Types by Need (Extended Version)

Beniamino Accattoli[1], Giulio Guerrieri[2], and Maico Leberle[1]

[1] Inria & LIX, École Polytechnique, UMR 7161, Palaiseau, France
{beniamino.accattoli,maico.leberle}@inria.fr
[2] University di Bath, Departement of Computer Science, Bath, United Kingdom
g.guerrieri@bath.ac.uk

Abstract. A cornerstone of the theory of λ-calculus is that intersection types characterise termination properties. They are a flexible tool that can be adapted to various notions of termination, and that also induces adequate denotational models.
Since the seminal work of de Carvalho in 2007, it is known that multi types (*i.e.* non-idempotent intersection types) refine intersection types with quantitative information and a strong connection to linear logic. Typically, type derivations provide bounds for evaluation lengths, and minimal type derivations provide exact bounds.
De Carvalho studied call-by-name evaluation, and Kesner used his system to show the termination equivalence of call-by-need and call-by-name. De Carvalho's system, however, cannot provide exact bounds on call-by-need evaluation lengths.
In this paper we develop a new multi type system for call-by-need. Our system produces exact bounds and induces a denotational model of call-by-need, providing the first tight quantitative semantics of call-by-need.

1 Introduction

Duplications and erasures have always been considered as key phenomena in the λ-calculus—the λI-calculus, where erasures are forbidden, is an example of this. The advent of linear logic [35] gave them a new, prominent logical status. Forbidding erasure and duplication enables single-use resources, i.e. linearity, but limits expressivity, as every computation terminates in linear time. Their controlled reintroduction via the non-linear modality ! recovers the full expressive power of cut-elimination and allows a fine analysis of resource consumption. Duplication and erasure are therefore the key ingredients for logical expressivity, and—via Curry-Howard—for the expressivity of the λ-calculus. They are also essential to understand evaluation strategies.

In a λ-term there can be many β-redexes, that is, places where β-reduction can be applied. In this sense, the λ-calculus is non-deterministic. Non-determinism does not affect the result of evaluation, if any, but it affects whether evaluation terminates, and in how many steps. There are two natural deterministic evaluation strategies, *call-by-name* (shortened to CbN) and *call-by-value* (CbV), which have dual behaviour with respect to duplication and erasure.

Call-by-Name = Silly Duplication + Wise Erasure. CbN *never* evaluates arguments of β-redexes before the redexes themselves. As a consequence, it never evaluates in subterms that will be erased. This is wise, and makes CbN a *normalising strategy*, that is, a strategy that reaches a result whenever one exists[3]. A second consequence is that if the argument of the redex is duplicated then it may be evaluated more than once. This is silly, as it repeats work already done.

Call-by-Value = Wise Duplication + Silly Erasure. CbV, on the other hand, *always* evaluates arguments of β-redexes before the redexes themselves. Consequently, arguments are not re-evaluated—this is wise with respect to duplication—but they are also evaluated when they are going to be erased. For instance, on $t := (\lambda x.\lambda y.y)\Omega$, where Ω is the famous looping λ-term, CbV evaluation diverges (it keeps evaluating Ω) while CbN converges in one β-step (simply erasing Ω). This CbV treatment of erasure is clearly as silly as the duplicated work of CbN.

Call-by-Need = Wise Duplication + Wise Erasure. It is natural to try to combine the advantages of both CbN and CbV. The strategy that is wise with respect to both duplications and erasures is usually called *call-by-need* (CbNeed), it was introduced by Wadsworth [54], and dates back to the '70s. Despite being at the core of Haskell, one of the most-used functional programming languages, and—in its strong variant—being at work in the kernel of Coq as designed by Barras [14], the theory of CbNeed is much less developed than that of CbN or CbV.

One of the reasons for this is that it cannot be defined inside the λ-calculus without some hacking. Manageable presentations of CbNeed indeed require first-class sharing and micro-step operational semantics where variable occurrences are replaced one at a time (when needed), and not all at once as in the λ-calculus. Another reason is the less natural logical interpretation.

Linear Logic, Names, Values, and Needs. CbN and CbV have neat interpretations in linear logic. They correspond to two different representations of intuitionistic logic in linear logic, based on two different representations of implication[4].

The logical interpretation of CbNeed—studied by Maraist et al. in [44]—is less neat than those of CbN and CbV. Within linear logic, CbNeed is usually understood as corresponding to the CbV representation where erasures are generalised to all terms, not only those under the scope of a ! modality. So, it is seen as a sort of *affine* CbV. Such an interpretation however is unusual, because it does not match exactly with cut-elimination in linear logic, as for CbN and CbV.

Call-by-Need, Abstractly. The main theorem of the theory of CbNeed is that it is termination equivalent to CbN, that is, on a fixed term, CbNeed evaluation terminates if and only if CbN evaluation terminates, and, moreover, they essentially

[3] If a term t admits both converging and diverging evaluation sequences then the diverging sequences occur in erasable subterms of t, which is why CbN avoids them.

[4] The CbN translation maps $A \Rightarrow B$ to $(!A^{\mathrm{CbN}}) \multimap B^{\mathrm{CbN}}$, while the CbV maps it to $!A^{\mathrm{CbV}} \multimap !B^{\mathrm{CbV}}$, or equivalently to $!(A^{\mathrm{CbV}} \multimap B^{\mathrm{CbV}})$.

produce the same result (up to some technical details that are irrelevant here). This is due to the fact that both strategies avoid silly divergent sequences such as that of $(\lambda x.\lambda y.y)\Omega$. Termination equivalence is an abstract theorem stating that CbNeed erases as wisely as CbN. Curiously, in the literature there are no abstract theorems reflecting the dual fact that CbNeed duplicates as wisely as CbV—we provide one, as a side contribution of this paper.

Call-by-Need and Denotational Semantics. CbNeed is then usually considered as a CbV optimisation of CbN. In particular, every denotational model of CbN is also a model of CbNeed, and adequacy—that is the fact that the denotation of t is not degenerated if and only if t terminates—transfers from CbN to CbNeed.

Denotational semantics is invariant by evaluation, and so is insensitive to evaluation lengths by definition. It then seems that denotational semantics cannot distinguish between CbN and CbNeed. The aim of this paper is, somewhat counter-intuitively, to separate CbN and CbNeed semantically. We develop a type system whose type judgements induce a model—this is typical of *intersection* type systems—and whose type derivations provide exact bounds for CbNeed evaluation—this is usually obtained via *non-idempotent* intersection types. Unsurprisingly, the design of the type system requires a delicate mix of erasure and duplication and builds on the linear logic understanding of CbN and CbV.

Multi Types. Our typing framework is given by *multi types*, which is an alternative name for *non-idempotent intersection types*[5]. Multi types characterise termination properties exactly as intersection types, having moreover the advantages that they are closely related to (the relational semantics of) linear logic, their type derivations provide quantitative information about evaluation lengths, and the proof techniques are simpler—no need for the reducibility method.

The seminal work of de Carvalho [20] (appeared in 2007 but unpublished until 2018) showed how to use multi types to obtain exact bounds on evaluation lengths in CbN. Ehrhard adapted multi types to CbV [29], and very recently Accattoli and Guerrieri adapted de Carvalho's study of exact bounds to Ehrhard's system and CbV evaluation [8]. Kesner used de Carvalho's CbN multi types to obtain a simple proof that CbNeed is termination equivalent with respect to CbN [37] (first proved with other techniques by Maraist, Odersky, and Wadler [45] and Ariola and Felleisen [10] in the nineties), and then Kesner and coauthors continued exploring the theory of CbNeed via CbN multi types [12,39,13].

Kesner's use of CbN multi types to study CbNeed is *qualitative*, as it deals with termination and not with exact bounds. For a *quantitative* study of CbNeed, de Carvalho's CbN system cannot really be informative: CbN multi types provide bounds for CbNeed which cannot be exact because they already provide exact bounds for CbN, which generally takes more steps than CbNeed.

Multi Types by Need. In this paper we provide the first multi type system characterising CbNeed termination and whose minimal type derivations provide

[5] The new terminology is due to the fact that a non-idempotent intersection $A \wedge A \wedge B \wedge C$ can be seen as a multi-set $[A, A, B, C]$.

exact bounds for CbNeed evaluation lengths. The design of the type system is delicate, as we explain in Sect. 6. One of the key points is that, in contrast to Ehrhard's system for CbV [29], multi types for CbNeed cannot be directly extracted by the relational semantics of linear logic, given that CbNeed does not have a clean representation in it. A by-product of our work is a new denotational semantics of CbNeed, the first one to precisely reflect its quantitative properties.

Beyond the result itself, the paper tries to stress how the key ingredients of our type system are taken from those for CbN and CbV and combined together. To this aim, we first present multi types for CbN and CbV, and only then we proceed to build the CbNeed system and prove its properties.

Along the way, we also prove the missing fundamental property of CbNeed, that is, that it duplicates as efficiently as CbV. The result is obtained by dualising Kesner's approach [37], showing that the CbV multi type system is correct also with respect to CbNeed evaluation, that is, its bounds are also valid with respect to CbNeed evaluation lengths. Careful: the CbV system is correct but of course not complete with respect to CbNeed, because CbNeed may normalise when CbV diverges. The proof of the result is straightforward, because of our presentations of (CbN,) CbV and CbNeed. We adopt a liberal, non-deterministic formulation of CbV, and assuming that garbage collection is always postponed. These two ingredients turn CbNeed into a fragment of CbV, obtaining the new fundamental result as a corollary of correctness of CbV multi types for CbV evaluation.

Technical Development. The paper is extremely uniform, technically speaking. The three evaluations are presented as strategies of Accattoli and Kesner's Linear Substitution Calculus (shortened to LSC) [1,6], a calculus with a simple but expressive form of explicit sharing. The LSC is strongly related to linear logic [2], and provides a neat and manageable presentation of CbNeed, introduced by Accattoli, Barenbaum, and Mazza in [3], and further developed by various authors in [9,37,12,4,5,39,13]. Our type systems count evaluation steps by annotating typing rules in the *exact* same way, and the proofs of correctness and completeness all follow the *exact* same structure. While the results for CbN are very minor variations with respect to those in the literature [20,7], those for CbV are the first ones with respect to a presentation of CbV with sharing.

As it is standard for CbNeed, we restrict our study to closed terms and weak evaluation (that is, out of abstractions). The main consequence of this fact is that normal forms are particularly simple (sometimes called *answers* in the literature). Compared with other recent works dealing with exact bounds such as Accattoli, Graham-Lengrand, and Kesner's [7] and Accattoli and Guerrieri's [8] the main difference is that the size of normal forms is not taken into account by type derivations. This is because of the simple notions of normal forms in the closed and weak case, and not because the type systems are not accurate.

Related work about CbNeed. Call-by-need was introduced by Wadsworth [54] in the '70s. In the '90s, it was first reformulated as operational semantics by Launchbury [43], Maraist, Odersky, and Wadler [45], and Ariola and Felleisen [10], and then implemented by Sestoft [52] and further studied by Kutzner and

Schmidt-Schauß [42]. More recent papers are Garcia, Lumsdaine, and Sabry's [31], Ariola, Herbelin, and Saurin's [11], Chang and Felleisen's [23], Danvy and Zerny's [26], Downen et al.'s [28], Pédrot and Saurin's [50], and Balabonski et al.'s [12].

Related work about Multi Types. Intersection types are a standard tool to study λ-calculi—see Coppo and Dezani [24,25], Pottinger [51], and Krivine [41]. Non-idempotent intersection types, *i.e.* multi types, were first considered by Gardner [32], and then by Kfoury [40], Neergaard and Mairson [47], and de Carvalho [20]—a survey is Bucciarelli, Kesner, and Ventura's [18].

Many recent works rely on multi types or relational semantics to study properties of programs and proofs. Beyond the cited ones, Diaz-Caro, Manzonetto, and Pagani's [27], Carraro and Guerrieri's [19], Ehrhard and Guerrieri's [30], and Guerrieri's [36] deal with CbV, while Bernadet and Lengrand's [15], de Carvalho, Pagani, and Tortora de Falco's [22] provide exact bounds. Further related work is by Bucciarelli, Ehrhard, and Manzonetto [16], de Carvalho and Tortora de Falco [21], Tsukada and Ong [53], Kesner and Vial [38], Piccolo, Paolini and Ronchi Della Rocca [49], Ong [48], Mazza, Pellissier, and Vial [46], Bucciarelli, Kesner and Ronchi Della Rocca [17]—this list is not exhaustive.

This is the long version (with all proofs) of a paper accepted to ESOP 2019.

2 Closed λ-Calculi

In this section we define the CbN, CbV, and CbNeed evaluation strategies. We present them in the context of the Accattoli and Kesner's *linear substitution calculus* (LSC) [1,6]. We mainly follow the uniform presentation of these strategies given by Accattoli, Barenbaum, and Mazza [3]. The only difference is that we adopt a non-deterministic presentation of CbV, subsuming both the left-to-right and the right-to-left strategies in [3], that makes our results slightly more general. Such a non-determinism is harmless: not only CbV evaluation is confluent, it even has the diamond property, so that all evaluations have the same length.

Terms and Contexts. The set of terms $\Lambda_{\mathtt{lsc}}$ of the LSC is given by the following grammar, where $t[x{\leftarrow}s]$ is called an *explicit substitution* (shortened to ES), that is a more compact notation for $\mathtt{let}\ x = s\ \mathtt{in}\ t$:

$$\text{LSC TERMS}\quad t, s ::= x \mid v \mid ts \mid t[x{\leftarrow}s] \qquad\qquad \text{LSC VALUES}\quad v ::= \lambda x.t$$

The set $\mathtt{fv}(t)$ of *free* variables of a term t is defined as expected, in particular, $\mathtt{fv}(t[x{\leftarrow}s]) := (\mathtt{fv}(t) \setminus \{x\}) \cup \mathtt{fv}(s)$. A term t is *closed* if $\mathtt{fv}(t) = \emptyset$, *open* otherwise. As usual, terms are identified up to α-equivalence.

Contexts are terms with exactly one occurrence of the *hole* $\langle\cdot\rangle$, an additional constant. We shall use many different contexts. The most general ones are *weak contexts* W (i.e. not under abstractions). The (evaluation) contexts C, V and E—used to define CbN, CbV and CbNeed evaluation strategies, respectively—are special cases of weak contexts (in fact, CbV contexts coincide with weak contexts, the consequences of that are discussed on p. 8). To define evaluation strategies, *substitution contexts* (*i.e.* lists of explicit substitutions) also play a role.

$$\begin{array}{rl}
\textsc{Weak contexts} & W ::= \langle\cdot\rangle \mid Wt \mid W[x{\leftarrow}t] \mid tW \mid t[x{\leftarrow}W] \\
\textsc{Substitution contexts} & S ::= \langle\cdot\rangle \mid S[x{\leftarrow}t] \\
\textsc{CbN contexts} & C ::= \langle\cdot\rangle \mid Ct \mid C[x{\leftarrow}t] \\
\textsc{CbV contexts} & V ::= W \\
\textsc{CbNeed contexts} & E ::= \langle\cdot\rangle \mid Et \mid E[x{\leftarrow}t] \mid E\langle\!\langle x\rangle\!\rangle[x{\leftarrow}E']
\end{array}$$

We write $W\langle t\rangle$ for the term obtained by replacing the hole $\langle\cdot\rangle$ in context W by the term t. This *plugging* operation, as usual with contexts, can capture variables—for instance $(\langle\cdot\rangle t)[x{\leftarrow}s])\langle x\rangle = (xt)[x{\leftarrow}s]$. We write $W\langle\!\langle t\rangle\!\rangle$ when we want to stress that the context W does not capture the free variables of t.

Micro-step semantics. The rewriting rules decompose the usual small-step semantics for λ-calculi, by substituting one variable occurrence at the time, and only when such an occurrence is in evaluation position. We emphasise this fact saying that we adopt a *micro-step semantics*. We now give the definitions, examples of evaluation sequences follow right next.

Formally, a micro-step semantics is defined by first giving its *root-steps* and then taking the closure of root-steps under suitable contexts.

$$\begin{array}{rc}
\textsc{Multiplicative root-step} & S\langle\lambda x.t\rangle s \mapsto_{\mathtt{m}} S\langle t[x{\leftarrow}s]\rangle \\
\textsc{Exponential CbN root-step} & C\langle\!\langle x\rangle\!\rangle[x{\leftarrow}t] \mapsto_{\mathtt{e}_{\text{cbn}}} C\langle\!\langle t\rangle\!\rangle[x{\leftarrow}t] \\
\textsc{Exponential CbV root-step} & V\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S\langle v\rangle] \mapsto_{\mathtt{e}_{\text{cbv}}} S\langle V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle \\
\textsc{Exponential CbNeed root-step} & E\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S\langle v\rangle] \mapsto_{\mathtt{e}_{\text{need}}} S\langle E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle
\end{array}$$

where, in the root-step $\mapsto_{\mathtt{m}}$ (resp. $\mapsto_{\mathtt{e}_{\text{cbv}}}$; $\mapsto_{\mathtt{e}_{\text{need}}}$), if $S := [y_1{\leftarrow}s_1]\dots[y_n{\leftarrow}s_n]$ for some $n \in \mathbb{N}$, then $\mathtt{fv}(s)$ (resp. $\mathtt{fv}(V\langle\!\langle x\rangle\!\rangle)$; $\mathtt{fv}(E\langle\!\langle x\rangle\!\rangle)$) and $\{y_1,\dots,y_n\}$ are disjoint. This condition can always be fulfilled by α-equivalence.

The *evaluation strategies* $\to_{\text{cbn}}$ for CbN, $\to_{\text{cbv}}$ for CbV, and $\to_{\text{need}}$ for CbNeed, are defined as the closure of root-steps under CbN, CbV and CbNeed evaluation contexts, respectively (so, all evaluation strategies do not reduce under abstractions, since all such contexts are weak):

$$\begin{array}{c|c|c}
\text{CbN} & \text{CbV} & \text{CbNeed} \\
\to_{\mathtt{m}_{\text{cbn}}} := C\langle\mapsto_{\mathtt{m}}\rangle & \to_{\mathtt{m}_{\text{cbv}}} := V\langle\mapsto_{\mathtt{m}}\rangle & \to_{\mathtt{m}_{\text{need}}} := E\langle\mapsto_{\mathtt{m}}\rangle \\
\to_{\mathtt{e}_{\text{cbn}}} := C\langle\mapsto_{\mathtt{e}_{\text{cbn}}}\rangle & \to_{\mathtt{e}_{\text{cbv}}} := V\langle\mapsto_{\mathtt{e}_{\text{cbv}}}\rangle & \to_{\mathtt{e}_{\text{need}}} := E\langle\mapsto_{\mathtt{e}_{\text{need}}}\rangle \\
\to_{\text{cbn}} := C\langle\mapsto_{\mathtt{m}}\cup\mapsto_{\mathtt{e}_{\text{cbn}}}\rangle & \to_{\text{cbv}} := V\langle\mapsto_{\mathtt{m}}\cup\mapsto_{\mathtt{e}_{\text{cbv}}}\rangle & \to_{\text{need}} := E\langle\mapsto_{\mathtt{m}}\cup\mapsto_{\mathtt{e}_{\text{need}}}\rangle
\end{array}$$

where the notation $\to\, := W\langle\mapsto\rangle$ means that, given a root-step $\mapsto$, the evaluation $\to$ is defined as follows: $t \to s$ if and only if there are terms t' and s' and a context W such that $t = W\langle t'\rangle$ and $s = W\langle s'\rangle$ and $t' \mapsto s'$.

Note that evaluations $\to_{\text{cbn}}$, $\to_{\text{cbv}}$ and $\to_{\text{need}}$ can equivalently be defined as $\to_{\mathtt{m}_{\text{cbn}}} \cup \to_{\mathtt{e}_{\text{cbn}}}$, $\to_{\mathtt{m}_{\text{cbn}}} \cup \to_{\mathtt{e}_{\text{cbv}}}$ and $\to_{\mathtt{m}_{\text{need}}} \cup \to_{\mathtt{e}_{\text{need}}}$, respectively.

Given an evaluation sequence $d\colon t \to^*_{\mathrm{cbn}} s$ we note with $|d|$ the length of d, and with $|d|_{\mathtt{m}}$ and $|d|_{\mathtt{e}}$ the number of multiplicative and exponential steps in d, respectively—and similarly for $\to_{\mathrm{cbv}}$ and $\to_{\mathrm{need}}$.

Erasing Steps. The reader may be surprised by our evaluation strategies, as none of them includes erasing steps, despite the absolute relevance of erasures pointed out in the introduction. There are no contradictions: in the LSC—in contrast to the λ-calculus—erasing steps can always be postponed, and so they are often simply omitted. This is actually close to programming language practice, as the garbage collector acts asynchronously with respect to the evaluation flow. For the sake of clarity let us spell out the erasing rules—they shall nonetheless be ignored in the rest of the paper. In CbN and CbNeed every term is erasable, so the root erasing step takes the following form

$$t[x{\leftarrow}s] \mapsto_{\mathtt{gc}} t \qquad \text{if } x \notin \mathtt{fv}(t)$$

and it is then closed by weak evaluation contexts.

In CbV only values are erasable; so, the root erasing step in CbV is:

$$t[x{\leftarrow}S\langle v\rangle] \mapsto_{\mathtt{gc}} S\langle t\rangle \qquad \text{if } x \notin \mathtt{fv}(t)$$

and it is then closed by weak evaluation contexts.

Example 1. A good example to observe the differences between CbN, CbV, and CbNeed is given by the term $t \coloneqq ((\lambda x.\lambda y.xx)(II))(II)$ where $I \coloneqq \lambda z.z$ is the identity combinator. In CbN, it evaluates with 5 multiplicative steps and 5 exponential steps, as follows:

$$\begin{array}{ll}
t \to_{\mathtt{m}_{\mathrm{cbn}}} (\lambda y.xx)[x{\leftarrow}II](II) & \to_{\mathtt{m}_{\mathrm{cbn}}} (xx)[y{\leftarrow}II][x{\leftarrow}II] \\
\quad \to_{\mathtt{e}_{\mathrm{cbn}}} ((II)x)[y{\leftarrow}II][x{\leftarrow}II] & \to_{\mathtt{m}_{\mathrm{cbn}}} (z[z{\leftarrow}I]x)[y{\leftarrow}II][x{\leftarrow}II] \\
\quad \to_{\mathtt{e}_{\mathrm{cbn}}} (I[z{\leftarrow}I]x)[y{\leftarrow}II][x{\leftarrow}II] & \to_{\mathtt{m}_{\mathrm{cbn}}} w[w{\leftarrow}x][z{\leftarrow}I][y{\leftarrow}II][x{\leftarrow}II] \\
\quad \to_{\mathtt{e}_{\mathrm{cbn}}} x[w{\leftarrow}x][z{\leftarrow}I][y{\leftarrow}II][x{\leftarrow}II] & \to_{\mathtt{e}_{\mathrm{cbn}}} (II)[w{\leftarrow}x][z{\leftarrow}I][y{\leftarrow}II][x{\leftarrow}II] \\
\quad \to_{\mathtt{m}_{\mathrm{cbn}}} x'[x'{\leftarrow}I][w{\leftarrow}x][z{\leftarrow}I][y{\leftarrow}II][x{\leftarrow}II] & \to_{\mathtt{e}_{\mathrm{cbn}}} I[x'{\leftarrow}I][w{\leftarrow}x][z{\leftarrow}I][y{\leftarrow}II][x{\leftarrow}II]
\end{array}$$

In CbV, t evaluates with 5 multiplicative steps and 5 exponential steps, for instance from right to left, as follows:

$$\begin{array}{ll}
t \to_{\mathtt{m}_{\mathrm{cbv}}} (\lambda x.\lambda y.xx)(II)(z[z{\leftarrow}I]) & \to_{\mathtt{e}_{\mathrm{cbv}}} (\lambda x.\lambda y.xx)(II)(I[z{\leftarrow}I]) \\
\quad \to_{\mathtt{m}_{\mathrm{cbv}}} (\lambda x.\lambda y.xx)(w[w{\leftarrow}I])(I[z{\leftarrow}I]) & \to_{\mathtt{e}_{\mathrm{cbv}}} (\lambda x.\lambda y.xx)(I[w{\leftarrow}I])(I[z{\leftarrow}I]) \\
\quad \to_{\mathtt{m}_{\mathrm{cbv}}} (\lambda y.xx)[x{\leftarrow}I[w{\leftarrow}I]](I[z{\leftarrow}I]) & \to_{\mathtt{m}_{\mathrm{cbv}}} (xx)[y{\leftarrow}I[z{\leftarrow}I]][x{\leftarrow}I[w{\leftarrow}I]] \\
\quad \to_{\mathtt{e}_{\mathrm{cbv}}} (xI)[y{\leftarrow}I[z{\leftarrow}I]][x{\leftarrow}I][w{\leftarrow}I] & \to_{\mathtt{e}_{\mathrm{cbv}}} (II)[y{\leftarrow}I[z{\leftarrow}I]][x{\leftarrow}I][w{\leftarrow}I] \\
\quad \to_{\mathtt{m}_{\mathrm{cbv}}} x'[x'{\leftarrow}I][y{\leftarrow}I[z{\leftarrow}I]][x{\leftarrow}I][w{\leftarrow}I] & \to_{\mathtt{e}_{\mathrm{cbv}}} I[x'{\leftarrow}I][y{\leftarrow}I[z{\leftarrow}I]][x{\leftarrow}I][w{\leftarrow}I]
\end{array}$$

Note that the fact that CbN and CbV take the same number of steps is by chance, as they reduce different redexes: CbN never reduce the unneeded redex II associated to y, but it reduces twice the needed II redex associated to x, while CbV reduces both, but each one only once.

In CbNeed, t evaluates in 4 multiplicative steps and 4 exponential steps.

$$\begin{array}{ll} t \to_{\mathtt{m}_{\text{need}}} (\lambda y.xx)[x{\leftarrow}II](II) & \to_{\mathtt{m}_{\text{need}}} (xx)[y{\leftarrow}II][x{\leftarrow}II] \\ \quad \to_{\mathtt{m}_{\text{need}}} (xx)[y{\leftarrow}II][x{\leftarrow}z[z{\leftarrow}I]] & \to_{\mathsf{e}_{\text{need}}} (xx)[y{\leftarrow}II][x{\leftarrow}I[z{\leftarrow}I]] \\ \quad \to_{\mathsf{e}_{\text{need}}} (Ix)[y{\leftarrow}II][x{\leftarrow}I][z{\leftarrow}I] & \to_{\mathtt{m}_{\text{need}}} (w[w{\leftarrow}x])[y{\leftarrow}II][x{\leftarrow}I][z{\leftarrow}I] \\ \quad \to_{\mathsf{e}_{\text{need}}} w[w{\leftarrow}I][y{\leftarrow}II][x{\leftarrow}I][z{\leftarrow}I] & \to_{\mathsf{e}_{\text{need}}} I[w{\leftarrow}I][y{\leftarrow}II][x{\leftarrow}I][z{\leftarrow}I] \end{array}$$

CbV Diamond Property. CbV contexts coincide with weak ones. As a consequence, our presentation of CbV is non-deterministic, as for instance one can have

$$x[x{\leftarrow}I](y[y{\leftarrow}I]) \;{}_{\mathtt{m}_{\text{cbv}}}\!\!\leftarrow (II)(y[y{\leftarrow}I]) \to_{\mathsf{e}_{\text{cbv}}} (II)(I[y{\leftarrow}I])$$

but it is easily seen that diagrams can be closed in exactly one step (if the two reducts are different). For instance,

$$x[x{\leftarrow}I](y[y{\leftarrow}I]) \to_{\mathsf{e}_{\text{cbv}}} x[x{\leftarrow}I](I[y{\leftarrow}I]) \;{}_{\mathtt{m}_{\text{cbv}}}\!\!\leftarrow (II)(I[y{\leftarrow}I])$$

Moreover, the kind of steps is preserved, as the example illustrates. This is an instance of the strong form of confluence called *diamond property.* A consequence is that either all evaluation sequences normalise or all diverge, and if they normalise they have all the same length and the same number of steps of each kind. Roughly, the diamond property is a form of relaxed determinism. In particular, it makes sense to talk about the number of multiplicative / exponential steps to normal form, independently of the evaluation sequence. The proof of the property is an omitted routine check of diagrams.

