or $\neg\mathtt{val}(t)$), then $\Phi \rhd \Gamma \Vdash_{\mathtt{v}}^{(b,m,d)} (t,s) : \kappa$, where $\mathtt{r} = \beta_{\mathtt{v}}$ implies $b' = b - 1$ and $m' = m$, while $\mathtt{r} \in \{\mathtt{g}, \mathtt{s}\}$ implies $b' = b$ and $m' = m - 1$.

We proceed by induction on $(t,s) \rightsquigarrow_{\mathtt{v}} (u,q)$:

- Case $(t,s) = ((\lambda x.p)v, s) \rightsquigarrow_{\beta_{\mathtt{v}}} (p\{x \setminus v\}, s) = (u,q)$. Then, Φ' has two premises $\Phi_{p\{x\setminus v\}} \rhd \Gamma_{p\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b_{p\{x\setminus v\}}, m_{p\{x\setminus v\}}, d_{p\{x\setminus v\}})} p\{x \setminus v\} : \mathscr{T} \Rightarrow \kappa$ and $\Pi_s \rhd \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}$, such that $\Gamma = \Gamma_{p\{x\setminus v\}} + \Delta_s$, $b' = b_{p\{x\setminus v\}} + b_s$, $m' = m_{p\{x\setminus v\}} + m_s$, and $d' = d_{p\{x\setminus v\}} + d_s$. By Lemma 51, there exist $\Phi_p \rhd \Gamma_p; x : \mathscr{M} \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T} \Rightarrow \kappa$ and $\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}$, such that $\Gamma_{p\{x\setminus v\}} = \Gamma_p + \Gamma_v$, $b_{p\{x\setminus v\}} = b_p + b_v$, $m_{p\{x\setminus v\}} = m_p + m_v$, and $d_{p\{x\setminus v\}} = d_p + d_v$. Therefore, we can build Φ as follows:

$$\dfrac{\dfrac{\dfrac{\Phi_p}{\Gamma_p \Vdash^{(b_p,m_p,d_p)} \lambda x.p : \mathscr{M} \to (\mathscr{T} \Rightarrow \kappa)}\,(\lambda)}{\Gamma_p \Vdash^{(b_p,m_p,d_p)} \lambda x.p : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T} \Rightarrow \kappa)) \times \mathscr{T})}\,(\uparrow) \qquad \dfrac{\Phi_v \rhd \Gamma_v \Vdash_{\mathtt{v}}^{(b_v,m_v,d_v)} v : \mathscr{M}}{\Gamma_v \Vdash^{(b_v,m_v,d_v)} v : \mathscr{T} \Rightarrow (\mathscr{M} \times \mathscr{T})}\,(\uparrow)}{\Gamma_p + \Gamma_v \Vdash^{(1+b_p+b_v, m_p+m_v, d_p+d_v)} (\lambda x.p)v : \mathscr{T} \Rightarrow \kappa}\,(@)$$

$$\dfrac{\Pi_s}{(\Gamma_p + \Gamma_v) + \Delta_s \Vdash^{(1+b_p+b_v+b_s, m_p+m_v+m_s, d_p+d_v+d_s)} ((\lambda x.t')v, s) : \kappa}\,(\mathtt{conf})$$

where $b = 1 + b_p + b_v + b_s$, $m = m_p + m_v + m_s$, and $d = d_p + d_v + d_s$. Thus, we can conclude with $\Gamma = \Gamma_{p\{x\setminus v\}} + \Delta_s = (\Gamma_p + \Gamma_v) + \Delta_s$, $b' = b_{p\{x\setminus v\}} + b_s = b_p + b_v + b_s = (1 + b_p + b_v + b_s) - 1 = b - 1$, $m' = m_{p\{x\setminus v\}} + m_s = (m_p + m_v) + m_s = m$, and $d' = d_{p\{x\setminus v\}} + d_s = (d_p + d_v) + d_s = d$.