Normal Forms. We use two predicates to characterise normal forms, one for both CbN and CbNeed normal forms, for which ES can contain whatever term, and one for CbV normal forms, where ES can only contain normal terms:

$$\frac{}{\mathsf{normal}(\lambda x.t)} \qquad \frac{\mathsf{normal}(t)}{\mathsf{normal}(t[x{\leftarrow}s])} \qquad \frac{}{\mathsf{normal}_{\text{cbv}}(\lambda x.t)} \qquad \frac{\mathsf{normal}_{\text{cbv}}(t) \quad \mathsf{normal}_{\text{cbv}}(s)}{\mathsf{normal}_{\text{cbv}}(t[x{\leftarrow}s])}$$

Proposition 1 (Syntactic characterisation of closed normal forms). *Let t be a closed term.*
1. CbN and CbNeed*: For* $\mathsf{r} \in \{\text{cbn}, \text{need}\}$*,* t *is* r*-normal if and only if* $\mathsf{normal}(t)$*.*
2. CbV*:* t *is* cbv*-normal if and only if* $\mathsf{normal}_{\text{cbv}}(t)$*.*

The simple structure of normal forms is the main point where the restriction to closed calculi plays a role in this paper.

From the syntactic characterization of normal forms (Proposition 1) it follows immediately that among closed terms, CbN and CbNeed normal forms coincide, while CbV normal forms are a subset of them. Such a subset is proper since the closed term $I[x{\leftarrow}\delta\delta]$ (where $I \coloneqq \lambda z.z$ and $\delta \coloneqq \lambda y.yy$) is CbN normal but not CbV normal (and it cannot normalise in CbV).

3 Preliminaries About Multi Types

In this section we define basic notions about multi types, type contexts, and (type) judgements that are shared by the three typing systems of the paper.

Multi-Sets. The type systems are based on two layers of types, defined in a mutually recursive way, *linear types* L and finite *multi-sets* M of linear types. The intuition is that a linear type L corresponds to a single use of a term, and that an argument t is typed with a multi-set M of n linear types if it is going to end up (at most) n times in evaluation position, with respect to the strategy associated with the type system. The three systems differ on the definition of linear types, that is therefore not specified here, while all adopt the same notion of finite multi-set M of linear types (named *multi type*), that we now introduce:

$$\text{MULTI TYPES} \qquad M, N ::= [L_i]_{i \in J} \quad (J \text{ a finite set})$$

where $[\ldots]$ denotes the multi-set constructor. The empty multi-set $[\,]$ (the multi type obtained for $J = \emptyset$) is called *empty (multi) type* and denoted by the special symbol $\mathbf{0}$. An example of multi-set is $[L, L, L']$, that contains two occurrences of L and one occurrence of L'. Multi-set union is noted $\uplus$.

Type Contexts. A *type context* Γ is a map from variables to multi types such that only finitely many variables are not mapped to $\mathbf{0}$. The *domain* of Γ is the set $\mathtt{dom}(\Gamma) := \{x \mid \Gamma(x) \neq \mathbf{0}\}$. The type context Γ is *empty* if $\mathtt{dom}(\Gamma) = \emptyset$.

Multi-set union $\uplus$ is extended to type contexts point-wise, *i.e.* $\Gamma \uplus \Pi$ maps each variable x to $\Gamma(x) \uplus \Pi(x)$. This notion is extended to several contexts as expected, so that $\biguplus_{i \in J} \Gamma_i$ denotes a finite union of contexts—when $J = \emptyset$ the notation is to be understood as the empty context. We write $\Gamma; x : M$ for $\Gamma \uplus (x \mapsto M)$ only if $x \notin \mathtt{dom}(\Gamma)$. More generally, we write $\Gamma; \Pi$ if the intersection between the domains of Γ and Π is empty.

The *restricted* context Γ with respect to the variable x, written $\Gamma \setminus\!\!\setminus x$ is defined by $(\Gamma \setminus\!\!\setminus x)(x) := \mathbf{0}$ and $(\Gamma \setminus\!\!\setminus x)(y) := \Gamma(y)$ if $y \neq x$.

Judgements. Type judgements are of the form $\Gamma \vdash^{(m,e)} t : L$ or $\Gamma \vdash^{(m,e)} t : M$, where the *indices* m and e are natural numbers whose intended meaning is that t evaluates to normal form in m multiplicative steps and e exponential steps, with respect to the evaluation strategy associated with the type system.

To make clear in which type systems the judgement is derived, we write $\Phi \triangleright_{\mathrm{cbn}} \Gamma \vdash^{(m,e)} t : L$ if Φ is a derivation in the CbN system ending in the judgement $\Gamma \vdash^{(m,e)} t : L$, and similarly for CbV and CbNeed.

4 Types by Name

In this section we introduce the CbN multi type system, together with intuitions about multi types. We also prove that derivations provide exact bounds on CbN evaluation sequences, and define the induced denotational model.

CbN Types. The system is essentially a reformulation of de Carvalho's system $\mathtt{R}$ [20], itself being a type-based presentation of the relational model of the CbN λ-calculus induced by relational model of linear logic via the CbN translation of λ-calculus into linear logic. Definitions:

$$\frac{}{x\!:\![L] \vdash^{(0,1)} x\!:\!L}\ \mathsf{ax} \qquad\qquad \frac{}{\vdash^{(0,0)} \lambda x.t\!:\!\mathsf{normal}}\ \mathsf{normal}$$

$$\frac{\Gamma \vdash^{(m,e)} t\!:\!L}{\Gamma \setminus\!\!\setminus x \vdash^{(m,e)} \lambda x.t\!:\!\Gamma(x) \multimap L}\ \mathtt{fun} \qquad\qquad \frac{(\Pi_i \vdash^{(m_i,e_i)} t\!:\!L_i)_{i\in J}}{\biguplus_{i\in J} \Pi_i \vdash^{(\sum_{i\in J} m_i, \sum_{i\in J} e_i)} t\!:\![L_i]_{i\in J}}\ \mathsf{many}$$

$$\frac{\Gamma \vdash^{(m,e)} t\!:\!M \multimap L \quad \Pi \vdash^{(m',e')} s\!:\!M}{\Gamma \uplus \Pi \vdash^{(m+m'+1,e+e')} ts\!:\!L}\ \mathsf{app} \qquad \frac{\Gamma, x\!:\!M \vdash^{(m,e)} t\!:\!L \quad \Pi \vdash^{(m',e')} s\!:\!M}{\Gamma \uplus \Pi \vdash^{(m+m',e+e')} t[x\leftarrow s]\!:\!L}\ \mathsf{ES}$$

Fig. 1. Type system for CbN evaluation

– CbN *linear types* are given by the following grammar:

$$\text{CbN linear types} \qquad\qquad L, L' ::= \mathsf{normal} \mid M \multimap L$$

Multi(-sets) types are defined as in Sect. 3, relatively to CbN linear types. Note the linear constant normal (used to type abstractions, which are normal terms): it plays a crucial role in our quantitative analysis of CbN evaluation.

– The CbN *typing rules* are in Fig. 1.

– The many *rule*: it has as many premises as the elements in the (possibly empty) set of indices J. When $J = \emptyset$, the rule has no premises, and it types t with the empty multi type $\mathbf{0}$. The many rule is needed to derive the right premises of the rules app and ES, that have a multi type M on their right-hand side. Essentially, it corresponds to the promotion rule of linear logic, that, in the CbN representation of the λ-calculus, is indeed used for typing the right subterm of applications and the content of explicit substitutions.

– The *size* of a derivation $\Phi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e)} t\!:\!L$ is the sum $m + e$ of the indices. A quick look to the typing rules shows that indices on typing judgements are not needed, as m can be recovered as the number of app rules, and e as the number of ax rules. It is however handy to note them explicitly.

Subtleties and easy facts. Let us overview some facts about our presentation of the type system.

1. *Introduction and destruction of multi-sets*: multi-set are introduced on the right by the many rule and on the left by ax. Moreover, on the left they are summed by app and ES.
2. *Vacuous abstractions*: we rely on the convention that the abstraction rule $\mathtt{fun}$ can always abstract a variable x not explicitly occurring in the context. Indeed, if $x \notin \mathtt{dom}(\Gamma)$, then $\Gamma \setminus\!\!\setminus x$ is equal to Γ since $\Gamma(x) = \mathbf{0}$.
3. *Relevance*: No weakening is allowed in axioms. An easy induction on type derivations shows that

Lemma 1 (Type contexts and variable occurrences for CbN). *Let $\Phi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e)} t\!:\!L$ be a derivation. If $x \notin \mathtt{fv}(t)$ then $x \notin \mathtt{dom}(\Gamma)$.*

Lemma 1 implies that derivations of closed terms have empty type context. Note that there can be free variables of t not in $\mathtt{dom}(\Gamma)$: the ones only occurring in subterms not touched by the evaluation strategy.

Key Ingredients. Two key points of the CbN system that play a role in the design of the CbNeed one in Sect. 6 are:

1. *Erasable terms and* $\mathbf{0}$: the empty multi type $\mathbf{0}$ is the type of erasable terms. Indeed, abstractions that erase their argument—whose paradigmatic example is $\lambda x.y$—can only be typed with $\mathbf{0} \multimap L$, because of *Lemma* 1. Note that in CbN every term—even diverging ones—can be typed with $\mathbf{0}$ by rule many (taking 0 premises), because, correctly, in CbN every term can be erased.
2. *Adequacy and linear types*: all CbN typing rules but many assign linear types. And many is used only as right premise of the rules app and ES, to derive M. It is with respect to linear types, in fact, that the adequacy of the system is going to be proved: a term is CbN normalising if and only if it is typable with a linear type, given by Theorem 1 and Theorem 2 below.

Tight derivations. A term may have several derivations, indexed by different pairs (m, e). They always provide upper bounds on CbN evaluation lengths. The interesting aspect of our type systems, however, is that there is a simple description of a class of derivations that provide *exact* bounds for these quantities, as we shall show. Their definition relies on the normal type constant.

Definition 1 (Tight derivations for CbN). *A derivation $\Phi \triangleright_{\mathrm{cbn}} \Gamma \vdash^{(m,e)} t{:}L$ is* tight *if $L = \mathsf{normal}$ and Γ is empty.*

Example 2. Let us return to the term $t \coloneqq ((\lambda x.\lambda y.xx)(II))(II)$ used in Example 1 for explaining the difference in reduction lengths among the different strategies. We now give a derivation for it in the CbN type system.

First, let us shorten normal to n. Then, we define Φ as the following derivation for the subterm $\lambda x.\lambda y.xx$ of t:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x:[[\mathsf{n}]\multimap\mathsf{n}]\vdash^{(0,1)} x:[\mathsf{n}]\multimap\mathsf{n}}\;\mathsf{ax} \qquad \dfrac{\dfrac{}{x:[\mathsf{n}]\vdash^{(0,1)} x:\mathsf{n}}\;\mathsf{ax}}{x:[\mathsf{n}]\vdash^{(0,1)} x:[\mathsf{n}]}\;\mathsf{many}}{x:[\mathsf{n},[\mathsf{n}]\multimap\mathsf{n}]\vdash^{(1,2)} xx:\mathsf{n}}\;\mathsf{app}}{x:[\mathsf{n},[\mathsf{n}]\multimap\mathsf{n}]\vdash^{(1,2)} \lambda y.xx:\mathbf{0}\multimap\mathsf{n}}\;\mathtt{fun}}{\vdash^{(1,2)} \lambda x.\lambda y.xx:[\mathsf{n},[\mathsf{n}]\multimap\mathsf{n}]\multimap(\mathbf{0}\multimap\mathsf{n})}\;\mathtt{fun}$$

Now, we need two derivations for II, one of type n, given by Ψ as follows

$$\dfrac{\dfrac{\dfrac{}{z:[\mathsf{n}]\vdash^{(0,1)} z:\mathsf{n}}\;\mathsf{ax}}{\vdash^{(0,1)} \lambda z.z:[\mathsf{n}]\multimap\mathsf{n}}\;\mathtt{fun} \qquad \dfrac{\dfrac{}{\vdash^{(0,0)} \lambda w.w:\mathsf{n}}\;\mathsf{normal}}{\vdash^{(0,0)} \lambda w.w:[\mathsf{n}]}\;\mathsf{many}}{\vdash^{(1,1)} II:\mathsf{n}}\;\mathsf{app}$$

and one of type $[\mathsf{n}]\multimap\mathsf{n}$, given by Ξ as follows

$$\dfrac{\dfrac{\dfrac{}{z : [[\mathsf{n}] \multimap \mathsf{n}] \vdash^{(0,1)} z : [\mathsf{n}] \multimap \mathsf{n}}\ \mathtt{ax}}{\vdash^{(0,1)} \lambda z.z : [[\mathsf{n}] \multimap \mathsf{n}] \multimap ([\mathsf{n}] \multimap \mathsf{n})}\ \mathtt{fun} \qquad \dfrac{\dfrac{\dfrac{}{w : [\mathsf{n}] \vdash^{(0,1)} w : \mathsf{n}}\ \mathtt{ax}}{\vdash^{(0,1)} \lambda w.w : [\mathsf{n}] \multimap \mathsf{n}}\ \mathtt{fun}}{\vdash^{(0,1)} \lambda w.w : [[\mathsf{n}] \multimap \mathsf{n}]}\ \mathtt{many}}{\vdash^{(1,2)} II : [\mathsf{n}] \multimap \mathsf{n}}\ \mathtt{app}$$

Finally, we put Φ, Ψ and Ξ together in the following derivation Θ for $t = (s(II))(II)$, where $s \coloneqq \lambda x.\lambda y.xx$ and $\mathsf{n}^{[\mathsf{n}]} \coloneqq [\mathsf{n}] \multimap \mathsf{n}$

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\Phi \\ \vdash^{(1,2)} s : [\mathsf{n}, \mathsf{n}^{[\mathsf{n}]}] \multimap (\mathbf{0} \multimap \mathsf{n})\end{array} \qquad \dfrac{\begin{array}{c}\vdots\,\Psi \\ \vdash^{(1,1)} II : \mathsf{n}\end{array} \quad \begin{array}{c}\vdots\,\Xi \\ \vdash^{(1,2)} II : \mathsf{n}^{[\mathsf{n}]}\end{array}}{\vdash^{(2,3)} II : [\mathsf{n}, \mathsf{n}^{[\mathsf{n}]}]}\ \mathtt{many}}{\vdash^{(4,5)} s(II) : \mathbf{0} \multimap \mathsf{n}}\ \mathtt{app} \qquad \dfrac{}{\vdash^{(0,0)} II : \mathbf{0}}\ \mathtt{many}}{\vdash^{(5,5)} (s(II))(II) : \mathsf{n}}\ \mathtt{app}$$

Note that that Θ is a tight derivation and the indices $(5,5)$ correspond exactly to the number of $\mathtt{m}_{\mathrm{cbn}}$-steps and $\mathtt{e}_{\mathrm{cbn}}$-steps, respectively, from t to its cbn-normal form, as shown in Example 1. Theorem 1 below shows that this is not by chance: tight derivations are minimal and provide exact bounds to evaluation lengths.

The next two subsections prove the two halves of the properties of the CbN type system, namely correctness and completeness.

4.1 CbN Correctness

Correctness is the fact that every typable term is CbN normalising. In our setting it comes with additional quantitative information: the indices m and e of a derivation $\Phi \triangleright_{\mathrm{cbn}} \Gamma \vdash^{(m,e)} t : L$ provide bounds for the length of the CbN evaluation of t, that are exact when the derivation is tight.

The proof technique is standard. Moreover, the correctness theorems for CbV and CbNeed in the next sections follow *exactly* the same structure. The proof relies on a quantitative subject reduction property showing that m decreases by *exactly one* at each $\mathtt{m}_{\mathrm{cbn}}$-step, and similarly for e and $\mathtt{e}_{\mathrm{cbn}}$-steps. In turn, subject reduction relies on a linear substitution lemma. Last, correctness for *tight* derivations requires a further property of normal forms.

Let us point out that correctness is stated with respect to closed terms only, but the auxiliary results have to deal with open terms, since they are proved by inductions (over predicates defined by induction) over the structure of terms.

Linear Substitution. The linear substitution lemma states that substituting over a variable occurrence as in the exponential rule consumes exactly one linear type and decreases of one the exponential index e.

Lemma 2 (CbN linear substitution). *If $\Phi \triangleright_{\mathrm{cbn}} \Gamma, x : M \vdash^{(m,e)} C\langle\langle x \rangle\rangle : L$ then there is a splitting $M = [L'] \uplus N$ such that for every derivation $\Psi \triangleright_{\mathrm{cbn}} \Pi \vdash^{(m',e')} t : L'$ there is a derivation $\Theta \triangleright_{\mathrm{cbn}} \Gamma \uplus \Pi, x : N \vdash^{(m+m',e+e'-1)} C\langle\langle t \rangle\rangle : L$.*

The proof is by induction over CbN evaluation contexts.

Quantitative Subject Reduction. A key point of multi types is that the size of type derivations shrinks after every evaluation step, which is what allows to bound evaluation lengths. Remarkably, the size (defined as the sum of the indices) shrinks by exactly 1 at every evaluation step.

Proposition 2 (Quantitative subject reduction for CbN). *Let* $\Phi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e)} t : L$ *be a derivation.*

1. Multiplicative*: if* $t \to_{\mathtt{m}_{\text{cbn}}} s$ *then* $m \geq 1$ *and there exists a derivation* $\Psi \triangleright_{\text{cbn}} \Gamma \vdash^{(m-1,e)} s : L$*.*
2. Exponential*: if* $t \to_{\mathtt{e}_{\text{cbn}}} s$ *then* $e \geq 1$ *and there exists a derivation* $\Psi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e-1)} s : L$*.*

The proof is by induction on $t \to_{\mathtt{m}_{\text{cbn}}} s$ and $t \to_{\mathtt{e}_{\text{cbn}}} s$, using the linear substitution lemma for the root exponential step.

Tightness and Normal Forms. Since the indices are always non-negative, quantitative subject reduction (Proposition 2) implies that they bound evaluation lengths. The bound is not necessarily exact, as derivations of normal forms can have strictly positive indices. If they are tight, however, they are indexed by $(0, 0)$, as we now show. The proof of this fact (by induction on the predicate normal) requires a slightly different statement, for the induction to go through.

Proposition 3 (normal typing of normal forms for CbN). *Let* t *be such that* $\mathsf{normal}(t)$*, and* $\Phi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e)} t : \mathsf{normal}$ *be a derivation. Then* Γ *is empty, and so* Φ *is tight, and* $m = e = 0$*.*

The Tight Correctness Theorem. The theorem is then proved by a straightforward induction on the evaluation length relying on quantitative subject reduction (Proposition 2) for the inductive case, and the properties of tight typings for normal forms (Proposition 3) for the base case.

Theorem 1 (CbN tight correctness). *Let* t *be a closed term. If* $\Phi \triangleright_{\text{cbn}} \vdash^{(m,e)} t : L$ *then there is* s *such that* $d \colon t \to^{*}_{\text{cbn}} s$*,* $\mathsf{normal}(s)$*,* $|d|_{\mathtt{m}} \leq m$*,* $|d|_{\mathtt{e}} \leq e$*. Moreover, if* Φ *is tight then* $|d|_{\mathtt{m}} = m$ *and* $|d|_{\mathtt{e}} = e$*.*

Note that Theorem 1 implicitly states that tight derivations have *minimal* size among derivations.

4.2 CbN Completeness

Completeness is the fact that every normalising term has a (tight) type derivation. As for correctness, the completeness theorem is always obtained via three intermediate steps, dual to those for correctness.

Normal Forms. The first step is to prove (by induction on the predicate normal) that every normal form is typable, and is actually typable with a tight derivation.

Proposition 4 (Normal forms are tightly typable for CbN). *Let* t *be such that* $\mathsf{normal}(t)$*. Then there is tight derivation* $\Phi \triangleright_{\text{cbn}} \vdash^{(0,0)} t : \mathsf{normal}$*.*

Linear Removal. In order to prove subject expansion, we have to first show that typability can also be pulled back along substitutions, via a linear removal lemma dual to the linear substitution lemma.

Lemma 3 (Linear removal for CbN). *Let $\Phi \triangleright_{\mathrm{cbn}} \Gamma; x : M \vdash^{(m,e)} C\langle\langle s\rangle\rangle : L$, where $x \notin \mathtt{fv}(s)$. Then there exist*

- *a linear type L',*
- *a derivation $\Phi_s \triangleright_{\mathrm{cbn}} \Gamma_s \vdash^{(m_s,e_s)} s : L'$, and*
- *a derivation $\Phi_{C\langle\langle x\rangle\rangle} \triangleright_{\mathrm{cbn}} \Gamma', x : M \uplus [L'] \vdash^{(m',e')} C\langle\langle x\rangle\rangle : L$*

such that

- Type contexts*: $\Gamma = \Gamma_s \uplus \Gamma'$.*
- Indices*: $(m,e) = (m' + m_s, e' + e_s - 1)$.*

Quantitative Subject Expansion. This property is the dual of subject reduction.

Proposition 5 (Quantitative subject expansion for CbN). *Let $\Phi \triangleright_{\mathrm{cbn}} \Gamma \vdash^{(m,e)} s : L$ be a derivation.*

1. Multiplicative*: if $t \to_{\mathtt{m}_{\mathrm{cbn}}} s$ then there is a derivation $\Psi \triangleright_{\mathrm{cbn}} \Gamma \vdash^{(m+1,e)} t : L$.*
2. Exponential*: if $t \to_{\mathtt{e}_{\mathrm{cbn}}} s$ then there is a derivation $\Psi \triangleright_{\mathrm{cbn}} \Gamma \vdash^{(m,e+1)} t : L$.*

The proof is by induction on $t \to_{\mathtt{m}_{\mathrm{cbn}}} s$ and $t \to_{\mathtt{e}_{\mathrm{cbn}}} s$, using the linear removal lemma for the root exponential step.

The Tight Completeness Theorem. The theorem is proved by a straightforward induction on the evaluation length relying on quantitative subject expansion (Proposition 5) in the inductive case, and the existence of tight typings for normal forms (Proposition 4) in the base case.

Theorem 2 (CbN tight completeness). *Let t be a closed term. If $d : t \to^*_{\mathrm{cbn}} s$ and $\mathsf{normal}(s)$ then there is a tight derivation $\Phi \triangleright_{\mathrm{cbn}} \vdash^{(|d|_{\mathtt{m}},|d|_{\mathtt{e}})} t : \mathsf{normal}$.*

Back to Erasing Steps. Our system can be easily adapted to measure also garbage collection steps (the CbN erasing rule is just before Example 1, page 7). First, a new, third index g on judgements is necessary. Second, one needs to distinguish the erasing and non-erasing cases of the the $\mathtt{app}$ and $\mathtt{ES}$ rules, discriminated by the $\mathbf{0}$ type. For instance, the $\mathtt{ES}$ rules are (the $\mathtt{app}$ rules are similar):

$$\frac{\Gamma \vdash^{(m,e,g)} t : L \quad \Gamma(x) = \mathbf{0}}{\Gamma \vdash^{(m,e,g+1)} t[x\leftarrow s] : L}\ \mathtt{ES}_{\mathrm{gc}} \qquad \frac{\Gamma, x : M \vdash^{(m,e,g)} t : L \quad \Pi \vdash^{(m',e',g')} s : M \quad M \neq \mathbf{0}}{\Gamma \uplus \Pi \vdash^{(m+m',e+e',g+g')} t[x\leftarrow s] : L}\ \mathtt{ES}$$

The index g bounds to the number of erasing steps. In the closed case, however, the bound cannot be, in general, exact. Variables typed with $\mathbf{0}$ by Γ do not exactly match variables not appearing in the typed term (that is the condition triggering the erasing step), because a variable typed with $\mathbf{0}$ may appear in the body of abstractions typed with the normal rule, as such bodies are not typed.

It is reasonable to assume that exact bounds for erasing steps can only by provided by a type system characterising strong evaluation, whose typing rules have to inspect abstraction bodies. These erasing typing rules are nonetheless going to play a role in the design of the CbNeed system in Sect. 6.

4.3 CbN Model

The idea to build the denotational model from the multi type system is that the interpretation (or semantics) of a term is simply the set of its type assignments, *i.e.* the set of its derivable types together with their type contexts. More precisely, let t be a term and $x_1, \ldots, x_n$ (with $n \geq 0$) be pairwise distinct variables. If $\mathtt{fv}(t) \subseteq \{x_1, \ldots, x_n\}$, we say that the list $\vec{x} = (x_1, \ldots, x_n)$ is *suitable for t*. If $\vec{x} = (x_1, \ldots, x_n)$ is suitable for t, the (*relational*) *semantics of t for $\vec{x}$* is

$$[\![t]\!]^{\mathrm{CbN}}_{\vec{x}} \coloneqq \{((M_1, \ldots, M_n), L) \mid \exists\, \Phi \rhd_{\mathrm{cbn}} x_1 : M_1, \ldots, x_n : M_n \vdash^{(m,e)} t : L\}\,.$$

Subject reduction (Proposition 2) and expansion (Proposition 5) guarantee that the semantics $[\![t]\!]^{\mathrm{CbN}}_{\vec{x}}$ of t (for *any* term t, possibly open) is *invariant* by CbN evaluation. Correctness (Theorem 1) and completeness (Theorem 2) guarantee that, given a *closed* term t, its interpretation $[\![t]\!]^{\mathrm{CbN}}_{\vec{x}}$ is non-empty if and only if t is CbN normalisable, that is, they imply that relational semantics is *adequate*.

In fact, adequacy also holds with respect to open terms. The issue in that case is that the characterisation of tight derivations is more involved, see Accattoli, Graham-Lengrand and Kesner's [7]. Said differently, weaker correctness and completeness theorems without exact bounds also hold in the open case. The same is true for the CbV and CbNeed systems of the next sections.

5 Types by Value

Here we introduce Ehrhard's CbV multi type system [29] adapted to our presentation of CbV in the LSC, and prove its properties. The system is similar, and yet in many aspects dual, to the CbN one, in particular the grammar of types is different. Linear types for CbV are defined by:

$$\textsc{CbV linear types} \qquad\qquad L, L' ::= M \multimap N$$

Multi(-sets) types are defined as in Sect. 3, relatively to CbV linear types. Note that linear types now have a multi type both as source and as target, and that the normal constant is absent—in CbV, its role is played by $\mathbf{0}$.

The typing rules are in Fig. 2. It is a type-based presentation of the relational model of the CbV λ-calculus induced by relational model of linear logic via the CbV translation of λ-calculus into linear logic. Some remarks:

- *Right-hand types*: all rules but $\mathtt{fun}$ assign a multi type to the term on the right-hand side, and not a linear type as in CbN.
- *Abstractions and* many: the many rule has a restricted form with respect to the CbN one, it can only be applied to abstractions, that in turn are the only terms that can be typed with a linear type.
- *Indices*: note as the indices are however incremented (on ax and app) and summed (in many and ES) exactly as in the CbN system.

$$\frac{}{x : M \vdash^{(0,1)} x : M}\ \mathsf{ax} \qquad \frac{\Gamma \vdash^{(m,e)} t : [M \multimap N] \quad \Pi \vdash^{(m',e')} s : M}{\Gamma \uplus \Pi \vdash^{(m+m'+1,e+e')} ts : N}\ \mathsf{app}$$

$$\frac{\Gamma, x : N \vdash^{(m,e)} t : M}{\Gamma \vdash^{(m,e)} \lambda x.t : N \multimap M}\ \mathtt{fun} \qquad \frac{(\Pi_i \vdash^{(m_i,e_i)} \lambda x.t : L_i)_{i \in J}}{\biguplus_{i \in J} \Pi_i \vdash^{(\sum_{i \in J} m_i, \sum_{i \in J} e_i)} \lambda x.t : [L_i]_{i \in J}}\ \mathsf{many}$$

$$\frac{\Gamma, x : N \vdash^{(m,e)} t : M \quad \Pi \vdash^{(m',e')} s : N}{\Gamma \uplus \Pi \vdash^{(m+m',e+e')} t[x \leftarrow s] : M}\ \mathtt{ES}$$

Fig. 2. Type system for CbV evaluation.

Intuitions: the Empty Type $\mathbf{0}$. The empty multi-set type $\mathbf{0}$ plays a special role in CbV. As in CbN, it is the type of terms that can be erased, but, in contrast to CbN, not every term is erasable in CbV.

In the CbN multi type system every term, even a diverging one, is typable with $\mathbf{0}$. On the one hand, this is correct, because in CbN every term can be erased, and erased terms can also be divergent, because they are never evaluated. On the other hand, adequacy is formulated with respect to non-empty types: a term terminates if and only if it is typable with a non-empty type.

In CbV, instead, terms have to be evaluated before being erased; and, of course, their evaluation has to terminate. Thus, terminating terms and erasable terms coincide. Since the multi type system is meant to characterise terminating terms, in CbV a term is typable if and only if it is typable with $\mathbf{0}$, as we shall prove in this section. Then the empty type is not a degenerate type excluded for adequacy from the interesting types of a term, as in CbN, it rather is *the* type, characterising (adequate) typability altogether. And this is also the reason for the absence of the constant normal—one way to see it is that in CbV $\mathsf{normal} = \mathbf{0}$.

Note that, in particular, in a type judgement $\Gamma \vdash t : M$ the type context Γ may give the empty type to a variable x occurring in t, as for instance in the axiom $x : \mathbf{0} \vdash x : \mathbf{0}$—this may seem very strange to people familiar with CbN multi types. We hope that instead, according to the provided intuition that $\mathbf{0}$ is the type of termination, it would rather seem natural.

Definition 2 (Tight derivation for CbV). *A derivation* $\Phi \triangleright_{\mathrm{cbv}} \Gamma \vdash^{(m,e)} t : M$ *is* tight *if* $M = \mathbf{0}$ *and* Γ *is empty.*

Example 3. Let's consider again the term $t := ((\lambda x.\lambda y.xx)(II))(II)$ of Example 1 (where $I := \lambda z.z$), for which a CbN tight derivation was given in Example 2, and let us type it in the CbV system with a tight derivation.

We define the following derivation Φ_1 for the subterm $s := \lambda x.\lambda y.xx$ of t

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : [\mathbf{0} \multimap \mathbf{0}] \vdash^{(0,1)} x : [\mathbf{0} \multimap \mathbf{0}]}\,\mathsf{ax} \qquad \dfrac{}{x : \mathbf{0} \vdash^{(0,1)} x : \mathbf{0}}\,\mathsf{ax}}{x : [\mathbf{0} \multimap \mathbf{0}] \vdash^{(1,2)} xx : \mathbf{0}}\,\mathsf{app}}{x : [\mathbf{0} \multimap \mathbf{0}] \vdash^{(1,2)} \lambda y.xx : \mathbf{0} \multimap \mathbf{0}}\,\mathtt{fun}}{x : [\mathbf{0} \multimap \mathbf{0}] \vdash^{(1,2)} \lambda y.xx : [\mathbf{0} \multimap \mathbf{0}]}\,\mathsf{many}}{\vdash^{(1,2)} s : [\mathbf{0} \multimap \mathbf{0}] \multimap [\mathbf{0} \multimap \mathbf{0}]}\,\mathtt{fun}}{\vdash^{(1,2)} s : [[\mathbf{0} \multimap \mathbf{0}] \multimap [\mathbf{0} \multimap \mathbf{0}]]}\,\mathsf{many}$$

Note that $[\mathbf{0} \multimap \mathbf{0}] \uplus \mathbf{0} = [\mathbf{0} \multimap \mathbf{0}]$, which explains the shape of the type context in the conclusion of the app rule. Next, we define the derivation Φ_2 as follows

$$\dfrac{\dfrac{\dfrac{\dfrac{}{z : [\mathbf{0} \multimap \mathbf{0}] \vdash^{(0,1)} z : [\mathbf{0} \multimap \mathbf{0}]}\,\mathsf{ax}}{\vdash^{(0,1)} \lambda z.z : [\mathbf{0} \multimap \mathbf{0}] \multimap [\mathbf{0} \multimap \mathbf{0}]}\,\mathtt{fun}}{\vdash^{(0,1)} \lambda z.z : [[\mathbf{0} \multimap \mathbf{0}] \multimap [\mathbf{0} \multimap \mathbf{0}]]}\,\mathsf{many} \qquad \dfrac{\dfrac{\dfrac{}{w : \mathbf{0} \vdash^{(0,1)} w : \mathbf{0}}\,\mathsf{ax}}{\vdash^{(0,1)} \lambda w.w : \mathbf{0} \multimap \mathbf{0}}\,\mathtt{fun}}{\vdash^{(0,1)} \lambda w.w : [\mathbf{0} \multimap \mathbf{0}]}\,\mathsf{many}}{\vdash^{(1,2)} II : [\mathbf{0} \multimap \mathbf{0}]}\,\mathsf{app}$$

and the derivation Φ_3 as follows

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x' : \mathbf{0} \vdash^{(0,1)} x' : \mathbf{0}}\,\mathsf{ax}}{\vdash^{(0,1)} \lambda x'.x' : \mathbf{0} \multimap \mathbf{0}}\,\mathtt{fun}}{\vdash^{(0,1)} \lambda x'.x' : [\mathbf{0} \multimap \mathbf{0}]}\,\mathsf{many} \qquad \dfrac{}{\vdash^{(0,0)} I : \mathbf{0}}\,\mathsf{many}}{\vdash^{(1,1)} II : \mathbf{0}}\,\mathsf{app}$$

Finally, we put Φ_1, Φ_2 and Φ_3 together in the following derivation Φ for t

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\Phi_1 \\ \vdash^{(1,2)} s : [[\mathbf{0} \multimap \mathbf{0}] \multimap [\mathbf{0} \multimap \mathbf{0}]]\end{matrix} \qquad \begin{matrix}\vdots\,\Phi_2 \\ \vdash^{(1,2)} II : [\mathbf{0} \multimap \mathbf{0}]\end{matrix}}{\vdash^{(3,4)} (\lambda x.\lambda y.xx)(II) : [\mathbf{0} \multimap \mathbf{0}]}\,\mathsf{app} \qquad \begin{matrix}\vdots\,\Phi_3 \\ \vdash^{(1,1)} II : \mathbf{0}\end{matrix}}{\vdash^{(5,5)} ((\lambda x.\lambda y.xx)(II))(II) : \mathbf{0}}\,\mathsf{app}$$

Note that the indices $(5, 5)$ correspond exactly to the number of $\mathtt{m}_{\text{cbv}}$-steps and $\mathtt{e}_{\text{cbv}}$-steps, respectively, from t to its CbV normal form, as shown in Example 1, and that Φ is a tight derivation. Forthcoming Theorem 3 shows that CbV tight derivations are minimal and provide exact bounds to evaluation lengths in CbV.

Correctness (*i.e.* typability implies normalisability) and *completeness* (*i.e.* normalisability implies typability) of the CbV type system with respect to CbV evaluation (together with quantitative information about evaluation lengths) follow exactly the same pattern of the CbN case, *mutatis mutandis.*

5.1 CbV Correctness

Lemma 4 (CbV linear substitution). *Let* $\Phi \triangleright_{\text{cbv}} \Gamma, x : M \vdash^{(m,e)} V\langle\langle x \rangle\rangle : N$*. Then there exists a splitting* $M = O \uplus P$ *such that, for every derivation* $\Psi \triangleright_{\text{cbv}} \Pi \vdash^{(m',e')} v : O$*, there is a derivation* $\Phi' \triangleright_{\text{cbv}} \Gamma \uplus \Pi, x : P \vdash^{(m+m',e+e'-1)} V\langle\langle v \rangle\rangle : N$*.*

Proposition 6 (Quantitative subject reduction for CbV). *Let $\Phi \triangleright_{\text{cbv}} \Gamma \vdash^{(m,e)} t : M$ be a derivation.*

1. Multiplicative*: if $t \to_{\mathtt{m}_{\text{cbv}}} t'$ then $m \geq 1$ and there exists a derivation $\Phi' \triangleright_{\text{cbv}} \Gamma \vdash^{(m-1,e)} t' : M$.*
2. Exponential*: if $t \to_{\mathtt{e}_{\text{cbv}}} t'$ then $e \geq 1$ and there exists a derivation $\Phi' \triangleright_{\text{cbv}} \Gamma \vdash^{(m,e-1)} t' : M$.*

Proposition 7 (Tight typings for normal forms for CbV). *Let $\Phi \triangleright_{\text{cbv}} \Gamma \vdash^{(m,e)} t : \mathbf{0}$ be a derivation, with $\mathsf{normal}_{\text{cbv}}(t)$. Then Γ is empty, and so Φ is tight, and $m = e = 0$.*

Theorem 3 (CbV tight correctness). *Let t be a closed term. If $\Phi \triangleright_{\text{cbv}} \Gamma \vdash^{(m,e)} t : M$ then there is s such that $d \colon t \to^*_{\text{cbv}} s$, $\mathsf{normal}_{\text{cbv}}(s)$, $|d|_{\mathtt{m}} \leq m$, $|d|_{\mathtt{e}} \leq e$. Moreover, if Φ is tight then $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$.*

5.2 CbV Completeness

Proposition 8 (Normal forms are tightly typable for CbV). *Let t be such that $\mathsf{normal}_{\text{cbv}}(t)$. Then there exists a tight derivation $\Phi \triangleright_{\text{cbv}} \vdash^{(0,0)} t : \mathbf{0}$.*

Lemma 5 (Linear removal for CbV). *Let $\Phi \triangleright_{\text{cbv}} \Gamma, x : M \vdash^{(m,e)} V\langle\langle v \rangle\rangle : N$ where $x \notin \mathtt{fv}(v)$. Then, there exist*
- *a multi type M' and two type contexts Γ' and Π,*
- *a derivation $\Psi \triangleright_{\text{cbv}} x : M \uplus M', \Pi \vdash^{(m'',e'')} V\langle\langle x \rangle\rangle : N$, and*
- *a derivation $\Phi' \triangleright_{\text{cbv}} \Gamma' \vdash^{(m',e')} v : M'$*

such that
- Type contexts: $\Gamma = \Gamma' \uplus \Pi$,
- Indices: $(m,e) = (m' + m'', e' + e'' - 1)$.

Proposition 9 (Quantitative subject expansion for CbV). *Let $\Phi' \triangleright_{\text{cbv}} \Gamma \vdash^{(m,e)} t' : M$ be a derivation.*

1. Multiplicative*: if $t \to_{\mathtt{m}_{\text{cbv}}} t'$ then there is a derivation $\Phi \triangleright_{\text{cbv}} \Gamma \vdash^{(m+1,e)} t : M$.*
2. Exponential*: if $t \to_{\mathtt{e}_{\text{cbv}}} t'$ then there is a derivation $\Phi \triangleright_{\text{cbv}} \Gamma \vdash^{(m,e+1)} t : M$.*

Theorem 4 (CbV tight completeness). *Let t be a closed term. If $d \colon t \to^*_{\text{cbv}} s$ with $\mathsf{normal}_{\text{cbv}}(s)$, then there is a tight derivation $\Phi \triangleright_{\text{cbv}} \vdash^{(|d|_{\mathtt{m}},|d|_{\mathtt{e}})} t : \mathbf{0}$.*

CbV Model. The interpretation of terms with respect to the CbV system is defined as follows (where $\vec{x} = (x_1, \ldots, x_n)$ is a list of variables suitable for t):

$$[\![t]\!]^{\text{CbV}}_{\vec{x}} \coloneqq \{((M_1, \ldots, M_n), N) \mid \exists\, \Phi \triangleright_{\text{cbv}} x_1 : M_1, \ldots, x_n : M_n \vdash^{(m,e)} t : N\}.$$

Note that rule $\mathtt{fun}$ assigns a linear type but the interpretation considers only multi types. The *invariance* and the *adequacy* of $[\![t]\!]^{\text{CbV}}_{\vec{x}}$ with respect to CbV evaluation are obtained exactly as for the CbN case.

$$\frac{}{x : M \vdash^{(0,1)} x : M}\ \mathsf{ax} \qquad \frac{\Gamma \vdash^{(m,e)} t : [N \multimap M] \quad \Pi \vdash^{(m',e')} s : N}{\Gamma \uplus \Pi \vdash^{(m+m'+1,e+e')} ts : M}\ \mathtt{app}$$

$$\frac{}{\vdash^{(0,0)} t : \mathbf{0}}\ \mathsf{many}_0 \qquad \frac{(\Pi_i \vdash^{(m_i,e_i)} \lambda x.t : L_i)_{i \in J} \quad J \neq \emptyset}{\biguplus_{i \in J} \Pi_i \vdash^{(\sum_{i \in J} m_i, \sum_{i \in J} e_i)} \lambda x.t : [L_i]_{i \in J}}\ \mathsf{many}_{>0}$$

$$\frac{x : N; \Gamma \vdash^{(m,e)} t : M}{\Gamma \vdash^{(m,e)} \lambda x.t : N \multimap M}\ \mathtt{fun} \qquad \frac{x : N; \Gamma \vdash^{(m,e)} t : M \quad \Pi \vdash^{(m',e')} s : N}{\Gamma \uplus \Pi \vdash^{(m+m',e+e')} t[x \leftarrow s] : M}\ \mathsf{ES}$$

$$\frac{}{\vdash^{(0,0)} \lambda x.t : \mathsf{normal}}\ \mathsf{normal}$$

Fig. 3. Naïve type system for CbNeed evaluation.

6 Types by Need

CbNeed as a Blend of CbN and CbV. The multi type system for CbNeed is obtained by carefully blending ingredients from the CbN and CbV ones:

– *Wise erasures from CbN*: in CbN wise erasures are induced by the fact that the empty multi type $\mathbf{0}$ (the type of erasable terms) and the linear type normal (the type of normalisable terms) are distinct and every term is typable with $\mathbf{0}$ by using the many rule with 0 premises. Adequacy is then formulated with respect to (non-empty) linear types.

– *Wise duplications from CbV*: in CbV wise duplications are due to two aspects. First, only abstractions can be collected in multi-sets by rule many. This fact accounts for the evaluation of arguments to normal form—that is, abstractions—before being substituted. Second, terms are typed with multi types instead of linear types. Roughly, this second fact allows the first one to actually work because the argument is reduced once for a whole multi set of types, and not once for each element of the multi set, as in CbN.

It seems then that a type system for CbNeed can easily be obtained by basically adopting the CbV system plus

– separating $\mathbf{0}$ and normal, that is, adding normal to the system;
– modifying the many rule by distinguishing two cases: with 0 premises it can assign $\mathbf{0}$ to whatever term—as in CbN—otherwise it is forced to work on abstractions, as in CbV;
– restricting adequacy to non-empty types.

Therefore, the grammar of linear types is:

$$\text{CbNeed linear types} \qquad L, L' ::= \mathsf{normal} \mid M \multimap N$$

Multi(-sets) types are defined as in Sect. 3, relatively to CbNeed linear types. The rules of this *naïve system* for CbNeed are in Fig. 3.

Issue with the Naïve System. Unfortunately, the naïve system does not work: tight derivations—defined as expected: empty type context and the term typed with

$[\mathsf{normal}]$—do not provide exact bounds. The problem is that the naïve blend of ingredients allows derivations of $\mathbf{0}$ with strictly positive indices m and e. Instead, derivations of $\mathbf{0}$ should always have 0 in both indices—as is the case when they are derived with a many_0 rule with 0 premises—because they correspond to terms to be erased, that are not evaluated in CbNeed. For any term t, indeed, one can for instance derive the following derivation Φ:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash^{(0,0)} x:\mathbf{0}}\,\mathsf{many}_0}{\vdash^{(0,0)} \lambda x.x:\mathbf{0}\multimap\mathbf{0}}\,\mathtt{fun}}{\vdash^{(0,0)} \lambda x.x:[\mathbf{0}\multimap\mathbf{0}]}\,\mathsf{many}_{>0} \qquad \dfrac{}{\vdash^{(0,0)} t:\mathbf{0}}\,\mathsf{many}_0}{\vdash^{(1,0)} (\lambda x.x)t:\mathbf{0}}\,\mathtt{app}$$

Note that introducing $\vdash^{(0,1)} x:\mathbf{0}$ with rule ax rather than via many_0 (the typing context $x:\mathbf{0}$ is equivalent to the empty type context) would give a derivation with final judgement $\vdash^{(1,1)} (\lambda x.x)t:\mathbf{0}$—thus, the system messes up both indices.

Such bad derivations of $\mathbf{0}$ are not a problem *per se*, because in CbNeed one expects correctness and completeness to hold only for derivations of non-empty multi types. However, they do mess up also derivations of non-empty multi types because they can still appear *inside* tight derivations, as sub-derivations of sub-terms to be erased; consider for instance:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash^{(0,0)} I:\mathsf{normal}}\,\mathsf{normal}}{\vdash^{(0,0)} I:[\mathsf{normal}]}\,\mathsf{many}_{>0}}{\vdash^{(0,0)} \lambda y.I:\mathbf{0}\multimap[\mathsf{normal}]}\,\mathtt{fun}}{\vdash^{(0,0)} \lambda y.I:[\mathbf{0}\multimap[\mathsf{normal}]]}\,\mathsf{many}_{>0} \qquad \begin{array}{c}\vdots\,\Phi\\ \vdash^{(1,0)} (\lambda x.x)t:\mathbf{0}\end{array}}{\vdash^{(2,0)} (\lambda y.I)((\lambda x.x)t):[\mathsf{normal}]}\,\mathtt{app}$$

The term normalises in just 1 $\mathtt{m}_{\mathrm{need}}$-step to $I[y\leftarrow(\lambda x.x)t]$ but the multiplicative index of the derivation is 2. The mismatch is due to a bad derivation of $\mathbf{0}$ used as right premise of an $\mathtt{app}$ rule. Similarly, the induced typing of $I[y\leftarrow(\lambda x.x)t]$ is an example of a bad derivation used as right premise of a rule $\mathtt{ES}$:

$$\dfrac{\dfrac{\dfrac{}{\vdash^{(0,0)} I:\mathsf{normal}}\,\mathsf{normal}}{\vdash^{(0,0)} I:[\mathsf{normal}]}\,\mathsf{many}_{>0} \qquad \begin{array}{c}\vdots\,\Phi\\ \vdash^{(1,0)} (\lambda x.x)t:\mathbf{0}\end{array}}{\vdash^{(1,0)} I[y\leftarrow(\lambda x.x)t]:[\mathsf{normal}]}\,\mathtt{ES}$$

The Actual Type System. Our solution to such an issue is to modify the system as to avoid as much as possible derivations of $\mathbf{0}$. The idea is that deriving $\mathbf{0}$ is only needed for the right premise of rules $\mathtt{app}$ and $\mathtt{ES}$, when $N = \mathbf{0}$, and so we add two dedicated rules $\mathtt{app}_{\mathrm{gc}}$ and $\mathtt{ES}_{\mathrm{gc}}$, and instead remove rule many_0 and forbid axioms to introduce $\mathbf{0}$—the system is in Fig. 4 and it is based on the same grammar of types of the naïve system. Note that rules $\mathtt{app}$ and $\mathtt{ES}$ now also require N to be different from $\mathbf{0}$, to avoid overlaps with $\mathtt{app}_{\mathrm{gc}}$ and $\mathtt{ES}_{\mathrm{gc}}$.

$$\dfrac{}{x : M \vdash^{(0,1)} x : M}\ \mathsf{ax} \qquad \dfrac{}{\vdash^{(0,0)} \lambda x.t : \mathsf{normal}}\ \mathsf{normal}$$

$$\dfrac{\Gamma, x : M \vdash^{(m,e)} t : N}{\Gamma \vdash^{(m,e)} \lambda x.t : M \multimap N}\ \mathtt{fun} \qquad \dfrac{(\Gamma_i \vdash^{(m_i,e_i)} \lambda x.t : L_i)_{i\in J} \quad J \neq \emptyset}{\biguplus_{i\in J} \Gamma_i \vdash^{(\Sigma_{i\in J} m_i, \Sigma_{i\in J} e_i)} \lambda x.t : [L_i]_{i\in J}}\ \mathsf{many}$$

$$\dfrac{\Gamma \vdash^{(m,e)} t : [\mathbf{0} \multimap M]}{\Gamma \vdash^{(m+1,e)} ts : M}\ \mathsf{app}_{\mathrm{gc}} \qquad \dfrac{\Gamma \vdash^{(m,e)} t : [N \multimap M] \quad \Pi \vdash^{(m',e')} s : N \quad N \neq \mathbf{0}}{\Gamma \uplus \Pi \vdash^{(m+m'+1,e+e')} ts : M}\ \mathsf{app}$$

$$\dfrac{\Gamma \vdash^{(m,e)} t : M \quad \Gamma(x) = \mathbf{0}}{\Gamma \vdash^{(m,e)} t[x{\leftarrow}s] : M}\ \mathtt{ES}_{\mathrm{gc}} \qquad \dfrac{\Gamma, x : N \vdash^{(m,e)} t : M \quad \Pi \vdash^{(m',e')} s : N \quad N \neq \mathbf{0}}{\Gamma \uplus \Pi \vdash^{(m+m',e+e')} t[x{\leftarrow}s] : M}\ \mathtt{ES}$$

Fig. 4. Type system for CbNeed evaluation.

Note that the indices m and e are incremented and summed exactly as in the CbN and CbV type systems.

Definition 3 (Tight derivations for CbNeed). *A derivation $\Phi \rhd_{\text{need}} \Gamma \vdash^{(m,e)} t : M$ is* tight *if $M = [\mathsf{normal}]$ and Γ is empty.*

Example 4 (The needed one). We return to $t := ((\lambda x.\lambda y.xx)(II))(II)$ used in Example 1 and we give it a tight derivation in the CbNeed type system.

Again, we shorten normal to n. Then, we define Ψ as follows

$$\begin{prooftree}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$x : [[\mathsf{n}] \multimap [\mathsf{n}]] \vdash^{(0,1)} x : [[\mathsf{n}] \multimap [\mathsf{n}]]$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$x : [\mathsf{n}] \vdash^{(0,1)} x : [\mathsf{n}]$}
\RightLabel{app}
\BinaryInfC{$x : [\mathsf{n}, [\mathsf{n}] \multimap [\mathsf{n}]] \vdash^{(1,2)} xx : [\mathsf{n}]$}
\RightLabel{$\mathtt{fun}$}
\UnaryInfC{$x : [\mathsf{n}, [\mathsf{n}] \multimap [\mathsf{n}]] \vdash^{(1,2)} \lambda y.xx : \mathbf{0} \multimap [\mathsf{n}]$}
\RightLabel{many}
\UnaryInfC{$x : [\mathsf{n}, [\mathsf{n}] \multimap [\mathsf{n}]] \vdash^{(1,2)} \lambda y.xx : [\mathbf{0} \multimap [\mathsf{n}]]$}
\RightLabel{$\mathtt{fun}$}
\UnaryInfC{$\vdash^{(1,2)} \lambda x.\lambda y.xx : [\mathsf{n}, [\mathsf{n}] \multimap [\mathsf{n}]] \multimap [\mathbf{0} \multimap [\mathsf{n}]]$}
\RightLabel{many}
\UnaryInfC{$\vdash^{(1,2)} \lambda x.\lambda y.xx : [[\mathsf{n}, [\mathsf{n}] \multimap [\mathsf{n}]] \multimap [\mathbf{0} \multimap [\mathsf{n}]]]$}
\end{prooftree}$$

and, shortening $[\mathsf{n}] \multimap [\mathsf{n}]$ to n^{n}, we define Θ as follows

$$\begin{prooftree}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$z : [\mathsf{n}, \mathsf{n}^{\mathsf{n}}] \vdash^{(0,1)} z : [\mathsf{n}, \mathsf{n}^{\mathsf{n}}]$}
\RightLabel{$\mathtt{fun}$}
\UnaryInfC{$\vdash^{(0,1)} \lambda z.z : [\mathsf{n}, \mathsf{n}^{\mathsf{n}}] \multimap [\mathsf{n}, \mathsf{n}^{\mathsf{n}}]$}
\RightLabel{many}
\UnaryInfC{$\vdash^{(0,1)} \lambda z.z : [[\mathsf{n}, \mathsf{n}^{\mathsf{n}}] \multimap [\mathsf{n}, \mathsf{n}^{\mathsf{n}}]]$}
\AxiomC{}
\RightLabel{normal}
\UnaryInfC{$\vdash^{(0,0)} \lambda w.w : \mathsf{n}$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$w : [\mathsf{n}] \vdash^{(0,1)} w : [\mathsf{n}]$}
\RightLabel{$\mathtt{fun}$}
\UnaryInfC{$\vdash^{(0,1)} \lambda w.w : \mathsf{n}^{\mathsf{n}}$}
\RightLabel{many}
\BinaryInfC{$\vdash^{(0,1)} \lambda w.w : [\mathsf{n}, \mathsf{n}^{\mathsf{n}}]$}
\RightLabel{app}
\BinaryInfC{$\vdash^{(1,2)} II : [\mathsf{n}, \mathsf{n}^{\mathsf{n}}]$}
\end{prooftree}$$

Finally, we put Ψ and Θ together in the following derivation Φ for t

$$\begin{prooftree}
\AxiomC{$\vdots\ \Psi$}
\noLine
\UnaryInfC{$\vdash^{(1,2)} \lambda x.\lambda y.xx : [[\mathsf{n}, [\mathsf{n}] \multimap [\mathsf{n}]] \multimap [\mathbf{0} \multimap [\mathsf{n}]]]$}
\AxiomC{$\vdots\ \Theta$}
\noLine
\UnaryInfC{$\vdash^{(1,2)} II : [\mathsf{n}, \mathsf{n}^{\mathsf{n}}]$}
\RightLabel{app}
\BinaryInfC{$\vdash^{(3,4)} (\lambda x.\lambda y.xx)(II) : [\mathbf{0} \multimap [\mathsf{n}]]$}
\RightLabel{$\mathsf{app}_{\mathrm{gc}}$}
\UnaryInfC{$\vdash^{(4,4)} ((\lambda x.\lambda y.xx)(II))(II) : [\mathsf{n}]$}
\end{prooftree}$$

Note that the indices $(4,4)$ correspond exactly to the number of $\mathtt{m}_{\text{need}}$-steps and $\mathtt{e}_{\text{need}}$-steps respectively, from t to its need-normal form—as shown in Example 1—, and that Φ is a *tight* derivation. Forthcoming Theorem 5 shows once again that this is not by chance: CbNeed tight derivations are minimal and provides exact bounds to evaluation lengths in $\rightarrow_{\text{need}}$.