- Case $(t,s) = (\mathtt{get}_l(\lambda x.p), s) \rightsquigarrow_{\mathtt{v}} (p\{x \setminus v\}, s) = (u,q)$, such that $q \equiv \mathrm{upd}_l(v,s') \equiv s$. Then, Φ' has two premises $\Phi_u \rhd \Gamma_{p\{x\setminus v\}} \Vdash_{\mathtt{v}}^{(b_{p\{x\setminus v\}}, m_{p\{x\setminus v\}}, d_{p\{x\setminus v\}})} p\{x \setminus v\} : \{l : \mathscr{M}_2\}; \mathscr{T} \Rightarrow \kappa$ and Π_q,

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

which must be of the following form:

$$\frac{\Phi_v^2 \rhd \Gamma_v^2 \Vdash_{\mathrm{v}}^{(b_v^2,m_v^2,d_v^2)} v : \mathscr{M}_2 \qquad \Pi_{s'} \rhd \Delta_{s'} \Vdash_{\mathrm{v}}^{(b_{s'},m_{s'},d_{s'})} s' : \{l : \mathscr{L}\}; \mathscr{T}}{\Gamma_v^2 + \Delta_{s'} \Vdash^{(b_v^2+b_{s'},m_v^2+m_{s'},d_v^2+d_{s'})} \mathtt{upd}_l(v,s') : \{l : \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T}}(\mathtt{upd})$$

where $\Gamma = \Gamma_{p\{x\setminus v\}} + (\Gamma_v^2 + \Delta_{s'})$, $b' = b_{p\{x\setminus v\}} + b_v^2 + b_{s'}$, $m' = m_{p\{x\setminus v\}} + b_v^2 + b_{s'}$, and $d' = d_{p\{x\setminus v\}} + d_v^2 + d_{s'}$. By Lemma 51, there exist $\Phi_p \rhd (b_p, m_p, d_p) \Vdash_{\mathrm{v}}^{\Gamma_p; x:\mathscr{M}_1} p : \{l : \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$ and $\Phi_v^1 \rhd \Gamma_v^1 \Vdash_{\mathrm{v}}^{(b_v^1,m_v^1,d_v^1)} v : \mathscr{M}_1$, such that $\Gamma_{p\{x\setminus v\}} = \Gamma_p + \Gamma_v^1$, $b_{p\{x\setminus v\}} = b_p + b_v^1$, $m_{p\{x\setminus v\}} = m_p + m_v^1$, and $d_{p\{x\setminus u\}} = d_p + d_v^1$. Therefore, we can build Φ_t as follows:

$$\frac{\Phi_p \rhd \Gamma_p; x : \mathscr{M}_1 \Vdash_{\mathrm{v}}^{(b_p,m_p,d_p)} p : \{l : \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}{\Gamma_p \Vdash^{(b_p,1+m_p,d_p)} \mathtt{get}_l(\lambda x.p) : \{l : \mathscr{M}_1 \sqcup \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{get})$$

By Lemma 47, there exists $\Phi_v \rhd \Gamma_v^1 + \Gamma_v^2 \Vdash_{\mathrm{v}}^{(b_v^1+b_v^2,m_v^1+m_v^2,d_v^1+d_v^2)} v : \mathscr{M}_1 \sqcup \mathscr{M}_2$. Therefore, we can build an alternative derivation Π'_q for q as follows:

$$\frac{\Phi_v \qquad \Pi_{s'}}{(\Gamma_v^1 + \Gamma_v^2) + \Delta_{s'} \Vdash^{(b_v^1+b_v^2+b_{s'},m_v^1+m_v^2+m_{s'},d_v^1+d_v^2+d_{s'})} \mathtt{upd}_l(v,s') : \{l : \mathscr{M}_1 \sqcup \mathscr{M}_2 \cdot \mathscr{L}\}; \mathscr{T}}(\mathtt{upd})$$

Then, we can pick $\Pi_s = \Pi'_q$ and build Φ as follows:

$$\frac{\Phi_t \qquad \Pi_s}{\Gamma_p + (\Gamma_v^1 + \Gamma_v^2) + \Delta_{s'} \Vdash^{(b_p+b_v^1+b_v^2+b_{s'},1+m_p+m_v^1+m_v^2+m_{s'},d_p+d_v^1+d_v^2+d_{s'})} (\mathtt{get}_l(\lambda x.p), s) : \kappa}(\mathtt{conf})$$

where $b = b_p + b_v^1 + b_v^2 + b_{s'}$, $m = 1 + m_p + m_v^1 + m_v^2 + m_{s'}$, and $d = d_p + d_v^1 + d_v^2 + d_{s'}$. Thus, we can conclude with $\Gamma = \Gamma_{p\{x\setminus v\}} + (\Gamma_v^2 + \Delta_{s'}) = \Gamma_p + \Gamma_v^1 + \Gamma_v^2 + \Delta_{s'}$, $b' = b_{p\{x\setminus v\}} + b_v^2 + b_{s'} = b_p + b_v^1 + b_v^2 + b_{s'} = b$, and $m' = m_{p\{x\setminus v\}} + m_v^2 + m_{s'} = m_p + m_v^1 + m_v^2 + m_{s'} = (1 + m_p + m_v^1 + m_v^2 + m_{s'}) - 1 = m - 1$, $d' = d_{p\{x\setminus v\}} + d_v^2 + d_{s'} = d_p + d_v^1 + d_v^2 + d_{s'} = d$.

- Case $(t,s) = (\mathtt{set}_l(v,p), s) \rightsquigarrow_{\mathrm{v}} (p, \mathtt{upd}_l(v,s)) = (u,q)$. Then, Φ' has two premises $\Phi_p \rhd \Gamma_p \Vdash_{\mathrm{v}}^{(b_p,m_p,d_p)} p : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa$ and Π_q, which must be of the following form:

$$\frac{\Phi_v \rhd \Gamma_v \Vdash_{\mathrm{v}}^{(b_v,m_v,d_v)} v : \mathscr{M} \qquad \Pi_s \rhd \Delta_s \Vdash_{\mathrm{v}}^{(b_s,m_s,d_s)} s : \{l : \mathscr{L}\}; \mathscr{T}}{\Gamma_v + \Delta_s \Vdash^{(b_v+b_s,m_v+m_s,d_v+d_s)} \mathtt{upd}_l(v,s) : \{l : \mathscr{M} \cdot \mathscr{L}\}; \mathscr{T}}(\mathtt{upd})$$

where $\Gamma = \Gamma_p + (\Gamma_v + \Delta_s)$, $b' = b_p + b_v + b_s$, $m' = m_p + m_v + m_s$, and $d' = d_p + d_v + d_s$. Therefore, we can build Φ as follows:

$$\frac{\dfrac{\Phi_v \qquad \Phi_p}{\Gamma_v + \Gamma_p \Vdash^{(b_v+b_p,1+m_v+m_p,d_v+d_p)} \mathtt{set}_l(v,p) : \{l : \mathscr{L}\}; \mathscr{T} \Rightarrow \kappa}(\mathtt{set}) \qquad \Pi_s}{(\Gamma_v + \Gamma_p) + \Delta_s \Vdash^{(b_v+b_p+b_s,1+m_v+m_p+m_s,d_v+d_p+d_s)} (\mathtt{set}_l(v,p), s) : \kappa}(\mathtt{conf})$$

where $b = b_v + b_p + b_s$, $m = 1 + m_v + m_p + m_s$, and $d = d_v + d_p + d_s$. Thus, we can conclude with $\Gamma = \Gamma_p + (\Gamma_v + \Delta_s)$, $b' = b_v + b_p + b_s = b$, $m' + 1 = (m_p + m_v + m_s) + 1 = m$, and $d' = d_v + d_p + d_s = d$.