Remarkably, the technical development to prove *correctness* and *completeness* of the CbNeed type system with respect to CbNeed evaluation follows smoothly along the same lines of the two other systems, *mutatis mutandis.*

6.1 CbNeed Correctness

Lemma 6 (CbNeed linear substitution). *Let $\Phi_{E\langle\langle x\rangle\rangle} \triangleright_{\text{need}} x : M; \Gamma \vdash^{(m,e)} E\langle\langle x\rangle\rangle : O$ be a derivation and v a value such that $O \neq \mathbf{0}$ and E does not capture the free variables of v. Then there exists a splitting $M = M_1 \uplus M_2$, with $M_1 \neq \mathbf{0}$, such that for every derivation $\Psi \triangleright_{\text{need}} \Pi \vdash^{(m',e')} v : M_1$ there exists a derivation $\Phi_{E\langle\langle v\rangle\rangle} \triangleright_{\text{need}} x : M_2; \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} E\langle\langle v\rangle\rangle : O$.*

Proposition 10 (Quantitative subject reduction for CbNeed). *Let $\Phi \triangleright_{\text{need}} \Gamma \vdash^{(m,e)} t : M$ be a derivation such that $M \neq \mathbf{0}$.*
- Multiplicative*: if $t \rightarrow_{\mathtt{m}_{\text{need}}} s$ then $m \geq 1$ and there is a derivation $\Phi' \triangleright_{\text{need}} \Gamma \vdash^{(m-1,e)} t : M$.*
- Exponential*: if $t \rightarrow_{\mathtt{e}_{\text{need}}} s$ then $e \geq 1$ and there exists a derivation $\Phi' \triangleright_{\text{need}} \Gamma \vdash^{(m,e-1)} t : M$.*

Proposition 11 ($[\mathsf{normal}]$ typings for normal forms for CbNeed). *Let $\Phi \triangleright_{\text{need}} \Gamma \vdash^{(m,e)} t : [\mathsf{normal}]$ be a derivation, with $\mathsf{normal}(t)$. Then Γ is empty, and so Φ is tight, and $m = e = 0$.*

Theorem 5 (CbNeed tight correctness). *Let t be a closed term. If $\Phi \triangleright_{\text{need}} \vdash^{(m,e)} t : M$ then there is s such that $d\colon t \rightarrow^{*}_{\text{need}} s$, $\mathsf{normal}(s)$, $|d|_{\mathtt{m}} \leq m$, $|d|_{\mathtt{e}} \leq e$. Moreover, if Φ is tight then $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$.*

6.2 CbNeed Completeness

Proposition 12 (Normal forms are tightly typable for CbNeed). *Let t be such that $\mathsf{normal}(t)$. Then there is a tight derivation $\Phi \triangleright_{\text{need}} \vdash^{(0,0)} t : [\mathsf{normal}]$.*

Lemma 7 (Linear removal for CbNeed). *Let $\Phi \triangleright_{\text{need}} \Gamma \vdash^{(m,e)} E\langle\langle v\rangle\rangle : O$ be a derivation, with $O \neq \mathbf{0}$ and $x \notin \mathtt{fv}(v)$. Then there exist*
- *a multi type M,*
- *a derivation $\Phi_v \triangleright_{\text{need}} \Gamma_v \vdash^{(m_v,e_v)} v : M$, and*
- *a derivation $\Phi_{E\langle\langle x\rangle\rangle} \triangleright_{\text{need}} \Gamma' \uplus \{x\colon M\} \vdash^{(m',e')} E\langle\langle x\rangle\rangle : O$*

such that
- Type contexts*: $\Gamma = \Gamma' \uplus \Gamma_v$.*
- Indices*: $(m,e) = (m' + m_v, e' + e_v - 1)$.*

Proposition 13 (Quantitative subject expansion for CbNeed). *Let $\Phi \triangleright_{\text{need}} \Gamma \vdash^{(m,e)} s : M$ be a derivation such that $M \neq \mathbf{0}$. Then,*

- Multiplicative: *if $t \to_{\mathtt{m}_{\text{need}}} s$ then there is a derivation $\Phi' \triangleright_{\text{need}} \Gamma \vdash^{(m+1,e)} t : M$,*
- Exponential*: if $t \to_{\mathtt{e}_{\text{need}}} s$ then there is a derivation $\Phi' \triangleright_{\text{need}} \Gamma \vdash^{(m,e+1)} t : M$.*

Theorem 6 (CbNeed tight completeness). *Let t be a closed term. If $d\colon t \to^*_{\text{need}} s$ and $\mathsf{normal}(s)$ then there exists a tight derivation $\Phi \triangleright_{\text{need}} \vdash^{(|d|_{\mathtt{m}},|d|_{\mathtt{e}})} t : [\mathsf{normal}]$.*

CbNeed Model. The interpretation $[\![t]\!]^{\text{CbNeed}}_{\vec{x}}$ with respect to the CbNeed system is defined as the set (where $\vec{x} = (x_1, \ldots, x_n)$ is a list of variables suitable for t):

$$\{((M_1, \ldots, M_n), N) \mid \exists \Phi \triangleright_{\text{need}} x_1 : M_1, \ldots, x_n : M_n \vdash^{(m,e)} t : N \text{ and } N \neq \mathbf{0}\}\,.$$

Note that the right multi type is required to be non-empty. The *invariance* and the *adequacy* of $[\![t]\!]^{\text{CbNeed}}_{\vec{x}}$ with respect to CbNeed evaluation are obtained exactly as for the CbN and CbV cases.

7 A New Fundamental Theorem for Call-by-Need

CbNeed Erases Wisely. In the literature, *the* theorem about CbNeed is the fact that it is operationally equivalent to CbN. This result was first proven independently by two groups, Maraist, Odersky, and Wadler [45], and Ariola and Felleisen [10], in the nineties, using heavy rewriting techniques.

Recently, Kesner gave a much simpler proof via CbN multi types [37]. She uses multi types to first show termination equivalence of CbN and CbNeed, from which she then infers operational equivalence. Termination equivalence means that a given term terminates in CbN if and only if terminates in CbNeed, and it is a consequence of our slogan that *CbN and CbNeed both erase wisely.*

With our terminology and notations, Kesner's result takes the following form.

Theorem 7 (Kesner [37]). *Let t be a closed term.*

1. Correctness*: if $\Phi \triangleright_{\text{cbn}} \vdash^{(m,e)} t : L$ then there exists s such that $d\colon t \to^*_{\text{need}} s$, $\mathsf{normal}(s)$, $|d|_{\mathtt{m}} \leq m$ and $|d|_{\mathtt{e}} \leq e$.*
2. Completeness*: if $d\colon t \to^*_{\text{need}} s$ and $\mathsf{normal}(s)$ then there is $\Phi \triangleright_{\text{cbn}} \vdash^{(m,e)} t : \mathsf{normal}$.*

Note that, with respect to the other similar theorems in this paper, the result does not cover tight derivations and it does not provide exact bounds. In fact, the CbN system *cannot* provide exact bounds for CbNeed, because it does provide them for CbN evaluation, that in general is slower than CbNeed. Consider for instance the term t in Example 1 and its CbN tight derivation in Example 2: the derivation provides indices $(5, 5)$ for t (and so t evaluates in 10 CbN steps), but t evaluates in 8 CbNeed steps. Closing such a gap is the main motivation behind this paper, achieved by the CbNeed multi type system in Sect. 6.

CbNeed Duplicates Wisely. Curiously, in the literature there are no dual results showing that CbNeed duplicates as wisely as CbV. One of the reasons is that it is a theorem that does not admit a simple formulation such as operational or termination equivalence, because CbNeed and CbV are not in such relationships. Morally, this is subsumed by the logical interpretation according to which CbNeed corresponds to an affine variant of the linear logic representation of CbV. Yet, it would be nice to have a precise, formal statement establishing that *CbNeed duplicates as wisely as CbV*—we provide it here.

Our result is that the CbV multi type system is correct with respect to CbNeed evaluation. In particular, the indices (m, e) provided by a CbV type derivation provide bounds for CbNeed evaluation lengths. Two important remarks before we proceed with the formal statement:

- *Bounds are not exact*: the indices of a CbV derivation do not generally provide exacts bounds for CbNeed, not even in the case of tight derivations. The reason is that CbNeed does not evaluate unneeded subterms (*i.e.* those typed with $\mathbf{0}$), while CbV does. Consider again the term t of Example 1, for instance, whose CbV tight derivation has indices $(5, 5)$ (and so t evaluates in 10 CbV steps) but it CbNeed evaluates in 8 steps.
- *Completeness cannot hold*: we prove correctness but not completeness simply because the CbV system is not complete with respect to CbNeed evaluation. Consider for instance $(\lambda x.I)\Omega$: it is CbV untypable by Theorem 4, because it is CbV divergent, and yet it is CbNeed normalisable.

CbV Correctness with Respect to CbNeed. Pleasantly, our presentations of CbV and CbNeed make the proof of the result straightforward. It is enough to observe that, since we do not consider garbage collection and we adopt a non-deterministic formulation of CbV, CbNeed is a subsystem of CbV. Formally, if $t \to_{\text{need}} s$ then $t \to_{\text{cbv}} s$, as it is easily seen from the definitions (CbNeed reduces only *some* subterms of applications and ES, while CbV reduces *all* such subterms). The result is then a corollary of the correctness theorem for CbV.

Corollary 1 (CbV correctness wrt CbNeed). *Let t be a closed term and $\Phi \triangleright_{\text{cbv}} \ \vdash^{(m,e)} t : M$ be a derivation. Then there exists s such that $d \colon t \to^*_{\text{need}} s$ and $\mathsf{normal}(s)$, with $|d|_{\mathtt{m}} \leq m$ and $|d|_{\mathtt{e}} \leq e$.*

Since the CbNeed system provides exact bounds (Theorem 5), we obtain that CbNeed duplicates as wisely as CbV, when the comparison makes sense, that is, on CbV normalisable terms.

Corollary 2 (CbNeed duplicates as wisely as CbV). *Let $d \colon t \to^*_{\text{cbv}} u$ with $\mathsf{normal}_{\text{cbv}}(u)$. Then there is $d' : t \to^*_{\text{need}} s$ with $\mathsf{normal}(s)$ and $|d'|_{\mathtt{m}} \leq |d|_{\mathtt{m}}$ and $|d'|_{\mathtt{e}} \leq |d|_{\mathtt{e}}$.*

8 Conclusions

Contributions. This paper introduces a multi type system for CbNeed evaluation, carefully blending ingredients from multi type systems for CbN and CbV evaluation in the literature. Notably, it is the first type system whose minimal derivations—explicitly characterised—provide exact bounds for evaluation lengths. It also characterises CbNeed termination, and thus its judgements provide an adequate relational semantics, which is the first one precisely reflecting CbNeed evaluation.

The technical development is simple, and uniform with respect to those of CbN and CbV multi type systems. The typing rules count evaluation steps following *exactly* the same schema of the CbN and CbV rules. The proofs of correctness and completeness also follow *exactly* the same structure.

A further side contribution of the paper is a new fundamental result of CbNeed, formally stating that it duplicates as wisely as CbV. More precisely, the CbV multi type system is (quantitatively) correct with respect to CbNeed evaluation. Pleasantly, our presentations of CbV and CbNeed provide the result for free. This result dualises the other fundamental theorem stating that CbNeed erases as wisely as CbN, usually formulated as termination equivalence, and recently re-proved by Kesner using CbN multi types [37].

Future Work. Recently, Barenbaum et al. extended CbNeed to strong evaluation [12], and it is natural to try to extend our type system as well. The definition of the system, in particular the extension of *tight* derivations to that setting, seems however far from being evident. Barembaum, Bonelli, and Mohamed also apply CbN multi types to a CbNeed calculus extended with pattern matching and fixpoints [13], that might be interesting to refine along the lines of our work.

An orthogonal direction is the study of the denotational models of CbNeed. It would be interesting to have a categorical semantics of CbNeed, as well as a categorical way of discriminating our quantitative precise model from the quantitatively lax one given by CbN multi types. It would also be interesting to obtain game semantics of CbNeed, hopefully satisfying a strong correspondence with our multi types in the style of what happens in CbN [33,34,53,48].

A further, unconventional direction is to dualise the inception of the CbNeed type system trying to mix silly duplication from CbN and silly erasure from CbV, obtaining—presumably—a multi types system measuring a perpetual strategy.

Acknowledgements. This work has been partially funded by the ANR JCJC grant COCA HOLA (ANR-16-CE40-004-01) and by the EPSRC grant EP/R029121/1 “Typed Lambda-Calculi with Sharing and Unsharing”.

References

1. Accattoli, B.: An abstract factorization theorem for explicit substitutions. In: RTA'12. LIPIcs, vol. 15, pp. 6–21 (2012)
2. Accattoli, B.: Proof nets and the Linear Substitution Calculus. In: ICTAC 2018. pp. 37–61 (2018)
3. Accattoli, B., Barenbaum, P., Mazza, D.: Distilling abstract machines. In: ICFP 2014. pp. 363–376 (2014)

4. Accattoli, B., Barras, B.: Environments and the complexity of abstract machines. In: PPDP 2017. pp. 4–16 (2017)
5. Accattoli, B., Barras, B.: The Negligible and Yet Subtle Cost of Pattern Matching. In: APLAS 2017. pp. 426–447 (2017)
6. Accattoli, B., Bonelli, E., Kesner, D., Lombardi, C.: A nonstandard standardization theorem. In: POPL 2014. pp. 659–670. ACM (2014)
7. Accattoli, B., Graham-Lengrand, S., Kesner, D.: Tight typings and split bounds. PACMPL **2**(ICFP), 94:1–94:30 (2018)
8. Accattoli, B., Guerrieri, G.: Types of Fireballs. In: Programming Languages and Systems - 16th Asian Symposium, APLAS 2018. Lecture Notes in Computer Science, vol. 11275, pp. 45–66 (2018)
9. Accattoli, B., Sacerdoti Coen, C.: On the value of variables. In: WoLLIC 2014. pp. 36–50 (2014)
10. Ariola, Z.M., Felleisen, M.: The call-by-need lambda calculus. J. Funct. Program. **7**(3), 265–301 (1997)
11. Ariola, Z.M., Herbelin, H., Saurin, A.: Classical call-by-need and duality. In: TLCA. pp. 27–44 (2011)
12. Balabonski, T., Barenbaum, P., Bonelli, E., Kesner, D.: Foundations of strong call by need. PACMPL **1**(ICFP), 20:1–20:29 (2017)
13. Barenbaum, P., Bonelli, E., Mohamed, K.: Pattern matching and fixed points: Resource types and strong call-by-need: Extended abstract. In: PPDP 2018. pp. 6:1–6:12 (2018). https://doi.org/10.1145/3236950.3236972
14. Barras, B.: Auto-validation d'un système de preuves avec familles inductives. Ph.D. thesis, Université Paris 7 (1999)
15. Bernadet, A., Graham-Lengrand, S.: Non-idempotent intersection types and strong normalisation. Logical Methods in Computer Science **9**(4) (2013)
16. Bucciarelli, A., Ehrhard, T., Manzonetto, G.: A relational semantics for parallelism and non-determinism in a functional setting. Annals of Pure and Applied Logic **163**(7), 918–934 (2012)
17. Bucciarelli, A., Kesner, D., Rocca, S.R.D.: Inhabitation for non-idempotent intersection types. Logical Methods in Computer Science **14**(3) (2018). https://doi.org/10.23638/LMCS-14(3:7)2018, `https://doi.org/10.23638/LMCS-14(3:7)2018`
18. Bucciarelli, A., Kesner, D., Ventura, D.: Non-idempotent intersection types for the lambda-calculus. Logic Journal of the IGPL **25**(4), 431–464 (2017)
19. Carraro, A., Guerrieri, G.: A semantical and operational account of call-by-value solvability. In: Muscholl, A. (ed.) FoSSaCS 2014. Lecture Notes in Computer Science, vol. 8412, pp. 103–118. Springer (2014)
20. de Carvalho, D.: Execution time of λ-terms via denotational semantics and intersection types. Mathematical Structures in Computer Science **28**(7), 1169–1203 (2018). https://doi.org/10.1017/S0960129516000396
21. de Carvalho, D., Tortora de Falco, L.: A semantic account of strong normalization in linear logic. Information and Computation **248**, 104–129 (2016)
22. de Carvalho, D., Pagani, M., Tortora de Falco, L.: A semantic measure of the execution time in linear logic. Theoretical Computer Science **412**(20), 1884–1902 (2011)
23. Chang, S., Felleisen, M.: The call-by-need lambda calculus, revisited. In: ESOP. pp. 128–147 (2012)
24. Coppo, M., Dezani-Ciancaglini, M.: A new type assignment for λ-terms. Arch. Math. Log. **19**(1), 139–156 (1978)

25. Coppo, M., Dezani-Ciancaglini, M.: An extension of the basic functionality theory for the λ-calculus. Notre Dame Journal of Formal Logic **21**(4), 685–693 (1980)
26. Danvy, O., Zerny, I.: A synthetic operational account of call-by-need evaluation. In: PPDP. pp. 97–108 (2013)
27. Díaz-Caro, A., Manzonetto, G., Pagani, M.: Call-by-value non-determinism in a linear logic type discipline. In: Artëmov, S.N., Nerode, A. (eds.) LFCS 2013. Lecture Notes in Computer Science, vol. 7734, pp. 164–178. Springer (2013)
28. Downen, P., Maurer, L., Ariola, Z.M., Varacca, D.: Continuations, processes, and sharing. In: PPDP 2014. pp. 69–80 (2014). https://doi.org/10.1145/2643135.2643155
29. Ehrhard, T.: Collapsing non-idempotent intersection types. In: Cégielski, P., Durand, A. (eds.) CSL'12. LIPIcs, vol. 16, pp. 259–273. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012)
30. Ehrhard, T., Guerrieri, G.: The bang calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In: PPDP 2016. pp. 174–187. ACM (2016)
31. Garcia, R., Lumsdaine, A., Sabry, A.: Lazy evaluation and delimited control. In: POPL. pp. 153–164 (2009)
32. Gardner, P.: Discovering needed reductions using type theory. In: Hagiya, M., Mitchell, J.C. (eds.) TACS '94. Lecture Notes in Computer Science, vol. 789, pp. 555–574. Springer (1994)
33. Gianantonio, P.D., Honsell, F., Lenisa, M.: A type assignment system for game semantics. Theor. Comput. Sci. **398**(1-3), 150–169 (2008). https://doi.org/10.1016/j.tcs.2008.01.023
34. Gianantonio, P.D., Lenisa, M.: Innocent game semantics via intersection type assignment systems. In: CSL 2013. pp. 231–247 (2013). https://doi.org/10.4230/LIPIcs.CSL.2013.231
35. Girard, J.Y.: Linear logic. Theoretical Computer Science **50**, 1–102 (1987)
36. Guerrieri, G.: Towards a semantic measure of the execution time in call-by-value lambda-calculus. Tech. rep. (2018), submitted to ITRS 2018
37. Kesner, D.: Reasoning about call-by-need by means of types. In: FOSSACS 2016. pp. 424–441 (2016). https://doi.org/10.1007/978-3-662-49630-5_25
38. Kesner, D., Vial, P.: Types as resources for classical natural deduction. In: Miller, D. (ed.) FSCD 2017. LIPIcs, vol. 84, pp. 24:1–24:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017)
39. Kesner, D., Viso, A., Ríos, A.: Call-by-need, neededness and all that. In: Baier, C., Lago, U.D. (eds.) FoSSaCS'18. Lecture Notes in Computer Science, vol. 10803, pp. 241–257. Springer (2018)
40. Kfoury, A.J.: A linearization of the lambda-calculus and consequences. Journal of Logic and Computation **10**(3), 411–436 (2000)
41. Krivine, J.L.: λ-calcul, types et modèles. Masson (1990)
42. Kutzner, A., Schmidt-Schauß, M.: A non-deterministic call-by-need lambda calculus. In: ICFP 1998. pp. 324–335 (1998). https://doi.org/10.1145/289423.289462
43. Launchbury, J.: A natural semantics for lazy evaluation. In: POPL. pp. 144–154 (1993)
44. Maraist, J., Odersky, M., Turner, D.N., Wadler, P.: Call-by-name, call-by-value, call-by-need and the linear lambda calculus. Theor. Comput. Sci. **228**(1-2), 175–210 (1999)
45. Maraist, J., Odersky, M., Wadler, P.: The call-by-need lambda calculus. J. Funct. Program. **8**(3), 275–317 (1998)
46. Mazza, D., Pellissier, L., Vial, P.: Polyadic approximations, fibrations and intersection types. PACMPL **2**(POPL), 6:1–6:28 (2018)

47. Neergaard, P.M., Mairson, H.G.: Types, potency, and idempotency: why nonlinearity and amnesia make a type system work. In: Okasaki, C., Fisher, K. (eds.) ICFP 2004. pp. 138–149. ACM (2004)
48. Ong, C.L.: Quantitative semantics of the lambda calculus: Some generalisations of the relational model. In: Ouaknine, J. (ed.) LICS 2017. pp. 1–12. IEEE Computer Society (2017)
49. Paolini, L., Piccolo, M., Ronchi Della Rocca, S.: Essential and relational models. Mathematical Structures in Computer Science **27**(5), 626–650 (2017)
50. Pédrot, P., Saurin, A.: Classical by-need. In: ESOP 2016. pp. 616–643. Springer (2016)
51. Pottinger, G.: A type assignment for the strongly normalizable λ-terms. In: Seldin, J., Hindley, J. (eds.) To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, pp. 561–578. Academic Press (1980)
52. Sestoft, P.: Deriving a lazy abstract machine. J. Funct. Program. **7**(3), 231–264 (1997), `http://journals.cambridge.org/action/displayAbstract?aid=44087`
53. Tsukada, T., Ong, C.L.: Plays as resource terms via non-idempotent intersection types. In: LICS '16. pp. 237–246 (2016). https://doi.org/10.1145/2933575.2934553
54. Wadsworth, C.P.: Semantics and pragmatics of the lambda-calculus. PhD Thesis, Oxford (1971), chapter 4

Proof Appendix

A Closed $\boldsymbol{\lambda}$-Calculi (Sect. 2)

Remark 1. Let t be a term. According to definition of the predicate normal, $\mathsf{normal}(t)$ if and only if $t \coloneqq S\langle\lambda x.s\rangle$ for some term s and substitution context S.

Remark 2. Let t be a term. According to definition of the predicate $\mathsf{normal}_{\mathrm{cbv}}$, $\mathsf{normal}_{\mathrm{cbv}}(t)$ if and only if $t \coloneqq (\lambda x.s)[y_1{\leftarrow}s_1]\dots[y_n{\leftarrow}s_n]$ for some $n \in \mathbb{N}$ and terms $s, s_1, \dots, s_n$ such that $\mathsf{normal}_{\mathrm{cbv}}(s_i)$ for all $1 \leq i \leq n$. In particular, if $\mathsf{normal}_{\mathrm{cbv}}(t)$ then $\mathsf{normal}(t)$.

Remark 3. Every term can be written in a unique way as $C\langle t\rangle$ for some CbN context C, where t is either a variable or an abstraction.

Remark 4. Every CbN context is a CbNeed context and a CbV context.

Proposition 14 (Syntactic characterisation of closed normal forms). *Let t be a closed term.*
1. CbN and CbNeed*: For* $\mathsf{r} \in \{\mathrm{cbn}, \mathrm{need}\}$*, t is* r*-normal if and only if* $\mathsf{normal}(t)$*.*
2. CbV*: t is* cbv*-normal if and only if* $\mathsf{normal}_{\mathrm{cbv}}(t)$*.*

Proof.
1. First, we prove that a closed term t is CbN normal if and only if $\mathsf{normal}(t)$.
   $\Rightarrow$: Let t be a closed and CbN normal term. We prove by induction on t that $\mathsf{normal}(t)$. Cases:
   - *Variable*, *i.e.* $t \coloneqq x$: it is impossible because t is closed by hypothesis.
   - *Abstraction*, *i.e.* $t \coloneqq \lambda y.s$: then, $\mathsf{normal}(t)$ according to the definition of the predicate normal.
   - *Application*, *i.e.* $t \coloneqq su$: this case is impossible, we prove it by contradiction. Suppose $t \coloneqq su$, then s would be closed and CbN normal (as t is so), and hence $\mathsf{normal}(s)$ by *i.h.* According to Remark 1, $s = S\langle\lambda y.r\rangle$ and so $t = S\langle\lambda y.r\rangle u$, which is impossible because t would be a $\mathtt{m}$-redex.
   - *Explicit substitution*, *i.e.* $t \coloneqq s[x{\leftarrow}u]$: then, s is CbN normal. There are four subcases:
     - s is closed, then $\mathsf{normal}(s)$ by *i.h.*, and hence $\mathsf{normal}(t)$;
     - $s \coloneqq S\langle\lambda y.r\rangle$, then $\mathsf{normal}(s)$ by Remark 1, and hence $\mathsf{normal}(t)$;
     - $s \coloneqq C\langle y\rangle$: this is impossible because otherwise (since t is closed) $t = C'\langle D\langle\langle y\rangle\rangle[y{\leftarrow}r]\rangle$ which is not $\mathsf{e}_{\mathrm{cbn}}$-normal;
     - s is none of the above, then by Remark 3, $s \coloneqq C\langle\lambda y.r\rangle$ for some CbN context C that is not a substitution context: this case is impossible because otherwise, by Remark 3, $s \coloneqq D\langle S\langle\lambda y.r\rangle q\rangle$ for some CbN context D, which is not $\mathtt{m}_{\mathrm{cbn}}$-normal.

$\Leftarrow$: We prove by induction on the definition of $\mathsf{normal}(t)$ a stronger statement: for any term t, if $\mathsf{normal}(t)$ then t is CbN normal (we dropped the hypothesis that t is closed). There are only two cases:
  - *Abstraction*, *i.e.* $t := \lambda x.s$, then t is CbN normal because $\to_{\text{cbn}}$ does not reduce under abstractions.
  - *Explicit substitution*, *i.e.* $t := s[x{\leftarrow}u]$ with $\mathsf{normal}(s)$, then $s = S\langle\lambda y.r\rangle$ for some substitution context S according to Remark 1, thus $t = S'\langle\lambda y.r\rangle$ for the substitution context $S' = S[x{\leftarrow}u]$. By *i.h.*, s is CbN normal. Hence, t is $\mathtt{m}_{\text{cbn}}$-normal. Moreover, $t \neq C\langle x\rangle$ for any CbN context C, so t is also e_{cbn}-normal and hence CbN normal.

2. We prove that a closed term t is CbNeed normal if and only if $\mathsf{normal}(t)$.

$\Rightarrow$: Let t be a closed and CbNeed normal term. We prove by induction on t that $\mathsf{normal}(t)$. Cases:
  - *Variable*, *i.e.* $t := x$: it is impossible because t is closed by hypothesis.
  - *Abstraction*, *i.e.* $t := \lambda y.s$: then, $\mathsf{normal}(t)$ according to the definition of the predicate normal.
  - *Application*, *i.e.* $t := su$: this case is impossible, we prove it by contradiction. Suppose $t := su$, then s would be closed and CbNeed normal (as t is so), and hence $\mathsf{normal}(s)$ by *i.h.* According to Remark 1, $s = S\langle\lambda y.r\rangle$ and so $t = S\langle\lambda y.r\rangle u$, which is impossible because t would be a $\mathtt{m}$-redex.
  - *Explicit substitution*, *i.e.* $t := s[x{\leftarrow}u]$: then, s is CbNeed normal. There are four subcases:
    - s is closed, then $\mathsf{normal}(s)$ by *i.h.*, and hence $\mathsf{normal}(t)$;
    - $s := S\langle\lambda y.r\rangle$, then $\mathsf{normal}(s)$ by Remark 1, and hence $\mathsf{normal}(t)$;
    - $s := E\langle y\rangle$: this is impossible because otherwise (since t is closed) $t = E'\langle E'\langle\langle y\rangle\rangle[y{\leftarrow}r]\rangle$ for some $r := S\langle v\rangle$ (by *i.h.* and Remark 1, since r is CbNeed normal), and so t would not be e_{need}-normal;
    - s is none of the above, then by Remark 3, $s := C\langle\lambda y.r\rangle$ for some CbN (and hence CbNeed, by Remark 4) context C that is not a substitution context: this case is impossible because otherwise, according to Remark 3, $s := D\langle S\langle\lambda y.r\rangle q\rangle$ for some CbN context D, which is not $\mathtt{m}_{\text{need}}$-normal (since D is a CbNeed context by Remark 4).

$\Leftarrow$: We prove by induction on the definition of $\mathsf{normal}(t)$ a stronger statement: for any term t, if $\mathsf{normal}(t)$ then t is CbNeed normal (we dropped the hypothesis that t is closed). There are only two cases:
  - *Abstraction*, *i.e.* $t := \lambda x.s$, then t is CbNeed normal because $\to_{\text{need}}$ does not reduce under abstractions.
  - *Explicit substitution*, *i.e.* $t := s[x{\leftarrow}u]$ with $\mathsf{normal}(s)$, then $s = S\langle\lambda y.r\rangle$ for some substitution context S according to Remark 1, thus $t = S'\langle\lambda y.r\rangle$ for the substitution context $S' = S[x{\leftarrow}u]$. By *i.h.*, s is CbNeed normal. Hence, t is $\mathtt{m}_{\text{need}}$-normal. Moreover, $t \neq E\langle x\rangle$ for any CbNeed context E, so t is also e_{need}-normal and hence CbNeed normal.

3. We prove that a closed term t is CbV normal if and only if $\mathsf{normal}_{\mathrm{cbv}}(t)$.

   $\Rightarrow$: Let t be a closed and CbV normal term. We prove by induction on t that $\mathsf{normal}_{\mathrm{cbv}}(t)$. Cases:
   - *Variable*, *i.e.* $t := x$: it is impossible because t is closed by hypothesis.
   - *Abstraction*, *i.e.* $t := \lambda y.s$: then, $\mathsf{normal}_{\mathrm{cbv}}(t)$ according to the definition of the predicate $\mathsf{normal}_{\mathrm{cbv}}$.
   - *Application*, *i.e.* $t := su$: this case is impossible, we prove it by contradiction. Suppose $t := su$, then s would be closed and CbV normal, and hence $\mathsf{normal}_{\mathrm{cbv}}(s)$ by *i.h.* According to Remark 2, $s = S\langle\lambda y.r\rangle$ and so $t = S\langle\lambda y.r\rangle u$, which is impossible because t would be a $\mathtt{m}$-redex.
   - *Explicit substitution*, *i.e.* $t := s[x{\leftarrow}u]$: then, s and u are CbV normal, and u is closed (as t is so), thus $\mathsf{normal}_{\mathrm{cbv}}(u)$ by *i.h.* There are four subcases:
     - s is closed, then $\mathsf{normal}_{\mathrm{cbv}}(s)$ by *i.h.*, and hence $\mathsf{normal}_{\mathrm{cbv}}(t)$;
     - $s := (\lambda y.r)[y_1{\leftarrow}s_1]\dots[y_n{\leftarrow}s_n]$, then all r_i's are CbV normal (as s is so); by *i.h.*, $\mathsf{normal}_{\mathrm{cbv}}(r_i)$ for all $1 \leq i \leq n$, thus $\mathsf{normal}_{\mathrm{cbv}}(s)$ (since $\mathsf{normal}_{\mathrm{cbv}}(\lambda y.r)$) and hence $\mathsf{normal}_{\mathrm{cbv}}(t)$;
     - $s := V\langle y\rangle$: this is impossible because otherwise (since t is closed) $t = V'\langle V''\langle\langle y\rangle\rangle[y{\leftarrow}r]\rangle$ for some $r := S\langle v\rangle$ (by *i.h.* and Remark 2, since r is CbV normal), and so t would not be $\mathsf{e}_{\mathrm{cbv}}$-normal;
     - s is none of the above, then by Remark 3, $s := C\langle\lambda y.r\rangle$ for some CbN (and hence CbV, by Remark 4) context C that is not a substitution context: this case is impossible because otherwise, by Remark 3, $s := D\langle S\langle\lambda y.r\rangle q\rangle$ for some CbN context D, which is not $\mathtt{m}_{\mathrm{cbv}}$-normal (since D is a CbV context by Remark 4).

   $\Leftarrow$: We prove by induction on the definition of $\mathsf{normal}_{\mathrm{cbv}}(t)$ a stronger statement: for any term t, if $\mathsf{normal}_{\mathrm{cbv}}(t)$ then t is CbV normal (we dropped the hypothesis that t is closed). There are only two cases:
   - *Abstraction*, *i.e.* $t := \lambda x.s$, then t is CbV normal because $\rightarrow_{\mathrm{cbv}}$ does not reduce under abstractions.
   - *Explicit substitution*, *i.e.* $t := s[x{\leftarrow}u]$ with $\mathsf{normal}_{\mathrm{cbv}}(s)$ and $\mathsf{normal}_{\mathrm{cbv}}(u)$, then $t = (\lambda y.r)[y_1{\leftarrow}s_1]\dots[y_n{\leftarrow}s_n][x{\leftarrow}u]$ where $\mathsf{normal}_{\mathrm{cbv}}(s_i)$ for all $1 \leq i \leq n$, by Remark 2 applied to s. By *i.h.*, $u, s_1, \dots, s_n$ and s are CbV normal. Hence, t is $\mathtt{m}_{\mathrm{cbv}}$-normal. Moreover, $t \neq V\langle x\rangle$ for any CbV context V, so t is also $\mathsf{e}_{\mathrm{cbv}}$-normal and hence CbV normal. □

B Types by Name (Sect. 4)

B.1 CbN Correctness

In order to prove subject reduction, we have to first show that typability is preserved by linear substitutions, via a dedicated lemma. We also need the following splitting property of multi-sets, whose proof is omitted because straightforward.

Lemma 8 (Splitting multi-sets with respect to derivations). *Let t be a term, $\Phi \triangleright_{CbN} \ \Gamma \vdash^{(m,e)} t : M$ a derivation, and $M = N \uplus O$ a splitting. Then there exist two derivations*
- $\Phi_N \triangleright_{CbN} \ \Gamma_N \vdash^{(m_N, e_N)} t : N$, *and*
- $\Phi_O \triangleright_{CbN} \ \Gamma_O \vdash^{(m_O, e_O)} t : O$

such that
- Type contexts*: $\Gamma = \Gamma_N \uplus \Gamma_O$,*
- Indices*: $m = m_N + m_O$ and $e = e_N + e_O$.*

Lemma 9 (CbN linear substitution). *If $\Phi \triangleright_{\text{cbn}} \ \Gamma, x : M \vdash^{(m,e)} C\langle\langle x\rangle\rangle : L$ then there is a splitting $M = [L'] \uplus N$ such that for every derivation $\Psi \triangleright_{\text{cbn}} \ \Pi \vdash^{(m',e')} t : L'$ there is a derivation $\Theta \triangleright_{\text{cbn}} \ \Gamma \uplus \Pi, x : N \vdash^{(m+m', e+e'-1)} C\langle\langle t\rangle\rangle : L$.*

Proof. By induction on C. Cases:
- *Empty context*, *i.e.* $C = \langle\cdot\rangle$. The typing derivation Φ is simply

$$\frac{}{x : [L] \vdash^{(0,1)} x : L}\ \mathsf{ax}$$

  and Γ is empty. Then $M = [L]$ and so N is empty. The statement then holds with respect to $\Phi_{C\langle\langle t\rangle\rangle} := \Psi$, because $m = 0$ and $e = 1$.
- *Left on an application*, *i.e.* $C = Ds$. The last rule of Φ can only be app, and so Φ has the form:

$$\frac{x\colon M_\Delta; \Delta \vdash^{(m_\Delta, e_\Delta)} D\langle\langle x\rangle\rangle : N \multimap L \quad x\colon M_\Sigma; \Sigma \vdash^{(m_\Sigma, e_\Sigma)} s : N}{x\colon (M_\Delta \biguplus M_\Sigma); (\Delta \biguplus \Sigma) \vdash^{(m_\Delta + m_\Sigma + 1, e_\Delta + e_\Sigma)} D\langle\langle x\rangle\rangle s : L}\ \mathsf{app}$$

  where $\Gamma = \Delta \biguplus \Sigma$, $\Delta(x) = \Sigma(x) = \mathbf{0}$, $M_\Delta \uplus M_\Sigma = M$, $m = m_\Delta + m_\Sigma + 1$, and $e = e_\Delta + e_\Sigma$.
  By *i.h.*, there exists a splitting $M_\Delta = [L'] \uplus O$ such that for every derivation $\Psi \triangleright_{\text{CbN}} \ \Pi \vdash^{(m',e')} t : L'$ there exists a derivation

$$\Phi_{D\langle\langle t\rangle\rangle} \triangleright_{\text{CbN}} \ x : O; \Delta \biguplus \Pi \vdash^{(m_\Delta + m', e_\Delta + e' - 1)} D\langle\langle t\rangle\rangle : N \multimap L$$

  By applying an app rule we obtain:

$$\frac{x : O; \Delta \uplus \Pi \vdash^{(m_\Delta + m', e_\Delta + e' - 1)} D\langle\langle t\rangle\rangle : N \multimap L \quad x\colon M_\Sigma; \Sigma \vdash^{(m_\Sigma, e_\Sigma)} s : N}{x\colon (O \uplus M_\Sigma); (\Delta \uplus \Pi \uplus \Sigma) \vdash^{(m_\Delta + m' + m_\Sigma + 1, e_\Delta + e' + e_\Sigma - 1)} D\langle\langle x\rangle\rangle s : L}\ \mathsf{app}$$

  Now, by defining $N := O \uplus M_\Sigma$, we obtain $M = M_\Delta \uplus M_\Sigma = [L'] \uplus O \uplus M_\Sigma = [L'] \uplus N$. Therefore by applying the equalities on the type context the last obtained judgement is in fact:

$$x\colon N; (\Gamma \uplus \Pi) \vdash^{(m_\Delta + m' + m_\Sigma + 1, e_\Delta + e' + e_\Sigma - 1)} D\langle\langle x\rangle\rangle s : L$$

  and by applying those on the indices we obtain:

$$x\colon N; (\Gamma \uplus \Pi) \vdash^{(m + m', e + e' - 1)} D\langle\langle x\rangle\rangle s : L$$

  as required.

– *Left of a substitution, i.e.* $C = D[y{\leftarrow}s]$. Note that $x \neq y$, because the hypothesis $C\langle\langle x\rangle\rangle$ implies that C does not capture x.
The last rule of Φ can only be ES, and so Φ has the form:

$$\frac{x\colon M_\Delta; y\colon M'; \Delta \vdash^{(m_\Delta,e_\Delta)} D\langle\langle x\rangle\rangle : L \quad x\colon M_\Sigma; \Sigma \vdash^{(m_\Sigma,e_\Sigma)} s : M'}{x\colon (M_\Delta \uplus M_\Sigma); (\Delta \uplus \Sigma) \vdash^{(m_\Delta+m_\Sigma+1, e_\Delta+e_\Sigma)} D\langle\langle x\rangle\rangle[y{\leftarrow}s] : L}\ \texttt{ES}$$

where $\Gamma = \Delta \uplus \Sigma$, $\Delta(x) = \Sigma(x) = \mathbf{0}$, $M_\Delta \uplus M_\Sigma = M$, $m = m_\Delta + m_\Sigma + 1$, and $e = e_\Delta + e_\Sigma$.
By *i.h.*, there exists a splitting $M_\Delta = [L'] \uplus O$ such that for every derivation $\Psi \triangleright_{\text{CbN}}\ \Pi \vdash^{(m',e')} t : L'$ there exists a derivation

$$\Phi_{D\langle\langle t\rangle\rangle} \triangleright_{\text{CbN}}\ x : O; y\colon M'; \Delta \uplus \Pi \vdash^{(m_\Delta+m',e_\Delta+e'-1)} D\langle\langle t\rangle\rangle : L$$

Note that by Lemma 1 and the fact that we are working up to α-equivalence, we can prove that $y \notin \texttt{dom}(\Pi)$. By applying a rule ES we obtain

$$\frac{x : O; y\colon M'; \Delta \uplus \Pi \vdash^{(m_\Delta+m',e_\Delta+e'-1)} D\langle\langle t\rangle\rangle : L \quad x\colon M_\Sigma; \Sigma \vdash^{(m_\Sigma,e_\Sigma)} s : M'}{x : O \uplus M_\Sigma; \Delta \uplus \Pi \uplus \Sigma \vdash^{(m_\Delta+m'+m_\Sigma+1, e_\Delta+e'+e_\Sigma-1)} D\langle\langle x\rangle\rangle[y{\leftarrow}s] : L}\ \texttt{ES}$$

Now, by defining $N := O \uplus M_\Sigma$, we obtain $M = M_\Delta \uplus M_\Sigma = [L'] \uplus O \uplus M_\Sigma = [L'] \uplus N$. Therefore by applying the equalities on the type context the last obtained judgement is in fact:

$$x\colon N; (\Gamma \uplus \Pi) \vdash^{(m_\Delta+m'+m_\Sigma+1, e_\Delta+e'+e_\Sigma-1)} D\langle\langle x\rangle\rangle[y{\leftarrow}s] : L$$

and by applying those on the indices we obtain:

$$x\colon N; (\Gamma \uplus \Pi) \vdash^{(m+m',e+e'-1)} D\langle\langle x\rangle\rangle[y{\leftarrow}s] : L$$

as required. □

Proposition 15 (Quantitative subject reduction for CbN). *Let $\Phi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e)} t : L$ be a derivation.*

1. Multiplicative*: if $t \to_{\mathtt{m}_{\text{cbn}}} s$ then $m \geq 1$ and there exists a derivation $\Psi \triangleright_{\text{cbn}} \Gamma \vdash^{(m-1,e)} s : L$.*
2. Exponential*: if $t \to_{\mathtt{e}_{\text{cbn}}} s$ then $e \geq 1$ and there exists a derivation $\Psi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e-1)} s : L$.*

Proof.
1. By induction on $t \to_{\mathtt{m}} s$. Cases:
   – *Step at top level, i.e.* $t = S\langle \lambda x.u\rangle r \to_{\mathtt{m}} S\langle u[x{\leftarrow}r]\rangle = s$. This case is itself by induction on S. Two sub-cases:

- *Empty substitution context, i.e.* $S = \langle\cdot\rangle$. By construction the derivation Φ is of the form:

$$\dfrac{\dfrac{x : M; \Gamma_u \vdash^{(m_u,e_u)} u : L}{\Gamma_u \vdash^{(m_u,e_u)} \lambda x.u : M \to L}\,\texttt{fun} \qquad \Gamma_r \vdash^{(m_r,e_r)} r : M}{\Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r+1,e_u+e_r)} (\lambda x.u)r : L}\,\texttt{app}$$

With $\Gamma = \Gamma_u \uplus \Gamma_r$, $m = m_u + m_r$, and $e = e_u + e_r$. Note that $m \geq 1$ as required. We construct the following derivation Ψ, verifying the statement:

$$\dfrac{x : M; \Gamma_u \vdash^{(m_u,e_u)} u : L \quad \Gamma_r \vdash^{(m_r,e_r)} r : M}{\Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r,e_u+e_r)} u[x{\leftarrow}r] : L}\,\texttt{ES}$$

- *Non-empty substitution context, i.e.* $S = S'[y{\leftarrow}q]$. Then Φ has the following structure:

$$\dfrac{\dfrac{y : N; \Gamma_u \vdash^{(m_u,e_u)} S'\langle\lambda x.u\rangle : M \to L \quad \Gamma_q \vdash^{(m_q,e_q)} q : N}{\Gamma_u \uplus \Gamma_q \vdash^{(m_u+m_q,e_u+e_q)} S'\langle\lambda x.u\rangle[y{\leftarrow}q] : M \to L}\,\texttt{ES} \qquad \Gamma_r \vdash^{(m_r,e_r)} r : M}{\Gamma_u \uplus \Gamma_q \uplus \Gamma_r \vdash^{(m_u+m_q+m_r+1,e_u+e_q+e_r)} S'\langle\lambda x.u\rangle[y{\leftarrow}q]r : L}\,\texttt{app}$$

With $\Gamma = \Gamma_u \uplus \Gamma_q \uplus \Gamma_r$, $m = m_u + m_q + m_r + 1$, and $e = e_u + e_q + e_r$. Note that $m \geq 1$ as required.
Consider the following derivation, obtained by removing the rule ES:

$$\dfrac{y : N; \Gamma_u \vdash^{(m_u,e_u)} S'\langle\lambda x.u\rangle : M \to L \quad \Gamma_r \vdash^{(m_r,e_r)} r : M}{y : N; \Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r+1,e_u+e_r)} S'\langle\lambda x.u\rangle r : L}\,\texttt{app}$$

By *i.h.*, we obtain a derivation

$$\Theta \triangleright\ y : N; \Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r,e_u+e_r)} S'\langle u[x{\leftarrow}r]\rangle : L$$

Now, we apply a rule ES with respect to y and q, obtaining the following derivation Ψ, satisfying the statement:

$$\dfrac{y : N; \Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r,e_u+e_r)} S'\langle u[x{\leftarrow}r]\rangle : L \quad \Gamma_q \vdash^{(m_q,e_q)} q : N}{\Gamma_u \uplus \Gamma_q \uplus \Gamma_r \vdash^{(m_u+m_q+m_r,e_u+e_q+e_r)} S'\langle u[x{\leftarrow}r]\rangle[y{\leftarrow}q] : L}\,\texttt{ES}$$

– *Contextual closure.* We have $t = C\langle u\rangle \to_{\mathtt{m}} C\langle r\rangle = s$. Cases of C:

- *Left on an application, i.e.* $C = Dq$. The last typing rule in Φ is necessarily app and Φ is of the form

$$\dfrac{\Gamma_u \vdash^{(m_u,e_u)} D\langle u\rangle : M \multimap L \quad \Gamma_q \vdash^{(m_q,e_q)} q : M}{\Gamma_u \uplus \Gamma_q \vdash^{(m_u+m_q+1,e_u+e_q)} D\langle u\rangle q : L}\,\texttt{app}$$

With $\Gamma = \Gamma_u \uplus \Gamma_q$, $m = m_u + m_q + 1$, and $e = e_u + e_q$.

By *i.h.*, $m_u \geq 1$ and there exists a derivation $\Gamma_u \vdash^{(m_u-1,e_u)} D\langle r\rangle : M \multimap L$, thus allowing us to construct Ψ as follows:

$$\dfrac{\Gamma_u \vdash^{(m_u-1,e_u)} D\langle r\rangle : M \multimap L \quad \Gamma_q \vdash^{(m_q,e_q)} q : M}{\Gamma_u \uplus \Gamma_q \vdash^{(m_u+m_q,e_u+e_q)} D\langle r\rangle q : L}\ \mathtt{app}_b$$

Note that $(m_u + m_q, e_u + e_q) = (m-1, e)$.

- Let $C = D[x{\leftarrow}q]$. The last typing rule in Φ is necessarily ES and Φ is of the form

$$\dfrac{\Gamma_u, x : M \vdash^{(m_u,e_u)} D\langle u\rangle : L \quad \Gamma_q \vdash^{(m_q,e_q)} q : M}{\Gamma_u \uplus \Gamma_q \vdash^{(m_u+m_q,e_u+e_q)} D\langle u\rangle[x{\leftarrow}q] : L}\ \text{ES}$$

With $\Gamma = \Gamma_u \uplus \Gamma_q$, $m = m_u + m_q$, and $e = e_u + e_q$.
By *i.h.*, $m_u \geq 1$, and so $m \geq 1$, and there exists a derivation $\Gamma_u, x : M \vdash^{(m_u-1,e_u)} D\langle r\rangle : L$, thus allowing us to construct Ψ as follows:

$$\dfrac{\Gamma_u, x : M \vdash^{(m_u-1,e_u)} D\langle r\rangle : L \quad \Gamma_q \vdash^{(m_q,e_q)} q : M}{\Gamma_u \uplus \Gamma_q \vdash^{(m_u+m_q-1,e_u+e_q)} D\langle u\rangle[x{\leftarrow}q] : L}\ \text{ES}$$

Note that $(m_u + m_q - 1, e_u + e_q) = (m-1, e)$.

2. By induction on $t \to_{\mathsf{e}} s$.
   - *Step at top level*, *i.e.* $t = C\langle\langle x\rangle\rangle[x{\leftarrow}u] \to_{\mathsf{e}} C\langle\langle u\rangle\rangle[x{\leftarrow}u] = s$. The last typing rule in Φ is necessarily ES and Φ is of the form

$$\dfrac{\Phi_{C\langle\langle x\rangle\rangle} \rhd\ \Gamma_C, x : M \vdash^{(m_C,e_C)} C\langle\langle x\rangle\rangle : L \quad \Pi \vdash^{(m',e')} u : M}{\Gamma_C \uplus \Pi \vdash^{(m_C+m_u,e_C+e_u)} C\langle\langle x\rangle\rangle[x{\leftarrow}u] : L}\ \text{ES}$$

With $\Gamma = \Gamma_C \uplus \Pi$, $m = m_C + m'$, and $e = e_C + e'$.
Let $M = [L'] \uplus N$ be the splitting of M given by the linear substitution lemma (Lemma 2) applied to $\Phi_{C\langle\langle x\rangle\rangle}$. By the multi-sets splitting lemma (Lemma 8) there exist two derivations
(a) $\Psi_{L'} \rhd_{\text{CbN}}\ \Pi_{L'} \vdash^{(m'_{L'},e'_{L'})} u : L'$ and
(b) $\Psi_N \rhd_{\text{CbN}}\ \Pi_N \vdash^{(m'_N,e'_N)} u : N$.
such that $\Pi = \Pi_{L'} \uplus \Pi_N$, $m' = m'_{L'} + m'_N$, and $e' = e'_{L'} + e'_N$.
Now, by applying again the linear substitution lemma to $\Phi_{C\langle\langle x\rangle\rangle}$ with respect to $\Psi_{L'}$, we obtain a derivation

$$\Phi_{C\langle\langle u\rangle\rangle} \rhd_{\text{CbN}}\ x : N; \Gamma_C \uplus \Pi_{L'} \vdash^{(m_C+m'_{L'},e_C+e'_{L'}-1)} C\langle\langle u\rangle\rangle : L$$

Then Ψ is built as follows:

$$\dfrac{x : N; \Gamma_C \uplus \Pi_{L'} \vdash^{(m_C+m'_{L'},e_C+e'_{L'}-1)} C\langle\langle u\rangle\rangle : L \quad \Pi_N \vdash^{(m'_N,e'_N)} u : N}{\Gamma_C \uplus \Pi_{L'} \uplus \Pi_N \vdash^{(m_C+m_{L'}+m_N,e_C+e_{L'}+e_N-1)} C\langle\langle u\rangle\rangle[x{\leftarrow}u] : L}\ \text{ES}$$

Now, note that the last judgement is in fact

$$\Gamma_C \uplus \Pi \vdash^{(m_C+m',e_C+e'-1)} C\langle\langle u\rangle\rangle[x{\leftarrow}u] : L$$

which in turn is

$$\Gamma \vdash^{(m,e-1)} C\langle\langle u\rangle\rangle[x{\leftarrow}u] : L$$

as required.

– *Contextual closure.* As in the $\to_{\mathtt{m}}$ case. Note that indeed those cases do not depend on the details of the step itself, but only on the context enclosing it. □

Proposition 16 (normal typing of normal forms for CbN). *Let t be such that* $\mathsf{normal}(t)$*, and $\Phi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e)} t : \mathsf{normal}$ be a derivation. Then Γ is empty, and so Φ is tight, and $m = e = 0$.*

Proof. By induction on the derivation of $\mathsf{normal}(t)$. Cases:

– *Base*, *i.e.* $\mathsf{normal}(\lambda x.t)$. Then Φ can only be:

$$\dfrac{}{\vdash^{(0,0)} \lambda x.t : \mathsf{normal}}\ \mathsf{normal}$$

which satisfies the statement.

– *Inductive*, *i.e.* $\mathsf{normal}(t[x{\leftarrow}s])$ because $\mathsf{normal}(t)$. Then Φ has the following shape.

$$\dfrac{\Phi_t \triangleright\ \Gamma_t; x : M \vdash^{(m_t,e_t)} t : \mathsf{normal} \quad \Phi_s \triangleright\ \Gamma_s \vdash^{(m_s,e_s)} s : M}{\Gamma_t \uplus \Gamma_s \vdash^{(m_t+m_s,e_t+e_s)} t[x{\leftarrow}s] : \mathsf{normal}}\ \mathsf{ES}$$

with $\Gamma = \Gamma_t \uplus \Gamma_s$, $m = m_t + m_s$, and $e = e_t + e_s$. We can apply the *i.h.* to Φ_t, obtaining that $M = \mathbf{0}$, Γ_t is empty, and $m_t = e_t = 0$. Then Φ_s is simply a many rule with 0 premises. Therefore, Γ_s is empty and $m_s = e_s = 0$. Then Γ is empty and $m = e = 0$. □

Theorem 8 (CbN tight correctness). *Let t be a closed term. If $\Phi \triangleright_{\text{cbn}} \vdash^{(m,e)} t : L$ then there is s such that $d \colon t \to^*_{\text{cbn}} s$,* $\mathsf{normal}(s)$*, $|d|_{\mathtt{m}} \leq m$, $|d|_{\mathtt{e}} \leq e$. Moreover, if Φ is tight then $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$.*

Proof. By induction on $m + e$ and case analysis on whether t reduces or not. If t is a $\to_{\text{cbn}}$ normal form then we only have to prove the *moreover* part, that states if Φ is tight then $m = e = 0$, which follows from Proposition 3.