- Case $(t,s) = (pr, s) \rightsquigarrow_{\mathrm{v}} (p'r, q) = (u,q)$, such that $(p,s) \rightsquigarrow_{\mathrm{v}} (p',q)$. Then, Φ' has two premises $\Phi_{p'r}$ and $\Pi_q \rhd \Delta_q \Vdash_{\mathrm{v}}^{(b_q,m_q,d_q)} q : \mathscr{T}$, and $\Phi_{p'r}$ must end with rule ($@$), ($@_{\mathrm{p}}^1$), or ($@_{\mathrm{p}}^2$):

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

– Case $\Phi_{p'r}$ ends with $(@)$. Then, $\Phi_{p'r}$ must be of the following form:

$$\frac{\Phi_{p'} \rhd \Gamma_{p'} \Vdash_{\mathtt{v}}^{(b_{p'},m_{p'},d_{p'})} p' : \mathscr{T} \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}') \quad \Phi_r \rhd \Gamma_r \Vdash_{\mathtt{v}}^{(b_r,m_r,d_r)} r : \mathscr{T}' \Rightarrow (\mathscr{M} \times \mathscr{T}'')}{\Gamma_{p'} + \Gamma_r \Vdash^{(1+b_{p'}+b_r,m_{p'}+m_r,d_{p'}+d_r)} p'r : \mathscr{T} \Rightarrow \kappa}\ (@)$$

where $\Gamma = (\Gamma_{p'} + \Gamma_r) + \Delta_q$, $b' = 1 + b_{p'} + b_r + b_q$, $m' = m_{p'} + m_r + m_q$, and $d' = d_{p'} + d_r + d_q$. Then, we can build $\Phi_{(p',q)}$ as follows:

$$\frac{\Phi_{p'} \qquad \Pi_s}{\Gamma_{p'} + \Delta_q \Vdash^{(b_{p'}+b_s,m_{p'}+m_s,d_{p'}+d_s)} (p', q) : ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}')}\ (\mathtt{conf})$$

Notice that $\neg \mathtt{val}(p)$, since $(p, s) \rightsquigarrow_{\mathtt{v}} (p', q)$. By applying the *i.h.* to $\Phi_{(p',q)}$, there exists $\Phi_{(p,s)}$ that must be of the following form:

$$\frac{\Phi_p \rhd \Gamma_p \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T}''' \Rightarrow ((\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}') \quad \Pi_s \rhd \Delta_s \Vdash_{\mathtt{v}}^{(b_s,m_s,d_s)} s : \mathscr{T}'''}{\Gamma_p + \Delta_s \Vdash^{(b_p+b_s,m_p+m_s,d_p+d_s)} (p, s) : \kappa}\ (\mathtt{conf})$$

where $\Gamma_{p'} + \Delta_q = \Gamma_p + \Delta_s$, and the counters $(b_p + b_s, m_p + m_s, d_p + d_s)$ and $(b_{p'} + b_q, m_{p'} + m_q, d_{p'} + d_q)$ are related properly. Therefore, we can build Φ as follows:

$$\frac{\dfrac{\Phi_p \qquad \Phi_r}{\Gamma_p + \Gamma_r \Vdash^{(1+b_p+b_r,m_p+m_r,d_p+d_r)} pr : \mathscr{T}''' \Rightarrow \kappa}\ (@) \qquad \Pi_s}{(\Gamma_p + \Gamma_r) + \Delta_s \Vdash^{(1+b_p+b_r+b_s,m_p+m_r+m_s,d_p+d_r+d_s)} (pr, s) : \kappa}\ (\mathtt{conf})$$

where $b = 1 + b_p + b_r + b_s$, $m = m_p + m_r + m_s$, and $d = d_p + d_r + d_s$. Thus, we can conclude with $\Gamma = (\Gamma_p + \Gamma_r) + \Delta_s = (\Gamma_{p'} + \Gamma_r) + \Delta_q$, and $(1 + b_p + b_r + b_s, m_p + m_r + m_s, d_p + d_r + d_s) = (b, m, d)$ and $(b', m', d') = (1 + b_{p'} + b_r + b_q, m_{p'} + m_r + m_q, d_{p'} + d_r + d_q)$ are properly related.