Otherwise, two cases:

1. *Multiplicative steps*: $t \to_{\mathtt{m}} u$ and by quantitative subject reduction (Proposition 2) there is a derivation $\Psi \triangleright_{\text{CbN}} \vdash^{(m-1,e)} u : L$. By *i.h.*, there exist s and d' such that $\mathsf{normal}(s)$ and $d' : u \to^*_{\text{cbn}} s$, $|d'|_{\mathtt{m}} \leq m - 1$ and $|d'|_{\mathtt{e}} \leq e$. Just note that $t \to_{\mathtt{m}} u$ and so, if $d : t \to^*_{\text{cbn}} s$ is d' preceeded by such a step, we have $|d|_{\mathtt{m}} \leq m$ and $|d|_{\mathtt{e}} \leq e$.
   If Φ is tight then Ψ is tight. Then $|d'|_{\mathtt{m}} = m - 1$ and $|d'|_{\mathtt{e}} = e$ by *i.h.*, that give $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$.

2. *Exponential steps*: $t \to_{\mathtt{e}} u$ and by quantitative subject reduction (Proposition 2) there is a derivation $\Psi \triangleright_{\text{CbN}} \vdash^{(m,e-1)} u : L$. By *i.h.*, there exist s and d' such that $\mathsf{normal}(s)$ and $d' : u \to^*_{\text{cbn}} s$, $|d'|_{\mathtt{m}} \leq m$ and $|d'|_{\mathtt{e}} \leq e - 1$. Just note that $t \to_{\mathtt{e}} u$ and so, if $d : t \to^*_{\text{cbn}} s$ is d' preceeded by such a step, we have $|d|_{\mathtt{m}} \leq m$ and $|d|_{\mathtt{e}} \leq e$.
If Φ is tight then Ψ is tight. Then $|d'|_{\mathtt{m}} = m$ and $|d'|_{\mathtt{e}} = e - 1$ by *i.h.*, that give $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$. □

B.2 CbN Completeness

Proposition 17 (Normal forms are tightly typable for CbN). *Let t be such that $\mathsf{normal}(t)$. Then there is tight derivation $\Phi \triangleright_{\text{cbn}} \vdash^{(0,0)} t : \mathsf{normal}$.*

Proof. By induction on $\mathsf{normal}(t)$. Cases:
– *Abstraction*: if $\mathsf{normal}(t)$ because $t = \lambda x.s$ then

$$\dfrac{}{\vdash^{(0,0)} \lambda x.s : \mathsf{normal}}\ \mathsf{normal}$$

– *Substitution*: if $\mathsf{normal}(t)$ because $t = s[x \leftarrow u]$ and $\mathsf{normal}(s)$ then by *i.h.* there exists a tight derivation $\Psi \triangleright_{\text{CbN}} \vdash^{(0,0)} s : \mathsf{normal}$. Then Φ is given by:

$$\dfrac{\Psi \triangleright \ \vdash^{(0,0)} s : \mathsf{normal} \qquad \dfrac{}{\vdash^{(0,0)} u : \mathbf{0}}\ \mathsf{many}}{\vdash^{(0,0)} s[x \leftarrow u] : \mathsf{normal}}\ \mathsf{ES}$$

□

In order to prove subject expansion, we have to first show that typability can also be pulled back along substitutions, via a linear removal lemma. We also need the following merging property of multi sets, whose proof is omitted because straightforward.

Lemma 10 (Merging of multi-sets with respect to derivations). *Let t be a term. For any two derivations*
– *$\Phi_N \triangleright_{CbN}\ \Gamma_N \vdash^{(m_N,e_N)} t : N$, and*
– *$\Phi_O \triangleright_{CbN}\ \Gamma_O \vdash^{(m_O,e_O)} t : O$*
there is a derivation $\Phi_{N \uplus O} \triangleright_{CbN}\ \Gamma_N \uplus \Gamma_O \vdash^{(m_N+m_O,e_N+e_O)} t : N \uplus O$.

Lemma 11 (Linear removal for CbN). *Let $\Phi \triangleright_{\text{cbn}} \Gamma; x : M \vdash^{(m,e)} C\langle\langle s\rangle\rangle : L$, where $x \notin \mathtt{fv}(s)$. Then there exist*
– *a linear type L',*
– *a derivation $\Phi_s \triangleright_{\text{cbn}} \Gamma_s \vdash^{(m_s,e_s)} s : L'$, and*
– *a derivation $\Phi_{C\langle\langle x\rangle\rangle} \triangleright_{\text{cbn}} \Gamma', x : M \uplus [L'] \vdash^{(m',e')} C\langle\langle x\rangle\rangle : L$*
such that
– Type contexts*: $\Gamma = \Gamma_s \uplus \Gamma'$.*
– Indices*: $(m, e) = (m' + m_s, e' + e_s - 1)$.*

Proof. By induction on C. Cases:

- *Empty context, i.e.* $C = \langle\cdot\rangle$. Then $\Phi \triangleright_{\mathrm{CbN}} \Gamma; x:M \vdash^{(m,e)} s:L$. By Lemma 1, $x \notin \mathtt{fv}(s)$ implies $M = \mathbf{0}$. Then we simply take
  - $\Phi_s := \Phi$, that implies $L' := L$, $\Gamma_s := \Gamma$, $m_s := m$, and $e_s := e$, and
  - Φ_x defined as the axiom

    $$\frac{}{x:[L] \vdash^{(0,1)} x:L}\ \mathtt{ax}$$

    and for which Γ' is empty, $m' = 0$, and $e' = 1$.

  Then the statement holds:
  - *Type contexts*: $\Gamma = \emptyset \uplus \Gamma = \emptyset \uplus \Gamma_s = \Gamma' \uplus \Gamma_s$ and
  - *Indices*: $(m, e) = (m_s, e_s) = (0 + m_s, 1 + e_s - 1) = (m' + m_s, e' + e_s - 1)$.
- *Left of an application, i.e.* $C = Du$. Then Φ has the form

  $$\frac{\Phi_{D\langle s\rangle} \triangleright\ \Gamma_{D\langle s\rangle}; x:M \vdash^{(m_{D\langle s\rangle}, e_{D\langle s\rangle})} D\langle s\rangle : N \multimap L \quad \Gamma_u \vdash^{(m_u,e_u)} u:N}{\Gamma_{D\langle s\rangle} \uplus \Gamma_u; x:M \vdash^{(m_{D\langle s\rangle}+m_u+1, e_{D\langle s\rangle}+e_u)} D\langle s\rangle u : L}\ \mathtt{app}$$

  where $x \notin \mathtt{dom}(\Gamma_u)$ (by Lemma 1, because $x \notin \mathtt{fv}(u)$ by hypothesis), $\Gamma = \Gamma_{D\langle s\rangle} \uplus \Gamma_u$, $m = m_{D\langle s\rangle} + m_u + 1$, and $e = e_{D\langle s\rangle} + e_u$.
  Applying the *i.h.* to $\Phi_{D\langle s\rangle}$ provides a type L' and derivations:

  $$\Phi_s \triangleright_{\mathrm{CbN}}\ \Gamma_s \vdash^{(m_s,e_s)} s:L'$$

  and

  $$\Phi_{D\langle\langle x\rangle\rangle} \triangleright_{\mathrm{CbN}}\ \Gamma''; x:M \uplus [L'] \vdash^{(m'',e'')} D\langle\langle x\rangle\rangle : N \multimap L$$

  such that $\Gamma_{D\langle s\rangle} = \Gamma'' \uplus \Gamma_s$ and $(m_{D\langle s\rangle}, e_{D\langle s\rangle}) = (m'' + m_s, e'' + e_s - 1)$.
  Then $\Phi_{C\langle\langle x\rangle\rangle}$ is given by:

  $$\frac{\Phi_{D\langle\langle x\rangle\rangle} \triangleright_{\mathrm{CbN}}\ \Gamma''; x:M \uplus [L'] \vdash^{(m'',e'')} D\langle\langle x\rangle\rangle : N \multimap L \quad \Gamma_u \vdash^{(m_u,e_u)} u:N}{(\Gamma'' \uplus \Gamma_u); x:M \uplus [L'] \vdash^{(m''+m_u+1, e''+e_u)} D\langle x\rangle u : L}\ \mathtt{app}$$

  that, by taking $\Gamma' := \Gamma'' \uplus \Gamma_u$, $m' = m'' + m_u + 1$, and $e' = e'' + e_u$, verifies the statement because:
  - *Type contexts*: $\Gamma = \Gamma_{D\langle s\rangle} \uplus \Gamma_u = \Gamma'' \uplus \Gamma_s \uplus \Gamma_u = \Gamma' \uplus \Gamma_s$, and
  - *Indices*: $(m, e) = (m_{D\langle s\rangle} + m_u + 1, e_{D\langle s\rangle} + e_u) = (m'' + m_s + m_u + 1, e'' + e_s - 1 + e_u) = (m' + m_s, e' + e_s - 1)$.
- *Left of a substitution, i.e.* $C = D[y{\leftarrow}u]$. Then Φ has the form

  $$\frac{\Phi_{D\langle s\rangle} \triangleright\ \Gamma_{D\langle s\rangle}; x:M; y:N \vdash^{(m_{D\langle s\rangle}, e_{D\langle s\rangle})} D\langle s\rangle : L \quad \Gamma_u \vdash^{(m_u,e_u)} u:N}{\Gamma_{D\langle s\rangle} \uplus \Gamma_u; x:M \vdash^{(m_{D\langle s\rangle}+m_u, e_{D\langle s\rangle}+e_u)} D\langle s\rangle[y{\leftarrow}u] : L}\ \mathtt{ES}$$

  where $x \notin \mathtt{dom}(\Gamma_u)$ (by Lemma 1, because $x \notin \mathtt{fv}(u)$ by hypothesis), $\Gamma = \Gamma_{D\langle s\rangle} \uplus \Gamma_u$, $m = m_{D\langle s\rangle} + m_u$, and $e = e_{D\langle s\rangle} + e_u$.
  Applying the *i.h.* to $\Phi_{D\langle s\rangle}$ provides a type L' and derivations:

  $$\Phi_s \triangleright_{\mathrm{CbN}}\ \Gamma_s \vdash^{(m_s,e_s)} s:L'$$

and
$$\Phi_{D\langle\langle x\rangle\rangle} \triangleright_{\text{CbN}} \ \Gamma''; x : M \uplus [L']; y : N \vdash^{(m'',e'')} D\langle\langle x\rangle\rangle : L$$
such that $\Gamma_{D\langle s\rangle} = \Gamma'' \uplus \Gamma_s$ and $(m_{D\langle s\rangle}, e_{D\langle s\rangle}) = (m'' + m_s, e'' + e_s - 1)$. Then $\Phi_{C\langle\langle x\rangle\rangle}$ is given by:

$$\dfrac{\Phi_{D\langle\langle x\rangle\rangle} \triangleright_{\text{CbN}} \ \Gamma''; x : M \uplus [L']; y : N \vdash^{(m'',e'')} D\langle\langle x\rangle\rangle : L \quad \Gamma_u \vdash^{(m_u,e_u)} u : N}{(\Gamma'' \uplus \Gamma_u); x : M \uplus [L'] \vdash^{(m''+m_u,e''+e_u)} D\langle x\rangle[y{\leftarrow}u] : L}\ \texttt{ES}$$

that, by taking $\Gamma' := \Gamma'' \uplus \Gamma_u$, $m' = m'' + m_u$, and $e' = e'' + e_u$, verifies the statement because:

- *Type contexts*: $\Gamma = \Gamma_{D\langle s\rangle} \uplus \Gamma_u = \Gamma'' \uplus \Gamma_s \uplus \Gamma_u = \Gamma' \uplus \Gamma_s$, and
- *Indices*: $(m, e) = (m_{D\langle s\rangle} + m_u, e_{D\langle s\rangle} + e_u) = (m'' + m_s + m_u, e'' + e_s - 1 + e_u) = (m' + m_s, e' + e_s - 1)$. □

Proposition 18 (Quantitative subject expansion for CbN). *Let $\Phi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e)} s : L$ be a derivation.*

1. Multiplicative*: if $t \to_{\mathtt{m}_{\text{cbn}}} s$ then there is a derivation $\Psi \triangleright_{\text{cbn}} \Gamma \vdash^{(m+1,e)} t : L$.*

2. Exponential*: if $t \to_{\mathtt{e}_{\text{cbn}}} s$ then there is a derivation $\Psi \triangleright_{\text{cbn}} \Gamma \vdash^{(m,e+1)} t : L$.*

Proof.

1. By induction on $t \to_{\mathtt{m}} s$. Cases:
   - *Step at top level, i.e.* $t = S\langle\lambda x.u\rangle r \to_{\mathtt{m}} S\langle u[x{\leftarrow}r]\rangle = s$. This case is itself by induction on S. Two sub-cases:
     - *Empty substitution context, i.e.* $S = \langle\cdot\rangle$. The derivation Φ has the form:
     $$\dfrac{x : M; \Gamma_u \vdash^{(m_u,e_u)} u : L \quad \Gamma_r \vdash^{(m_r,e_r)} r : M}{\Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r,e_u+e_r)} u[x{\leftarrow}r] : L}\ \text{ES}$$
     With $\Gamma = \Gamma_u \uplus \Gamma_r$, $m = m_u + m_r$, and $e = e_u + e_r$. We construct the following derivation Ψ, verifying the statement:
     $$\dfrac{\dfrac{x : M; \Gamma_u \vdash^{(m_u,e_u)} u : L}{\Gamma_u \vdash^{(m_u,e_u)} \lambda x.u : M \to L}\ \texttt{fun} \qquad \Gamma_r \vdash^{(m_r,e_r)} r : M}{\Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r+1,e_u+e_r)} (\lambda x.u) r : L}\ \texttt{app}$$
     - *Non-empty substitution context, i.e.* $S = S'[y{\leftarrow}q]$. Then Φ has the following structure:
     $$\dfrac{y : N; \Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r,e_u+e_r)} S'\langle u[x{\leftarrow}r]\rangle : L \quad \Gamma_q \vdash^{(m_q,e_q)} q : N}{\Gamma_u \uplus \Gamma_q \uplus \Gamma_r \vdash^{(m_u+m_q+m_r,e_u+e_q+e_r)} S'\langle u[x{\leftarrow}r]\rangle[y{\leftarrow}q] : L}\ \text{ES}$$
     With $\Gamma = \Gamma_u \uplus \Gamma_q \uplus \Gamma_r$, $m = m_u + m_q + m_r + 1$, and $e = e_u + e_q + e_r$. By *i.h.* applied to the left premise, we obtain a derivation

$$y : N; \Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r+1,e_u+e_r)} S'\langle\lambda x.u\rangle r : L$$

that has the following structure:

$$\frac{y : N; \Gamma_u \vdash^{(m_u,e_u)} S'\langle\lambda x.u\rangle : M \to L \quad \Gamma_r \vdash^{(m_r,e_r)} r : M}{y : N; \Gamma_u \uplus \Gamma_r \vdash^{(m_u+m_r+1,e_u+e_r)} S'\langle\lambda x.u\rangle r : L}\ \mathtt{app}$$

We then construct Ψ has follows:

$$\frac{\dfrac{y : N; \Gamma_u \vdash^{(m_u,e_u)} S'\langle\lambda x.u\rangle : M \to L \quad \Gamma_q \vdash^{(m_q,e_q)} q : N}{\Gamma_u \uplus \Gamma_q \vdash^{(m_u+m_q,e_u+e_q)} S'\langle\lambda x.u\rangle[y\leftarrow q] : M \to L}\ \mathrm{ES} \qquad \Gamma_r \vdash^{(m_r,e_r)} r : M}{\Gamma_u \uplus \Gamma_q \uplus \Gamma_r \vdash^{(m_u+m_q+m_r+1,e_u+e_q+e_r)} S'\langle\lambda x.u\rangle[y\leftarrow q] r : L}\ \mathtt{app}$$

– *Contextual closure.* We have $t = C\langle u\rangle \to_{\mathtt{m}} C\langle r\rangle = s$. Cases of C:
  - *Left on an application, i.e.* $C = Dq$. The last typing rule in Φ is necessarily $\mathtt{app}$ and Φ is of the form

$$\frac{\Gamma_r \vdash^{(m_r,e_r)} D\langle r\rangle : M \multimap L \quad \Gamma_q \vdash^{(m_q,e_q)} q : M}{\Gamma_r \uplus \Gamma_q \vdash^{(m_r+m_q+1,e_r+e_q)} D\langle r\rangle q : L}\ \mathtt{app}_b$$

    With $\Gamma = \Gamma_r \uplus \Gamma_q$, $m = m_r + m_q + 1$, and $e = e_r + e_q$.
    By *i.h.*, there exists a derivation $\Gamma_u \vdash^{(m_r+1,e_r)} D\langle u\rangle : M \multimap L$, thus allowing us to construct Ψ as follows:

$$\frac{\Gamma_r \vdash^{(m_r+1,e_r)} D\langle u\rangle : M \multimap L \quad \Gamma_q \vdash^{(m_q,e_q)} q : M}{\Gamma_r \uplus \Gamma_q \vdash^{(m_r+m_q+2,e_r+e_q)} D\langle u\rangle q : L}\ \mathtt{app}$$

    Note that $(m_r + m_q + 2, e_r + e_q) = (m + 1, e)$.
  - Let $C = D[x\leftarrow q]$. The last typing rule in Φ is necessarily ES and Φ is of the form

$$\frac{\Gamma_r, x : M \vdash^{(m_r,e_r)} D\langle r\rangle : L \quad \Gamma_q \vdash^{(m_q,e_q)} q : M}{\Gamma_r \uplus \Gamma_q \vdash^{(m_r+m_q,e_r+e_q)} D\langle u\rangle[x\leftarrow q] : L}\ \mathrm{ES}$$

    With $\Gamma = \Gamma_r \uplus \Gamma_q$, $m = m_r + m_q$, and $e = e_r + e_q$.
    By *i.h.*, there exists a derivation $\Gamma_r, x : M \vdash^{(m_r+1,e_r)} D\langle u\rangle : L$, thus allowing us to construct Ψ as follows:

$$\frac{\Gamma_r, x : M \vdash^{(m_r+1,e_r)} D\langle u\rangle : L \quad \Gamma_q \vdash^{(m_q,e_q)} q : M}{\Gamma_r \uplus \Gamma_q \vdash^{(m_r+m_q+1,e_r+e_q)} D\langle u\rangle[x\leftarrow q] : L}\ \mathrm{ES}$$

    Note that $(m_r + m_q + 1, e_r + e_q) = (m + 1, e)$.

2. By induction on $t \to_{\mathtt{e}} s$.

– *Step at top level, i.e.* $t = C\langle\langle x\rangle\rangle[x{\leftarrow}u] \to_{\mathtt{e}} C\langle\langle u\rangle\rangle[x{\leftarrow}u] = s$. The last typing rule of Φ is necessarily ES and Φ is of the form

$$\dfrac{\Phi_{C\langle\langle x\rangle\rangle} \rhd\ \Gamma_C, x : M \vdash^{(m_C,e_C)} C\langle\langle u\rangle\rangle : L \quad \Pi \vdash^{(m',e')} u : M}{\Gamma_C \uplus \Pi \vdash^{(m_C+m_u,e_C+e_u)} C\langle\langle u\rangle\rangle[x{\leftarrow}u] : L}\ \text{ES}$$

With $\Gamma = \Gamma_C \uplus \Pi$, $m = m_C + m'$, and $e = e_C + e'$.
The linear removal lemma (Lemma 3) applied to $\Phi_{C\langle\langle u\rangle\rangle}$ gives a type L' and two derivations
(a) $\Phi_u \rhd_{\text{CbN}}\ \Gamma_u \vdash^{(m_u,e_u)} u : L'$, and
(b) $\Phi_{C\langle\langle x\rangle\rangle} \rhd_{\text{CbN}}\ \Gamma'; x{:}M \uplus [L'] \vdash^{(m'',e'')} C\langle\langle x\rangle\rangle : L$
such that $\Gamma_C = \Gamma_u \uplus \Gamma'$, $m_C = m_u + m''$, and $e_C = e_u + e'' - 1$.
Now, by applying the multi-sets merging lemma (Lemma 10) to Φ_u and the right premise of Φ:

$$\Pi \vdash^{(m',e')} u : M$$

we obtain a derivation

$$\Psi_u \rhd_{\text{CbN}}\ \Pi \uplus \Gamma_u \vdash^{(m'+m_u,e'+e_u)} u : M \uplus [L']$$

Then Ψ is built as follows:

$$\dfrac{\Gamma'; x{:}M \uplus [L'] \vdash^{(m'',e'')} C\langle\langle x\rangle\rangle : L \quad \Pi \uplus \Gamma_u \vdash^{(m'+m_u,e'+e_u)} u : M \uplus [L']}{\Gamma' \uplus \Pi \uplus \Gamma_u \vdash^{(m''+m'+m_u,e''+e'+e_u)} C\langle\langle x\rangle\rangle[x{\leftarrow}u] : L}\ \text{ES}$$

Now, note that the last judgement is in fact

$$\Gamma_C \uplus \Pi \vdash^{(m_C+m',e_C+e'+1)} C\langle\langle x\rangle\rangle[x{\leftarrow}u] : L$$

which in turn is

$$\Gamma \vdash^{(m,e+1)} C\langle\langle x\rangle\rangle[x{\leftarrow}u] : L$$

as required.

– *Contextual closure.* As in the $\to_{\mathtt{m}}$ case. Note that indeed those cases do not depend on the details of the step itself, but only on the context enclosing it. □

Theorem 9 (CbN tight completeness). *Let t be a closed term. If $d{:}\,t \to^*_{\text{cbn}} s$ and $\mathsf{normal}(s)$ then there is a tight derivation $\Phi \rhd_{\text{cbn}} \vdash^{(|d|_{\mathtt{m}},|d|_{\mathtt{e}})} t : \mathsf{normal}$.*

Proof. By induction on the length $k := |d|$ of the evaluation $d{:}\ t \to^*_{\text{cbn}} s$. If $k = 0$ then $t = s$ and $\mathsf{normal}(t)$. Proposition 4 gives the existence of a tight derivation $\Phi \rhd_{\text{cbn}}\ \vdash^{(0,0)} t : \mathsf{normal}$, that satisfies the statement because $|d|_{\mathtt{m}} = |d|_{\mathtt{e}} = 0$.

If $k > 0$ then $t \to_{\text{cbn}} u \to^{k-1}_{\text{cbn}} s$. Let d' be the evaluation $u \to^{k-1}_{\text{cbn}} s$. By *i.h.* there exists a tight derivation $\Psi \rhd_{\text{cbn}} \vdash^{(|d'|_{\mathtt{m}},|d'|_{\mathtt{e}})} u : \mathsf{normal}$. By quantitative subject expansion Proposition 5 there exists a derivation Φ of t with the same types in the ending judgement of Ψ—then Φ is tight—and with indices $(|d|_{\mathtt{m}}, |d|_{\mathtt{e}})$. □

C Types by Value (Sect. 5)

Lemma 12 (Type contexts and variable occurrences for CbV). *Let $\Phi \triangleright_{\text{cbv}} \ \Gamma \vdash^{(m,e)} t : M$ be a derivation. If $x \notin \mathtt{fv}(t)$ then $x \notin \mathtt{dom}(\Gamma)$.*

Proof. By straightforward induction on the derivation Φ. □

C.1 CbV Correctness

Lemma 13 (Typing values). *Let v be a value and $\Phi \triangleright_{\text{cbv}} \ \Gamma \vdash^{(m,e)} v : M$ be a derivation for it. Then,*

1. Empty multi-set implies null size*: if $M = \mathbf{0}$, then $\mathtt{dom}(\Gamma) = \emptyset$ with $m = 0 = e$.*
2. Multi-set splitting*: if $M = N \uplus O$, then there are two type contexts Π and Δ and two derivations $\Psi \triangleright_{\text{cbv}} \ \Pi \vdash^{(m',e')} v : N$ and $\Theta \triangleright_{\text{cbv}} \ \Delta \vdash^{(m'',e'')} v : O$ such that $\Gamma = \Pi \uplus \Delta$, $m = m' + m''$ and $e = e' + e''$*

Lemma 14 (CbV linear substitution). *Let $\Phi \triangleright_{\text{cbv}} \Gamma, x : M \vdash^{(m,e)} V\langle\!\langle x \rangle\!\rangle : N$. Then there exists a splitting $M = O \uplus P$ such that, for every derivation $\Psi \triangleright_{\text{cbv}} \ \Pi \vdash^{(m',e')} v : O$, there is a derivation $\Phi' \triangleright_{\text{cbv}} \Gamma \uplus \Pi, x : P \vdash^{(m+m',e+e'-1)} V\langle\!\langle v \rangle\!\rangle : N$.*

Proof. By induction on V. Cases:

- *Hole, i.e.* $V = \langle\cdot\rangle$, then $V\langle\!\langle v \rangle\!\rangle = v$ and $V\langle\!\langle x \rangle\!\rangle = x$, hence Φ consists only of an ax-rule and so $N = M$ and $\mathtt{dom}(\Gamma) = \emptyset$, with $m = 0$ and $e = 1$. Let $O \coloneqq M$ and $P \coloneqq \mathbf{0}$. Thus, every $\Psi \triangleright_{\text{cbv}} \ \Pi \vdash^{(m',e')} v : O$ coincides with a type derivation $\Phi' \triangleright_{\text{cbv}} \ \Gamma \uplus \Pi, x : P \vdash^{(m+m',e+e'-1)} V\langle\!\langle v \rangle\!\rangle : N$, since $\Gamma \uplus \Pi, x : P = \Gamma$ and $N = O$ and $(m + m', e + e' - 1) = (m', e')$.
- *Left application, i.e.* $V \coloneqq V't$. Then, the derivation Φ has the form

$$\frac{x : M_1, \Gamma_1 \vdash^{(m_1,e_1)} V'\langle\!\langle x \rangle\!\rangle : [N' \multimap N] \qquad x : M_2, \Gamma_2 \vdash^{(m_2,e_2)} t : N'}{x : M, \Gamma \vdash^{(m,e)} V'\langle\!\langle x \rangle\!\rangle t : N}\ \mathtt{app}$$

  where $M = M_1 \uplus M_2$, $\Gamma = \Gamma_1 \uplus \Gamma_2$, $m = m_1 + m_2 + 1$ and $e = e_1 + e_2$. By *i.h.*, there exists a splitting $M_1 = O \uplus P'$ such that, for every derivation $\Psi \triangleright_{\text{cbv}} \ \Pi \vdash^{(m',e')} v : O$, there exists a derivation with conclusion $\Gamma_1 \uplus \Pi, x : P' \vdash^{(m_1+m',e_1+e'-1)} V'\langle\!\langle v \rangle\!\rangle : [N' \multimap N]$. So, we can construct the following derivation Φ'

$$\frac{\Gamma_1 \uplus \Pi, x : P' \vdash^{(m_1+m',e_1+e'-1)} V'\langle\!\langle v \rangle\!\rangle : [N' \multimap N] \qquad x : M_2, \Gamma_2 \vdash^{(m_2,e_2)} t : N'}{x : M_2 \uplus P', \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} V'\langle\!\langle v \rangle\!\rangle t : N}\ \mathtt{app}$$

  where $P \coloneqq M_2 \uplus P'$ and $M = M_1 \uplus M_2 = O \uplus P' \uplus M_2 = O \uplus P$.
- *Right application, i.e.* $V \coloneqq tV'$. Then, the derivation Φ has the form

$$\frac{x : M_1, \Gamma_1 \vdash^{(m_1,e_1)} t : [N' \multimap N] \qquad x : M_2, \Gamma_2 \vdash^{(m_2,e_2)} V'\langle\!\langle x \rangle\!\rangle : N'}{x : M, \Gamma \vdash^{(m,e)} t\, V'\langle\!\langle x \rangle\!\rangle : N}\ \mathtt{app}$$

where $M = M_1 \uplus M_2$, $\Gamma = \Gamma_1 \uplus \Gamma_2$, $m = m_1 + m_2 + 1$ and $e = e_1 + e_2$. By *i.h.*, there exists a splitting $M_2 = O \uplus P'$ such that, for every derivation $\Psi \triangleright_{\text{cbv}} \ \Pi \vdash^{(m',e')} v : O$, there exists a derivation with conclusion $\Gamma_2 \uplus \Pi, x : P' \vdash^{(m_2+m',e_2+e'-1)} V'\langle\langle v \rangle\rangle : N'$. So, we can construct the following derivation Φ'

$$\dfrac{x : M_1, \Gamma_1 \vdash^{(m_1,e_1)} t : [N' \multimap N] \qquad \Gamma_2 \uplus \Pi, x : P' \vdash^{(m_2+m',e_2+e'-1)} V'\langle\langle v \rangle\rangle : N'}{x : M_1 \uplus P', \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} tV'\langle\langle v \rangle\rangle : N} \ \texttt{app}$$

where $P \coloneqq M_1 \uplus P'$ and $M = M_1 \uplus M_2 = M_1 \uplus O \uplus P' = O \uplus P$.

– *Left explicit substitution*, *i.e.* $V \coloneqq V'[y \leftarrow t]$. We can suppose without loss of generality that $y \notin \mathtt{fv}(t) \cup \mathtt{fv}(v) \cup \{x\}$, and hence $y \notin \mathtt{dom}(\Pi)$ by Lemma 12. So, the derivation Φ has the form

$$\dfrac{x : M_1, y : N', \Gamma_1 \vdash^{(m_1,e_1)} V'\langle\langle x \rangle\rangle : N \qquad x : M_2, \Gamma_2 \vdash^{(m_2,e_2)} t : N'}{x : M, \Gamma \vdash^{(m,e)} V'\langle\langle x \rangle\rangle[y \leftarrow t] : N} \ \texttt{ES}$$

where $M = M_1 \uplus M_2$, $\Gamma = \Gamma_1 \uplus \Gamma_2$, $m = m_1 + m_2$ and $e = e_1 + e_2$. By *i.h.*, there exists a splitting $M_1 = O \uplus P'$ such that, for every derivation $\Psi \triangleright_{\text{cbv}}$ $\Pi \vdash^{(m',e')} v : O$, there exists a derivation of $\Gamma_1 \uplus \Pi, x : P', y : N' \vdash^{(m_1+m',e_1+e'-1)} V'\langle\langle v \rangle\rangle : N$. Therefore, we can construct the following derivation Φ'

$$\dfrac{\Gamma_1 \uplus \Pi, x : P', y : N' \vdash^{(m_1+m',e_1+e'-1)} V'\langle\langle v \rangle\rangle : N \qquad x : M_2, \Gamma_2 \vdash^{(m_2,e_2)} t : N'}{x : M_2 \uplus P', \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} V'\langle\langle v \rangle\rangle[y \leftarrow t] : N} \ \texttt{ES}$$

where $P \coloneqq M_2 \uplus P'$ and $M = M_1 \uplus M_2 = O \uplus P' \uplus M_2 = O \uplus P$.

– *Right explicit substitution*, *i.e.* $V \coloneqq t[y \leftarrow V']$. We can suppose without loss of generality that $y \notin \mathtt{fv}(v) \cup \{x\}$, and hence $y \notin \mathtt{dom}(\Pi)$ by Lemma 12. Then, the derivation Φ has the form

$$\dfrac{x : M_1, y : N', \Gamma_1 \vdash^{(m_1,e_1)} t : N \qquad x : M_2, \Gamma_2 \vdash^{(m_2,e_2)} V'\langle\langle x \rangle\rangle : N'}{x : M, \Gamma \vdash^{(m,e)} t[y \leftarrow V'\langle\langle x \rangle\rangle] : N} \ \texttt{ES}$$

where $M = M_1 \uplus M_2$, $\Gamma = \Gamma_1 \uplus \Gamma_2$, $m = m_1 + m_2$ and $e = e_1 + e_2$. By *i.h.*, there is a splitting $M_2 = O \uplus P'$ such that, for every derivation $\Psi \triangleright_{\text{cbv}} \ \Pi \vdash^{(m',e')} v : O$, there exists a derivation of $\Gamma_2 \uplus \Pi, x : P' \vdash^{(m_2+m',e_2+e'-1)} V'\langle\langle v \rangle\rangle : N'$. So, we can construct the following derivation Φ'

$$\dfrac{x : M_1, y : N', \Gamma_1 \vdash^{(m_1,e_1)} t : N \qquad \Gamma_2 \uplus \Pi, x : P' \vdash^{(m_2+m',e_2+e'-1)} V'\langle\langle v \rangle\rangle : N'}{x : M_1 \uplus P', \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} t[y \leftarrow V'\langle\langle v \rangle\rangle] : N} \ \texttt{ES}$$

where $P \coloneqq M_1 \uplus P'$ and $M = M_1 \uplus M_2 = M_1 \uplus O \uplus P' = O \uplus P$. □

Proposition 19 (Quantitative subject reduction for CbV). *Let $\Phi \triangleright_{\text{cbv}}$ $\Gamma \vdash^{(m,e)} t : M$ be a derivation.*

1. Multiplicative*: if* $t \to_{\mathtt{m}_{\mathrm{cbv}}} t'$ *then* $m \geq 1$ *and there exists a derivation* $\Phi' \rhd_{\mathrm{cbv}} \Gamma \vdash^{(m-1,e)} t' : M$.
2. Exponential*: if* $t \to_{\mathtt{e}_{\mathrm{cbv}}} t'$ *then* $e \geq 1$ *and there exists a derivation* $\Phi' \rhd_{\mathrm{cbv}} \Gamma \vdash^{(m,e-1)} t' : M$.

Proof. By induction on the reduction relation $\to_{\mathrm{cbv}}$, with the root rules $\mapsto_{\mathtt{m}}$ and $\mapsto_{\mathtt{e}_{\mathrm{cbv}}}$ as the base case, and the closure by CbV contexts of $\mapsto_{\mathrm{cbv}} := \mapsto_{\mathtt{m}} \cup \mapsto_{\mathtt{e}_{\mathrm{cbv}}}$ as the inductive one.

- *Root step for* $\to_{\mathtt{m}_{\mathrm{cbv}}}$ *i.e.* $t = S\langle \lambda x.u\rangle r \mapsto_{\mathtt{m}} S\langle u[x{\leftarrow}r]\rangle = t'$ where $S := [y_1{\leftarrow}s_1]\dots[y_n{\leftarrow}s_n]$ for some $n \geq 0$. We proceed by induction on $n \in \mathbb{N}$.
  If $n = 0$ then $S = \langle\cdot\rangle$ and so $t = S\langle \lambda x.u\rangle r = (\lambda x.u)r$ and $t' = S\langle u[x{\leftarrow}r]\rangle = u[x{\leftarrow}r]$. Hence, Φ has the form

$$\dfrac{\dfrac{\dfrac{\Psi \rhd_{\mathrm{cbv}} \Pi, x:O \vdash^{(m',e')} u:M}{\Pi \vdash^{(m',e')} \lambda x.u : O \multimap M}\,\texttt{fun}}{\Pi \vdash^{(m',e')} \lambda x.u : [O \multimap M]}\,\texttt{many} \qquad \Theta \rhd_{\mathrm{cbv}} \Delta \vdash^{(m'',e'')} r:O}{\Pi \uplus \Delta \vdash^{(1+m'+m'',e'+e'')} (\lambda x.u)r : M}\,\texttt{app}$$

  where $\Gamma := \Pi \uplus \Delta$, $m := 1 + m' + m''$ and $e := e' + e''$. Therefore, $m \geq 1$. We can construct the following derivation Φ':

$$\dfrac{\Psi \rhd_{\mathrm{cbv}} \Pi, x:O \vdash^{(m',e')} u:M \quad \Theta \rhd_{\mathrm{cbv}} \Delta \vdash^{(m'',e'')} r:O}{\Gamma \vdash^{(m'+m'',e'+e'')} u[x{\leftarrow}r] : M}\,\texttt{ES}$$

  where $(m' + m'', e' + e'') = (m - 1, e)$.
  Suppose now $n > 0$. Let $S' := [y_1{\leftarrow}s_1]\dots[y_{n-1}{\leftarrow}s_{n-1}]$: then, $t = S\langle \lambda x.u\rangle r = S'\langle \lambda x.u\rangle[y_n{\leftarrow}s_n]r$ and $t' = S\langle u[x{\leftarrow}r]\rangle = S'\langle u[x{\leftarrow}r]\rangle[y_n{\leftarrow}s_n]$. Hence, Φ has the form

$$\dfrac{\dfrac{\Psi'' \rhd_{\mathrm{cbv}} \Pi, y_n:N_n \vdash^{(m'',e'')} S'\langle \lambda x.u\rangle : M \qquad \Psi_n \rhd_{\mathrm{cbv}} \Gamma_n \vdash^{(m_n,e_n)} s_n : N_n}{\Pi \uplus \Gamma_n \vdash^{(m''+m_n,e''+e_n)} S\langle \lambda x.u\rangle : M}\,\texttt{ES} \qquad \Theta \rhd_{\mathrm{cbv}} \Gamma_0' \vdash^{(m_0',e_0')} r:O}{\Pi \uplus \Gamma_n \uplus \Gamma_0' \vdash^{(m''+m_n+m_0'+1,e''+e_n+e_0')} S\langle \lambda x.u\rangle r : M}\,\texttt{app}$$

  where $\Gamma := \Pi \uplus \Gamma_n \uplus \Gamma_0'$ and $(m, e) := (m'' + m_n + m_0' + 1, e'' + e_n + e_0')$. Note that $m \geq 1$ as required. Consider the following derivation Ψ

$$\dfrac{\Psi'' \rhd_{\mathrm{cbv}} \Pi, y_n:N_n \vdash^{(m'',e'')} S'\langle \lambda x.u\rangle : M \qquad \Theta \rhd_{\mathrm{cbv}} \Gamma_0' \vdash^{(m_0',e_0')} r:O}{\Pi \uplus \Gamma_0', y_n:N_n \vdash^{(m''+m_0'+1,e''+e_0')} S'\langle \lambda x.u\rangle r : M}\,\texttt{app}$$

  By *i.h.* applied to Ψ (since $S'\langle \lambda x.u\rangle r \mapsto_{\mathtt{m}} S'\langle u[x{\leftarrow}r]\rangle$), there is a derivation

$$\Psi' \rhd_{\mathrm{cbv}} \Pi \uplus \Gamma_0', y_n:N_n \vdash^{(m',e')} S'\langle u[x{\leftarrow}r]\rangle : M$$

  where $(m', e') = (m'' + m_0', e'' + e_0')$. We can then construct the following derivation Φ'

$$\dfrac{\Psi' \rhd_{\mathrm{cbv}} \Pi \uplus \Gamma_0', y_n:N_n \vdash^{(m',e')} S'\langle u[x{\leftarrow}r]\rangle : M \qquad \Psi_n \rhd_{\mathrm{cbv}} \Gamma_n \vdash^{(m_n,e_n)} s_n : N_n}{\Gamma \vdash^{(m'+m_n,e'+e_n)} S'\langle u[x{\leftarrow}r]\rangle[y_n{\leftarrow}s_n] : M}\,\texttt{ES}$$

  where $(m' + m_n, e' + e_n) = (m'' + m_0' + m_n, e'' + e_0' + e_n) = (m - 1, e)$.

– *Root step for* $\to_{\mathsf{e}_{\text{cbv}}}$, *i.e.* $t \coloneqq V\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S\langle v\rangle] \mapsto_{\mathsf{e}_{\text{cbv}}} S\langle V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle =: t'$ with $S \coloneqq [y_1{\leftarrow}s_1]\dots[y_n{\leftarrow}s_n]$ for some $n \geq 0$. We proceed by induction on $n \in \mathbb{N}$. If $n = 0$ then $S = \langle\cdot\rangle$ and so $t = V\langle\!\langle x\rangle\!\rangle[x{\leftarrow}v]$ and $t' = V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]$. Hence, Φ has the form

$$\frac{\Psi \rhd_{\text{cbv}} \ \Pi, x\!:\!O \vdash^{(m',e')} V\langle\!\langle x\rangle\!\rangle : M \qquad \Theta \rhd_{\text{cbv}} \ \Gamma_0 \vdash^{(m_0,e_0)} v : O}{\Pi \uplus \Gamma_0 \vdash^{(m'+m_0,e'+e_0)} V\langle\!\langle x\rangle\!\rangle[x{\leftarrow}v] : M}\ \texttt{ES}$$

where $\Gamma \coloneqq \Pi \uplus \Gamma_0$, and $m \coloneqq m' + m_0$ and $e \coloneqq e' + e_0$. Let $O = O' \uplus O''$ be the splitting of O given by linear substitution for CbV (Lemma 4). According to the multiset splitting property (Lemma 13.2), there exist a splitting $\Gamma_0 = \Gamma_0' \uplus \Gamma_0''$ and the derivations $\Theta' \rhd_{\text{cbv}} \Gamma_0' \vdash^{(m_0',e_0')} v : O'$ and $\Theta'' \rhd_{\text{cbv}} \Gamma_0'' \vdash^{(m_0'',e_0'')} v : O''$, with $m_0 = m_0' + m_0''$ and $e_0 = e_0' + e_0''$. By linear substitution for CbV (Lemma 4), there exists a derivation $\Psi' \rhd_{\text{cbv}} \ \Pi \uplus \Gamma_0', x\!:\!O'' \vdash^{(m'+m_0',e'+e_0'-1)} V\langle\!\langle v\rangle\!\rangle : M$. We can then construct the following derivation Φ'

$$\frac{\Psi' \rhd_{\text{cbv}} \ \Pi \uplus \Gamma_0', x\!:\!O'' \vdash^{(m'+m_0',e'+e_0'-1)} V\langle\!\langle v\rangle\!\rangle : M \qquad \Theta'' \rhd_{\text{cbv}} \ \Gamma_0'' \vdash^{(m_0'',e_0'')} v : O''}{\Pi \uplus \Gamma_0' \uplus \Gamma_0'' \vdash^{(m'+m_0'+m_0'',e'+e_0'+e_0''-1)} V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v] : M}\ \texttt{ES}$$

where $\Pi \uplus \Gamma_0' \uplus \Gamma_0'' = \Pi \uplus \Gamma_0 = \Gamma$ and $(m' + m_0' + m_0'', e' + e_0' + e_0'' - 1) = (m' + m_0, e' + e_0 - 1) = (m, e - 1)$.
Suppose now $n > 0$. Let $S' \coloneqq [y_1{\leftarrow}s_1]\dots[y_{n-1}{\leftarrow}s_{n-1}]$: then, $S\langle v\rangle = S'\langle v\rangle[y_n{\leftarrow}s_n]$ and so $t = V\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S'\langle v\rangle[y_n{\leftarrow}s_n]]$ and $t' = S\langle V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle = S'\langle V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle[y_n{\leftarrow}s_n]$. Hence, Φ has the form

$$\frac{\Theta \rhd_{\text{cbv}} \ \Gamma_0', x\!:\!N \vdash^{(m_0',e_0')} V\langle\!\langle x\rangle\!\rangle : M \quad \dfrac{\Theta'' \rhd_{\text{cbv}} y_n\!:\!N_n, \Gamma_0'' \vdash^{(m_0'',e_0'')} S'\langle v\rangle : N \quad \Theta_n \rhd_{\text{cbv}} \Gamma_n \vdash^{(m_n,e_n)} s_n : N_n}{\Gamma_0'' \uplus \Gamma_n \vdash^{(m_0''+m_n,e_0''+e_n)} S\langle v\rangle : N}\ \texttt{ES}}{\Gamma_0' \uplus \Gamma_0'' \uplus \Gamma_n \vdash^{(m_0'+m''+m_n,e_0'+e''+e_n)} V\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S\langle v\rangle] : M}\ \texttt{ES}$$

where $\Gamma \coloneqq \Gamma_0' \uplus \Gamma_0'' \uplus \Gamma_n$ and $(m, e) \coloneqq (m_0' + m'' + m_n, e_0' + e'' + e_n)$. Consider the following derivation Ψ

$$\frac{\Theta \rhd_{\text{cbv}} \ \Gamma_0', x\!:\!N \vdash^{(m_0',e_0')} V\langle\!\langle x\rangle\!\rangle : M \quad \Theta'' \rhd_{\text{cbv}} y_n\!:\!N_n, \Gamma_0'' \vdash^{(m_0'',e_0'')} S'\langle v\rangle : N}{\Gamma_0' \uplus \Gamma_0'', y_n\!:\!N_n \vdash^{(m_0'+m'',e_0'+e'')} V\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S'\langle v\rangle] : M}\ \texttt{ES}$$

By *i.h.* applied to Ψ (since $V\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S'\langle v\rangle] \mapsto_{\mathsf{e}_{\text{cbv}}} S'\langle V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle$), one has $e_0' + e'' \geq 1$ and there exists a derivation

$$\Psi' \rhd_{\text{cbv}} \ \Gamma_0' \uplus \Gamma_0'', y_n\!:\!N_n \vdash^{(m',e')} S'\langle V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle : M$$

where $(m', e') \coloneqq (m_0' + m'', e_0' + e'' - 1)$. We can then construct the following derivation Φ':

$$\frac{\Psi' \rhd_{\text{cbv}} \ \Gamma_0' \uplus \Gamma_0'', y_n\!:\!N_n \vdash^{(m',e')} S'\langle V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle : M \qquad \Theta_n \rhd_{\text{cbv}} \Gamma_n \vdash^{(m_n,e_n)} s_n : N_n}{\Gamma \vdash^{(m'+m_n,e'+e_n)} S'\langle V\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle[y_n{\leftarrow}s_n] : M}\ \texttt{ES}$$

where $(m' + m_n, e' + e_n) = (m_0' + m'' + m_n, e_0' + e'' - 1 + e_n) = (m, e - 1)$.

– *Application left, i.e.* $t \coloneqq su \to_{\mathsf{r}} s'u =: t'$ with $s \to_{\mathsf{r}} s'$ and $\mathsf{r} \in \{\mathtt{m}_{\mathrm{cbv}}, \mathsf{e}_{\mathrm{cbv}}\}$. So, Φ has the form

$$\frac{\Psi_1 \rhd_{\mathrm{cbv}} \ \Gamma_1 \vdash^{(m_1,e_1)} s : [N \multimap M] \qquad \Psi_2 \rhd_{\mathrm{cbv}} \ \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m_2+1,e_1+e_2)} t : M} \ \mathtt{app}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $m \coloneqq m_1 + m_2 + 1$ and $e \coloneqq e_1 + e_2$. By *i.h.* applied to Ψ_1, there is a derivation $\Psi \rhd_{\mathrm{CbV}} \ \Gamma_1 \vdash^{(m',e')} s' : [N \multimap M]$ where
- $m' \coloneqq m_1 - 1$ and $e' \coloneqq e_1$ if $\mathsf{r} = \mathtt{m}_{\mathrm{cbv}}$,
- $m' \coloneqq m_1$ and $e' \coloneqq e_1 - 1$ if $\mathsf{r} = \mathsf{e}_{\mathrm{cbv}}$.

Thus, we can construct the following derivation Φ'

$$\frac{\Psi \rhd_{\mathrm{cbv}} \ \Gamma_1 \vdash^{(m',e')} s' : [N \multimap M] \qquad \Psi_2 \rhd_{\mathrm{cbv}} \ \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m'+m_2+1,e'+e_2)} t' : M} \ \mathtt{app}$$

where
- $m' + m_2 + 1 = m_1 - 1 + m_2 + 1 = m - 1$ and $e' + e_2 = e_1 + e_2 = e$ if $\mathsf{r} = \mathtt{m}_{\mathrm{cbv}}$,
- $m' + m_2 + 1 = m_1 + m_2 + 1 = m$ and $e' + e_2 = e_1 - 1 + e_2 = e - 1$ if $\mathsf{r} = \mathsf{e}_{\mathrm{cbv}}$.

– *Application right, i.e.* $t \coloneqq us \to_{\mathsf{r}} us' =: t'$ with $s \to_{\mathsf{r}} s'$ and $\mathsf{r} \in \{\mathtt{m}_{\mathrm{cbv}}, \mathsf{e}_{\mathrm{cbv}}\}$. So, Φ has the form

$$\frac{\Psi_1 \rhd_{\mathrm{cbv}} \ \Gamma_1 \vdash^{(m_1,e_1)} u : [N \multimap M] \qquad \Psi_2 \rhd_{\mathrm{cbv}} \ \Gamma_2 \vdash^{(m_2,e_2)} s : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m_2+1,e_1+e_2)} t : M} \ \mathtt{app}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $m \coloneqq m_1 + m_2 + 1$ and $e \coloneqq e_1 + e_2$. By *i.h.* applied to Ψ_2, there is a derivation $\Psi \rhd_{\mathrm{CbV}} \ \Gamma_2 \vdash^{(m',e')} s' : N$ where
- $m' \coloneqq m_2 - 1$ and $e' \coloneqq e_2$ if $\mathsf{r} = \mathtt{m}_{\mathrm{cbv}}$,
- $m' \coloneqq m_2$ and $e' \coloneqq e_2 - 1$ if $\mathsf{r} = \mathsf{e}_{\mathrm{cbv}}$.

Thus, we can construct the following derivation Φ'

$$\frac{\Psi_1 \rhd_{\mathrm{cbv}} \ \Gamma_1 \vdash^{(m_1,e_1)} u : [N \multimap M] \qquad \Psi \rhd_{\mathrm{cbv}} \ \Gamma_2 \vdash^{(m',e')} s' : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m'+1,e_1+e')} t' : M} \ \mathtt{app}$$

where
- $m_1 + m' + 1 = m_1 + m_2 - 1 + 1 = m - 1$ and $e_1 + e' = e_1 + e_2 = e$ if $\mathsf{r} = \mathtt{m}_{\mathrm{cbv}}$,
- $m_1 + m' + 1 = m_1 + m_2 + 1 = m$ and $e_1 + e' = e_1 + e_2 - 1 = e - 1$ if $\mathsf{r} = \mathsf{e}_{\mathrm{cbv}}$.

– *Left explicit substitution, i.e.* $t \coloneqq s[x{\leftarrow}u] \to_{\mathsf{r}} s'[x{\leftarrow}u] =: t'$ with $s \to_{\mathsf{r}} s'$ and $\mathsf{r} \in \{\mathtt{m}_{\mathrm{cbv}}, \mathsf{e}_{\mathrm{cbv}}\}$. So, Φ has the form

$$\frac{\Gamma_1, x : N \vdash^{(m_1,e_1)} s : M \qquad \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m_2,e_1+e_2)} t : M} \ \mathtt{ES}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $m \coloneqq m_1 + m_2$ and $e \coloneqq e_1 + e_2$. By *i.h.* applied to Ψ_1, there is a derivation $\Psi \rhd_{\mathrm{CbV}} \ \Gamma_1, x : N \vdash^{(m',e')} s' : M$ where

- $m' := m_1 - 1$ and $e' := e_1$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m' := m_1$ and $e' := e_1 - 1$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$.

Thus, we construct the following derivation Φ'

$$\frac{\Psi \rhd_{\text{cbv}} \;\; \Gamma_1, x : N \vdash^{(m',e')} s' : M \qquad \Psi_2 \rhd_{\text{cbv}} \;\; \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m'+m_2,e'+e_2)} t' : M}\ \mathtt{ES}$$

where

- $m' + m_2 = m_1 - 1 + m_2 = m - 1$ and $e' + e_2 = e_1 + e_2 = e$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m' + m_2 = m_1 + m_2 = m$ and $e' + e_2 = e_1 - 1 + e_2 = e - 1$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$.

– *Right explicit substitution, i.e.* $t := s[x \leftarrow u] \rightarrow_{\mathsf{r}} s[x \leftarrow u'] =: t'$ with $u \rightarrow_{\mathsf{r}} u'$ and $\mathsf{r} \in \{\mathtt{m}_{\text{cbv}}, \mathsf{e}_{\text{cbv}}\}$. So, Φ has the form

$$\frac{\Gamma_1, x : N \vdash^{(m_1,e_1)} s : M \qquad \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m_2,e_1+e_2)} t : M}\ \mathtt{ES}$$

where $\Gamma := \Gamma_1 \uplus \Gamma_2$ and $m := m_1 + m_2$ and $e := e_1 + e_2$. By *i.h.* applied to Ψ_2, there is a derivation $\Psi \rhd_{\text{CbV}} \;\; \Gamma_2 \vdash^{(m',e')} u' : N$ where

- $m' := m_2 - 1$ and $e' := e_2$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m' := m_2$ and $e' := e_2 - 1$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$.

Thus, we can construct the following derivation Φ'

$$\frac{\Psi_1 \rhd_{\text{cbv}} \;\; \Gamma_1, x : N \vdash^{(m_1,e_1)} s : M \qquad \Psi \rhd_{\text{cbv}} \;\; \Gamma_2 \vdash^{(m',e')} u' : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m',e_1+e')} t' : M}\ \mathtt{ES}$$

where

- $m_1 + m' = m_1 + m_2 - 1 = m - 1$ and $e_1 + e' = e_1 + e_2 = e$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_1 + m' = m_1 + m_2 = m$ and $e_1 + e' = e_1 + e_2 - 1 = e - 1$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$. □

Proposition 20 (Tight typings for normal forms for CbV). *Let* $\Phi \rhd_{\text{cbv}} \Gamma \vdash^{(m,e)} t : \mathbf{0}$ *be a derivation, with* $\mathsf{normal}_{\text{cbv}}(t)$. *Then* Γ *is empty, and so* Φ *is tight, and* $m = e = 0$.

Proof. By induction on the derivation of $\mathsf{normal}_{\text{cbv}}(t)$. Cases:

– *Base, i.e.* $t := \lambda x.s$. Since $\Phi \rhd_{\text{cbv}} \;\; \Gamma \vdash^{(m,e)} t : \mathbf{0}$, the last rule of Φ can only be a 0-ary instance of many, thus $\mathtt{dom}(\Gamma) = \emptyset$ and $m = e = 0$.