– Case $\Phi_{p'r}$ ends with rule $(@^1_{\mathtt{p}})$ and $(@^2_{\mathtt{p}})$. These cases are very similar to the previous case.

- Case $(t, s) = (pr, s) \rightsquigarrow_{\mathtt{v}} (pr', q) = (u, q)$, such that $(p, s) \not\rightsquigarrow_{\mathtt{v}}$, (p, s) is unblocked, and $(r, s) \rightsquigarrow_{\mathtt{v}} (r', q)$. Then, Φ' has two premises $\Phi_{pr'}$ and $\Pi_q \rhd \Delta_q \Vdash_{\mathtt{v}}^{(b_q,m_q,d_q)} q : \mathscr{T}$, and $\Phi_{pr'}$ must end with rule $(@)$, $(@^1_{\mathtt{p}})$, or $(@^2_{\mathtt{p}})$:

  – Case $\Phi_{pr'}$ ends with $(@)$. Then, $\Phi_{pr'}$ must be of the following form:

$$\frac{\Phi_p \rhd \Gamma_p \Vdash_{\mathtt{v}}^{(b_p,m_p,d_p)} p : \mathscr{T} \Rightarrow (\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}' \quad \Phi_{r'} \rhd \Gamma_{r'} \Vdash_{\mathtt{v}}^{(b_{r'},m_{r'},d_{r'})} r' : \mathscr{T}' \Rightarrow (\mathscr{M} \times \mathscr{T}'')}{\Gamma_p + \Gamma_{r'} \Vdash^{(1+b_p+b_{r'},m_p+m_{r'},d_p+d_{r'})} pr' : \mathscr{T} \Rightarrow \kappa}\ (@)$$

‘’ where $\Gamma = (\Gamma_p + \Gamma_{r'}) + \Delta_q$, $b' = 1 + b_p + b_{r'} + b_q$, $m' = m_p + m_{r'} + m_q$, and $d' = d_p + d_{r'} + d_q$. Notice that, since $(p, s) \not\rightsquigarrow_{\mathtt{v}}$ and (p, s) is unblocked (therefore (p, s) is an unblocked normal configuration), then $p \in \mathtt{no}_{\mathtt{v}}$, by definition and Proposition 5. Therefore, $\mathscr{T} = \mathscr{T}'$,

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press

according to Lemma 15. And, we can build $\Phi_{(r',q)}$ as follows:

$$\frac{\Phi_{r'} \qquad \Pi_q}{\Gamma_{r'} + \Delta_q \Vdash^{(b_{r'}+b_q, m_{r'}+m_q, d_{r'}+d_q)} (r', q) : (\mathscr{M} \times \mathscr{T}'')}(\texttt{conf})$$

Since Γ is tight, then $\Gamma_{r'}$ and Δ_q are both tight. Moreover, $\neg\texttt{val}(r)$, since $(r,s) \rightsquigarrow_{\mathtt{v}} (r',q)$. By applying the *i.h.* to $\Phi_{(r',q)}$, there exists $\Phi_{(r,s)}$, which must be of the following form:

$$\frac{\Phi_r \triangleright \Gamma_r \Vdash_{\mathtt{v}}^{(b_r, m_r, d_r)} r : \mathscr{T}''' \Rightarrow (\mathscr{M} \times \mathscr{T}'') \qquad \Pi_s \triangleright \Delta_s \Vdash_{\mathtt{v}}^{(b_s, m_s, d_s)} s : \mathscr{T}'''}{\Gamma_r + \Delta_s \Vdash^{(b_r+b_s, m_r+m_s, d_r+d_s)} (r,s) : (\mathscr{M} \times \mathscr{T}'')}(\texttt{conf})$$

where $\Gamma_{r'} + \Delta_q = \Gamma_r + \Delta_s$, and $(b_{r'} + b_q, m_{r'} + m_q, d_{r'} + d_q)$ and $(b_r + b_s, m_r + m_s, d_r + d_s)$ are related properly. Moreover, there exists $\Phi'_p \triangleright \Gamma_p \Vdash_{\mathtt{v}}^{(b_p, m_p, d_p)} p : \mathscr{T}''' \Rightarrow (\mathscr{M} \to (\mathscr{T}'' \Rightarrow \kappa)) \times \mathscr{T}'''$, by Lemma 15. Therefore, we can build Φ as follows:

$$\frac{\dfrac{\Phi'_p \qquad \Phi_r}{\Gamma_p + \Gamma_r \Vdash^{(1+b_p+b_r, m_p+m_r, d_p+d_r)} pr : (\mathscr{T}''' \times \kappa)}(@) \qquad \Pi_s}{\Gamma_p + \Gamma_r + \Delta_s \Vdash^{(1+b_p+b_r+b_s, m_p+m_r+m_s, d_p+d_r+d_s)} (pr, s) : \kappa}(\texttt{conf})$$

where $b = 1 + b_p + b_r + b_s$, $m = m_p + m_r + m_s$, and $d = d_p + d_r + d_s$. Thus, we can conclude with $\Gamma = (\Gamma_p + \Gamma_{r'}) + \Delta_q = (\Gamma_p + \Gamma_r) + \Delta_s$, and $(1 + b_p + b_{r'} + b_q, m_p + m_{r'} + m_q, d_p + d_{r'} + d_q) = (b', m', d')$ and $(b, m, d) = (1 + b_p + b_r + b_s, m_p + m_r + m_s, d_p + d_r + d_s)$ are properly related.

– Case $\Phi_{pr'}$ ends with $(@^1_{\mathtt{p}})$ or $(@^2_{\mathtt{p}})$. These two cases are very similar to the previous case. □

Theorem 6 (Quantitative completeness). *If* $(t,s) \rightsquigarrow_{\mathtt{v}}^{(b,m,d)} (u,q)$ *and* $u \in \mathtt{no}_{\mathtt{v}}$, *then there exists a tight derivation* $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b,m,|(u,q)|_{\mathtt{v}})} (t,s) : \kappa$.

Proof. By induction on $b + m$:

- Case $b + m = 0$. Then, $b = m = 0$, and thus, $(t,s) = (u,q)$. Therefore, $t \in \mathtt{no}_{\mathtt{v}}$. Thus, we can conclude by Lemma 18.
- Case $b + m > 0$. Then, there exists (t',s'), such that $(t,s) \rightsquigarrow_{\mathtt{v}}^{(1,0)} (t',s') \rightsquigarrow_{\mathtt{v}}^{(b-1,m)} (u,q)$ or $(t,s) \rightsquigarrow_{\mathtt{v}}^{(0,1)} (t',s') \rightsquigarrow_{\mathtt{v}}^{(b,m-1)} (u,q)$. By the *i.h.*, there exists $\Phi' \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b',m',|(u,q)|_{\mathtt{v}})} (t',s') : \kappa$ tight, such that $b' + m' = b + m - 1$. By Lemma 20, we have $\Phi \triangleright \Gamma \Vdash_{\mathtt{v}}^{(b'',m'',|(u,q)|_{\mathtt{v}})} (t,s) : \kappa$ tight, such that $b'' + m'' = 1 + b' + m'$. Therefore, $b'' + m'' = b + m$, since the fact that $b'' = b$, and $m'' = m$ can be easily checked by a simple case analysis. □

Cite this article: Alves S, Kesner D and Ramos M (2025). A quantitative approach to global state composition. *Mathematical Structures in Computer Science* **35**(e28), 1–96. https://doi.org/10.1017/S0960129525100285

https://doi.org/10.1017/S0960129525100285 Published online by Cambridge University Press