– *Inductive step, i.e.* $t := s[x \leftarrow u]$ with $\mathsf{normal}_{\text{cbv}}(s)$ and $\mathsf{normal}_{\text{cbv}}(u)$. Hence, Φ has the form

$$\frac{\Psi \rhd_{\text{cbv}} \;\; \Pi, x : N \vdash^{(m',e')} s : \mathbf{0} \qquad \Theta \rhd_{\text{cbv}} \;\; \Delta \vdash^{(m'',e'')} u : N}{\Gamma \vdash^{(m,e)} t : \mathbf{0}}\ \mathtt{ES}$$

where $\Gamma := \Pi \uplus \Delta$ and $m := m' + m''$ and $e := e' + e''$. By *i.h.* applied to Ψ, $\mathtt{dom}(\Pi) = \emptyset$ and $N = \mathbf{0}$ and $m' = 0 = e'$. By *i.h.* applied to Θ (as $N = \mathbf{0}$), $\mathtt{dom}(\Delta) = \emptyset$ and $m'' = 0 = e'$. Therefore, $\mathtt{dom}(\Gamma) = \emptyset$ and $m = 0 = e$. □

Theorem 10 (CbV tight correctness). *Let t be a closed term. If $\Phi \triangleright_{\text{cbv}} \Gamma \vdash^{(m,e)} t : M$ then there is s such that $d\colon t \to^*_{\text{cbv}} s$, $\mathsf{normal}_{\text{cbv}}(s)$, $|d|_{\mathtt{m}} \leq m$, $|d|_{\mathtt{e}} \leq e$. Moreover, if Φ is tight then $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$.*

Proof. By induction on $m + e$ and case analysis on whether t reduces or not.

If $\mathsf{normal}_{\text{cbv}}(t)$ then the statement holds with $s \coloneqq t$ and d the empty evaluation, so that $|d|_{\mathtt{m}} = 0 = |d|_{\mathtt{e}}$. If moreover Φ is tight then $|d|_{\mathtt{m}} = 0 = m$ and $|d|_{\mathtt{e}} = 0 = e$ by Proposition 7.

Otherwise $\neg\mathsf{normal}_{\text{cbv}}(t)$, then $t \to_{\text{cbv}} u$ according to the syntactic characterization of closed cbv-normal forms (Proposition 1), since t is closed. As $t \to_{\text{cbv}} u$ means either $t \to_{\mathtt{m}_{\text{cbv}}} u$ or $t \to_{\mathtt{e}_{\text{cbv}}} u$, by quantitative subject reduction (Proposition 6) there is $\Psi \triangleright_{\text{cbv}} \Gamma \vdash^{(m',e')} u : M$ with:

- $m' \coloneqq m - 1$ and $e' = e$ if $t \to_{\mathtt{m}_{\text{cbv}}} u$,
- $m' \coloneqq m$ and $e' = e - 1$ if $t \to_{\mathtt{e}_{\text{cbv}}} u$.

By *i.h.* (since $m' + e' = m + e - 1$), there is a term s such that $d'\colon u \to^*_{\text{cbv}} s$ and $\mathsf{normal}_{\text{cbv}}(s)$, with $|d'|_{\mathtt{m}} \leq m'$ and $|d'|_{\mathtt{e}} \leq e'$; and if, moreover, Ψ is tight, then $|d'|_{\mathtt{m}} = m'$ and $|d'|_{\mathtt{e}} = e'$. The evaluation $d\colon t \to^*_{\text{cbv}} s$ obtained by prefixing d' with the step $t \to_{\text{cbv}} u$ verifies $|d|_{\mathtt{m}} \leq m$ and $|d|_{\mathtt{e}} \leq e$ (and $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$ if moreover Φ—and hence Ψ since $\mathtt{dom}(\Gamma) = \emptyset$ and $M = \mathbf{0}$—is tight) because:

- if $t \to_{\mathtt{m}_{\text{cbv}}} u$ then $|d|_{\mathtt{m}} = |d'|_{\mathtt{m}} + 1 \leq m' + 1 = m$ and $|d|_{\mathtt{e}} = |d'|_{\mathtt{e}} \leq e' = e$ (and $|d|_{\mathtt{m}} = |d'|_{\mathtt{m}} + 1 = m' + 1 = m$ and $|d|_{\mathtt{e}} = |d'|_{\mathtt{e}} = e' = e$ if moreover Φ is tight),
- if $t \to_{\mathtt{e}_{\text{cbv}}} u$ then $|d|_{\mathtt{m}} = |d'|_{\mathtt{m}} \leq m' = m$ and $|d|_{\mathtt{e}} = |d'|_{\mathtt{e}} + 1 \leq e' + 1 = e$ (and $|d|_{\mathtt{m}} = |d'|_{\mathtt{m}} = m' = m$ and $|d|_{\mathtt{e}} = |d'|_{\mathtt{e}} + 1 = e' + 1 = e$ if moreover Φ is tight). □

C.2 CbV Completeness

Proposition 21 (Normal forms are tightly typable for CbV). *Let t be such that $\mathsf{normal}_{\text{cbv}}(t)$. Then there exists a tight derivation $\Phi \triangleright_{\text{cbv}} \vdash^{(0,0)} t : \mathbf{0}$.*

Proof. By induction on $\mathsf{normal}_{\text{cbv}}(t)$. Cases:

- *Abstraction*: if $\mathsf{normal}_{\text{cbv}}(t)$ because $t = \lambda x.s$ then Φ is given by

$$\frac{}{\vdash^{(0,0)} \lambda x.s : \mathbf{0}}\ \mathsf{many}$$

- *Substitution*: if $\mathsf{normal}_{\text{cbv}}(t)$ because $t = s[x{\leftarrow}u]$ with $\mathsf{normal}_{\text{cbv}}(s)$ and $\mathsf{normal}_{\text{cbv}}(u)$ then by *i.h.* there exist tight derivations $\Psi \triangleright_{\text{cbv}} \vdash^{(0,0)} s : \mathbf{0}$ and $\Theta \triangleright_{\text{cbv}} \vdash^{(0,0)} u : \mathbf{0}$. Then Φ is given by:

$$\frac{\Psi \triangleright \ \vdash^{(0,0)} s : \mathbf{0} \qquad \Theta \triangleright \ \vdash^{(0,0)} u : \mathbf{0}}{\vdash^{(0,0)} s[x{\leftarrow}u] : \mathbf{0}}\ \mathtt{ES}$$

□

Lemma 15 (Typability of values). *Let v be a value.*

1. Empty judgement*: There is a derivation* $\Psi \triangleright_{\text{cbv}} \ \vdash^{(0,0)} v : \mathbf{0}$.
2. Multi-set merging*: If there are derivations* $\Phi \triangleright_{\text{cbv}} \ \Gamma \vdash^{(m,e)} v : M$ *and* $\Psi \triangleright_{\text{cbv}} \ \Pi \vdash^{(m',e')} v : N$*, then there is a derivation* $\Theta \triangleright_{\text{cbv}} \ \Gamma \uplus \Pi \vdash^{(m'',e'')} v : M \uplus N$ *with* $m'' = m + m'$ *and* $e'' = e + e'$.

Lemma 16 (Linear removal for CbV). *Let* $\Phi \triangleright_{\text{cbv}} \ \Gamma, x : M \vdash^{(m,e)} V\langle\!\langle v \rangle\!\rangle : N$ *where* $x \notin \mathtt{fv}(v)$*. Then, there exist*

- *a multi type* M' *and two type contexts* Γ' *and* Π*,*
- *a derivation* $\Psi \triangleright_{\text{cbv}} \ x : M \uplus M', \Pi \vdash^{(m'',e'')} V\langle\!\langle x \rangle\!\rangle : N$*, and*
- *a derivation* $\Phi' \triangleright_{\text{cbv}} \ \Gamma' \vdash^{(m',e')} v : M'$

such that

- Type contexts: $\Gamma = \Gamma' \uplus \Pi$*,*
- Indices: $(m, e) = (m' + m'', e' + e'' - 1)$*.*

Proof. By induction on V. Cases:

- *Hole, i.e.* $V \coloneqq \langle\cdot\rangle$: then, $V\langle v \rangle = v$ and $V\langle x \rangle = x$. Since $x \notin \mathtt{fv}(v)$, then $M = \mathbf{0}$ according to Lemma 12. Let $M' \coloneqq N$ and $\Gamma' \coloneqq \Gamma$ and Π be such that $\mathtt{dom}(\Pi) = \emptyset$: hence, $\Gamma = \Gamma' \uplus \Pi$. Let $\Phi' \coloneqq \Phi$ and Ψ be the following derivation
$$\frac{}{x : N \vdash^{(0,1)} x : N}\ \mathsf{ax}$$
Thus, $\Psi \triangleright_{\text{cbv}} \ \Pi, x : M \uplus M' \vdash^{(m'',e'')} V\langle\!\langle x \rangle\!\rangle : N$ with $(m'', e'') \coloneqq (0, 1)$, because $\Pi, x : M \uplus M' = x : M' = x : N$; and $\Phi' \triangleright_{\text{cbv}} \ \Gamma' \vdash^{(m,e)} v : M$ because $\Gamma, x : M = \Gamma'$. Moreover, $(m, e) = (m + 0, e + 1 - 1) = (m + m'', e + e'' - 1)$.
- *Left application, i.e.* $V = V't$: then, $V\langle\!\langle x \rangle\!\rangle = V'\langle\!\langle x \rangle\!\rangle t$ and $V\langle\!\langle v \rangle\!\rangle = V'\langle\!\langle v \rangle\!\rangle t$. So, Φ has the form
$$\frac{\Phi_1 \triangleright_{\text{cbv}} \ \Gamma_1, x : M_1 \vdash^{(m_1,e_1)} V'\langle\!\langle v \rangle\!\rangle : [O \multimap N] \qquad \Phi_2 \triangleright_{\text{cbv}} \ \Gamma_2, x : M_2 \vdash^{(m_2,e_2)} t : O}{\Gamma, x : M \vdash^{(m,e)} V'\langle\!\langle v \rangle\!\rangle t : N}\ \mathsf{app}$$
where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $M \coloneqq M_1 \uplus M_2$ and $(m, e) \coloneqq (m_1 + m_2 + 1, e_1 + e_2)$. By *i.h.*, there exist a multi type M', two type contexts Γ_1' and Π_1, and two derivations $\Psi_1 \triangleright_{\text{cbv}} \ \Pi_1, x : M_1 \uplus M' \vdash^{(m_1'',e_1'')} V'\langle\!\langle x \rangle\!\rangle : [O \multimap N]$ and $\Phi' \triangleright_{\text{cbv}} \ \Gamma' \vdash^{(m',e')} v : M'$ such that $\Gamma_1 = \Gamma' \uplus \Pi_1$ ad $(m_1, e_1) = (m' + m_1'', e' + e_1'' - 1)$. We can then construct the following derivation Ψ
$$\frac{\Psi_1 \triangleright_{\text{cbv}} \ \Pi_1, x : M_1 \uplus M' \vdash^{(m_1'',e_1'')} V'\langle\!\langle x \rangle\!\rangle : [O \multimap N] \qquad \Phi_2 \triangleright_{\text{cbv}} \ \Gamma_2, x : M_2 \vdash^{(m_2,e_2)} t : O}{\Pi_1 \uplus \Gamma_2, x : M_1 \uplus M' \uplus M_2 \vdash^{(m_1''+m_2+1, e_1''+e_2)} V'\langle\!\langle x \rangle\!\rangle t : N}\ \mathsf{app}$$
where $M_1 \uplus M' \uplus M_2 = M \uplus M'$. If we set $\Pi \coloneqq \Pi_1 \uplus \Gamma_2$ and $(m'', e'') \coloneqq (m_1'' + m_2 + 1, e_1'' + e_2)$, then we have $\Gamma' \uplus \Pi = \Gamma' \uplus \Pi_1 \uplus \Gamma_2 = \Gamma_1 \uplus \Gamma_2 = \Gamma$ and $(m' + m'', e' + e'' - 1) = (m' + m_1'' + m_2 + 1, e' + e_1'' + e_2 - 1) = (m_1 + m_2 + 1, e_1 + e_2) = (m, e)$, as required.

– *Right application*, *i.e.* $V = tV'$: then, $V\langle\!\langle x\rangle\!\rangle = tV'\langle\!\langle x\rangle\!\rangle$ and $V\langle\!\langle v\rangle\!\rangle = tV'\langle\!\langle v\rangle\!\rangle$. So, Φ has the form

$$\dfrac{\Phi_1 \triangleright_{\mathrm{cbv}} \ \Gamma_1, x\!:\!M_1 \vdash^{(m_1,e_1)} t : [O \multimap N] \qquad \Phi_2 \triangleright_{\mathrm{cbv}} \ \Gamma_2, x\!:\!M_2 \vdash^{(m_2,e_2)} V'\langle\!\langle v\rangle\!\rangle : O}{\Gamma, x\!:\!M \vdash^{(m,e)} tV'\langle\!\langle v\rangle\!\rangle : N}\ \texttt{app}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $M \coloneqq M_1 \uplus M_2$ and $(m,e) \coloneqq (m_1 + m_2 + 1, e_1 + e_2)$. By *i.h.*, there exist a multi type M', two type contexts Γ_2' and Π_2, and two derivations $\Psi_2 \triangleright_{\mathrm{cbv}} \ \Pi_2, x\!:\!M_2 \uplus M' \vdash^{(m_2'',e_2'')} V'\langle\!\langle x\rangle\!\rangle : O$ and $\Phi' \triangleright_{\mathrm{cbv}} \ \Gamma' \vdash^{(m',e')} v : M'$ such that $\Gamma_2 = \Gamma' \uplus \Pi_2$ ad $(m_2, e_2) = (m' + m_2'', e' + e_2'' - 1)$. We can then construct the following derivation Ψ

$$\dfrac{\Phi_1 \triangleright_{\mathrm{cbv}} \ \Gamma_1, x\!:\!M_1 \vdash^{(m_1,e_1)} t : [O \multimap N] \qquad \Psi_2 \triangleright_{\mathrm{cbv}} \ \Pi_2, x\!:\!M_2 \uplus M' \vdash^{(m_2'',e_2'')} V'\langle\!\langle x\rangle\!\rangle : O}{\Pi_2 \uplus \Gamma_1, x\!:\!M_2 \uplus M' \uplus M_1 \vdash^{(m_2''+m_1+1, e_2''+e_1)} V'\langle\!\langle x\rangle\!\rangle t : N}\ \texttt{app}$$

where $M_1 \uplus M' \uplus M_2 = M \uplus M'$. If we set $\Pi \coloneqq \Pi_2 \uplus \Gamma_1$ and $(m'', e'') \coloneqq (m_2'' + m_1 + 1, e_2'' + e_1)$, then we have $\Gamma' \uplus \Pi = \Gamma' \uplus \Pi_2 \uplus \Gamma_1 = \Gamma_1 \uplus \Gamma_2 = \Gamma$ and $(m' + m'', e' + e'' - 1) = (m' + m_2'' + m_1 + 1, e' + e_2'' + e_1 - 1) = (m_1 + m_2 + 1, e_1 + e_2) = (m, e)$, as required.

– *Left explicit substitution*, *i.e.* $V = V'[y{\leftarrow}t]$: then, $V\langle\!\langle x\rangle\!\rangle = V'\langle\!\langle x\rangle\!\rangle[y{\leftarrow}t]$ and $V\langle\!\langle v\rangle\!\rangle = V'\langle\!\langle v\rangle\!\rangle[y{\leftarrow}t]$ where $y \notin \mathtt{fv}(v) \cup \{x\}$. We can suppose without loss of generality that $y \notin \mathtt{fv}(t)$. So, Φ has the form

$$\dfrac{\Phi_1 \triangleright_{\mathrm{cbv}} \ \Gamma_1, x\!:\!M_1, y\!:\!O \vdash^{(m_1,e_1)} V'\langle\!\langle v\rangle\!\rangle : N \qquad \Phi_2 \triangleright_{\mathrm{cbv}} \ \Gamma_2, x\!:\!M_2 \vdash^{(m_2,e_2)} t : O}{\Gamma, x\!:\!M \vdash^{(m,e)} V'\langle\!\langle v\rangle\!\rangle[y{\leftarrow}t] : N}\ \texttt{ES}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $M \coloneqq M_1 \uplus M_2$ and $(m,e) \coloneqq (m_1 + m_2, e_1 + e_2)$. By *i.h.*, there exist a multi type M', two type contexts Γ_1' and Π_1, and two derivations $\Psi_1 \triangleright_{\mathrm{cbv}} \ \Pi_1, x\!:\!M_1 \uplus M', y\!:\!O \vdash^{(m_1'',e_1'')} V'\langle\!\langle x\rangle\!\rangle : N$ and $\Phi' \triangleright_{\mathrm{cbv}} \ \Gamma' \vdash^{(m',e')} v : M'$ such that $\Gamma_1 = \Gamma' \uplus \Pi_1$ ad $(m_1, e_1) = (m' + m_1'', e' + e_1'' - 1)$ (note that $y \notin \mathtt{dom}(\Gamma')$ because of Lemma 12, since $y \notin \mathtt{fv}(v)$). We can then construct the following derivation Ψ

$$\dfrac{\Psi_1 \triangleright_{\mathrm{cbv}} \ \Pi_1, x\!:\!M_1 \uplus M', y\!:\!O \vdash^{(m_1'',e_1'')} V'\langle\!\langle x\rangle\!\rangle : N \qquad \Phi_2 \triangleright_{\mathrm{cbv}} \ \Gamma_2, x\!:\!M_2 \vdash^{(m_2,e_2)} t : O}{\Pi_1 \uplus \Gamma_2, x\!:\!M_1 \uplus M' \uplus M_2 \vdash^{(m_1''+m_2, e_1''+e_2)} V'\langle\!\langle x\rangle\!\rangle t : N}\ \texttt{ES}$$

where $M_1 \uplus M' \uplus M_2 = M \uplus M'$. If we set $\Pi \coloneqq \Pi_1 \uplus \Gamma_2$ and $(m'', e'') \coloneqq (m_1'' + m_2, e_1'' + e_2)$, then we have $\Gamma' \uplus \Pi = \Gamma' \uplus \Pi_1 \uplus \Gamma_2 = \Gamma_1 \uplus \Gamma_2 = \Gamma$ and $(m' + m'', e' + e'' - 1) = (m' + m_1'' + m_2, e' + e_1'' + e_2 - 1) = (m_1 + m_2, e_1 + e_2) = (m, e)$, as required.

– *Right explicit substitution*, *i.e.* $V = t[y{\leftarrow}V']$: then, $V\langle\!\langle x\rangle\!\rangle = t[y{\leftarrow}V'\langle\!\langle x\rangle\!\rangle]$ and $V\langle\!\langle v\rangle\!\rangle = t[y{\leftarrow}V'\langle\!\langle v\rangle\!\rangle]$. So, Φ has the form

$$\dfrac{\Phi_1 \triangleright_{\mathrm{cbv}} \ \Gamma_1, x\!:\!M_1, y\!:\!O \vdash^{(m_1,e_1)} V''\langle\!\langle y\rangle\!\rangle : N \qquad \Phi_2 \triangleright_{\mathrm{cbv}} \ \Gamma_2, x\!:\!M_2 \vdash^{(m_2,e_2)} V'\langle\!\langle v\rangle\!\rangle : O}{\Gamma, x\!:\!M \vdash^{(m,e)} t[y{\leftarrow}V'\langle\!\langle v\rangle\!\rangle] : N}\ \texttt{ES}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $M \coloneqq M_1 \uplus M_2$ and $(m,e) \coloneqq (m_1+m_2, e_1+e_2)$. By *i.h.*, there exist a multi type M', two type contexts Γ'_2 and Π_2, and two derivations $\Psi_2 \triangleright_{\mathrm{cbv}} \Pi_2, x:M_2 \uplus M', y:O \vdash^{(m''_2,e''_2)} V'\langle\langle x\rangle\rangle : O$ and $\Phi' \triangleright_{\mathrm{cbv}} \Gamma' \vdash^{(m',e')} v : M'$ such that $\Gamma_2 = \Gamma' \uplus \Pi_2$ ad $(m_2,e_2) = (m'+m''_2, e'+e''_2-1)$ (note that $y \notin \mathtt{dom}(\Gamma')$ because of Lemma 12, since $y \notin \mathtt{fv}(v)$). We can then construct the following derivation Ψ

$$\dfrac{\Phi_1 \triangleright_{\mathrm{cbv}} \Gamma_1, x:M_1, y:O \vdash^{(m_1,e_1)} t:N \qquad \Psi_2 \triangleright_{\mathrm{cbv}} \Pi_2, x:M_2 \uplus M' \vdash^{(m''_2,e''_2)} V'\langle\langle x\rangle\rangle : O}{\Pi_2 \uplus \Gamma_1, x:M_2 \uplus M' \uplus M_1 \vdash^{(m''_2+m_1+1, e''_2+e_1)} t[y\leftarrow V'\langle\langle x\rangle\rangle] : N}\ \texttt{ES}$$

where $M_1 \uplus M' \uplus M_2 = M \uplus M'$. If we set $\Pi \coloneqq \Pi_2 \uplus \Gamma_1$ and $(m'',e'') \coloneqq (m''_2+m_1, e''_2+e_1)$, then we have $\Gamma' \uplus \Pi = \Gamma' \uplus \Pi_2 \uplus \Gamma_1 = \Gamma_1 \uplus \Gamma_2 = \Gamma$ and $(m'+m'', e'+e''-1) = (m'+m''_2+m_1, e'+e''_2+e_1-1) = (m_1+m_2, e_1+e_2) = (m,e)$, as required. □

Proposition 22 (Quantitative subject expansion for CbV). *Let $\Phi' \triangleright_{\mathrm{cbv}} \Gamma \vdash^{(m,e)} t' : M$ be a derivation.*

1. Multiplicative*: if $t \to_{\mathtt{m}_{\mathrm{cbv}}} t'$ then there is a derivation $\Phi \triangleright_{\mathrm{cbv}} \Gamma \vdash^{(m+1,e)} t : M$.*
2. Exponential*: if $t \to_{\mathtt{e}_{\mathrm{cbv}}} t'$ then there is a derivation $\Phi \triangleright_{\mathrm{cbv}} \Gamma \vdash^{(m,e+1)} t : M$.*

Proof. By induction on the reduction relation $\to_{\mathrm{cbv}}$, with the root rules $\mapsto_{\mathtt{m}}$ and $\mapsto_{\mathtt{e}_{\mathrm{cbv}}}$ as the base case, and the closure by CbV contexts of $\mapsto_{\mathrm{cbv}} \coloneqq \mapsto_{\mathtt{m}} \cup \mapsto_{\mathtt{e}_{\mathrm{cbv}}}$ as the inductive one.

– *Root step for* $\to_{\mathtt{m}_{\mathrm{cbv}}}$ *i.e.* $t \coloneqq S\langle\lambda x.u\rangle r \mapsto_{\mathtt{m}} S\langle u[x\leftarrow r]\rangle \eqqcolon t'$ where $S \coloneqq [y_1\leftarrow s_1]\ldots[y_n\leftarrow s_n]$ for some $n \geq 0$. We proceed by induction on $n \in \mathbb{N}$.
If $n = 0$ then $S = \langle\cdot\rangle$ and so $t = S\langle\lambda x.u\rangle r = (\lambda x.u)r$ and $t' = S\langle u[x\leftarrow r]\rangle = u[x\leftarrow r]$. Hence, Φ' has the form

$$\dfrac{\Psi \triangleright_{\mathrm{cbv}} \Pi, x:O \vdash^{(m',e')} u:M \qquad \Theta \triangleright_{\mathrm{cbv}} \Delta \vdash^{(m'',e'')} r:O}{\Gamma \vdash^{(m'+m'',e'+e'')} u[x\leftarrow r] : M}\ \texttt{ES}$$

where $\Gamma \coloneqq \Pi \uplus \Delta$ and $m \coloneqq m'+m''$ and $e \coloneqq e'+e''$. We can then construct the following typing derivation Φ:

$$\dfrac{\dfrac{\dfrac{\Psi \triangleright_{\mathrm{cbv}} \Pi, x:O \vdash^{(m',e')} u:M}{\Pi \vdash^{(m',e')} \lambda x.u : O \multimap M}\ \texttt{fun}}{\Pi \vdash^{(m',e')} \lambda x.u : [O \multimap M]}\ \texttt{many} \qquad \Theta \triangleright_{\mathrm{cbv}} \Delta \vdash^{(m'',e'')} r:O}{\Gamma \vdash^{(1+m'+m'',e'+e'')} (\lambda x.u)r : M}\ \texttt{app}$$

where $(1+m'+m'', e'+e'') = (m+1, e)$.
Suppose now $n > 0$. Let $S' \coloneqq [y_1\leftarrow s_1]\ldots[y_{n-1}\leftarrow s_{n-1}]$: then, $t = S\langle\lambda x.u\rangle r = S'\langle\lambda x.u\rangle[y_n\leftarrow s_n]r$ and $t' = S\langle u[x\leftarrow r]\rangle = S'\langle u[x\leftarrow r]\rangle[y_n\leftarrow s_n]$. Hence, Φ' has the form

$$\dfrac{\Psi' \triangleright_{\mathrm{cbv}} \Gamma', y_n:N_n \vdash^{(m',e')} S'\langle u[x\leftarrow r]\rangle : M \qquad \Psi_n \triangleright_{\mathrm{cbv}} \Gamma_n \vdash^{(m_n,e_n)} s_n : N_n}{\Gamma \vdash^{(m,e)} S'\langle u[x\leftarrow r]\rangle[y_n\leftarrow s_n] : M}\ \texttt{ES}$$

where $\Gamma \coloneqq \Gamma' \uplus \Gamma_n$ and $(m, e) \coloneqq (m' + m_n, e' + e_n)$. By *i.h.* applied to Ψ' (since $S'\langle \lambda x.u\rangle r \mapsto_{\mathtt{m}} S'\langle u[x\leftarrow r]\rangle$), there exists a derivation with conclusion $\Gamma', y_n : N_n \vdash^{(m'+1,e')} S'\langle \lambda x.u\rangle r : M$, which necessarily has the form (as $y_n \notin \mathtt{dom}(\Gamma'_0)$ by Lemma 12, since $y_n \notin \mathtt{fv}(r)$)

$$\dfrac{\Psi \triangleright_{\mathrm{cbv}} \ \Pi, x : O, y_n : N_n \vdash^{(m'',e'')} S'\langle \lambda x.u\rangle : M \qquad \Theta \triangleright_{\mathrm{cbv}} \ \Gamma'_0 \vdash^{(m'_0,e'_0)} r : O}{\Gamma', y_n : N_n \vdash^{(m',e')} S'\langle \lambda x.u\rangle r : M}\ \mathtt{ES}$$

where $\Gamma' \coloneqq \Pi \uplus \Gamma'_0$ and $(m', e') = (m'' + m'_0, e'' + e'_0)$. Therefore, we can construct the following derivation Φ:

$$\dfrac{\dfrac{\Psi \triangleright_{\mathrm{cbv}} \ \Pi, x : O, y_n : N_n \vdash^{(m'',e'')} S'\langle \lambda x.u\rangle : M \qquad \Psi_n \triangleright_{\mathrm{cbv}} \ \Gamma_n \vdash^{(m_n,e_n)} s_n : N_n}{\Pi \uplus \Gamma_n, x : O \vdash^{(m''+m_n,e''+e_n)} S\langle \lambda x.u\rangle : M}\ \mathtt{ES} \qquad \Theta \triangleright_{\mathrm{cbv}} \ \Gamma'_0 \vdash^{(m'_0,e'_0)} r : O}{\Pi \uplus \Gamma_n \uplus \Gamma'_0 \vdash^{(m''+m_n+m'_0+1,e''+e_n+e'_0)} S\langle \lambda x.u\rangle r : M}\ \mathtt{app}$$

where $\Pi \uplus \Gamma_n \uplus \Gamma'_0 = \Gamma' \uplus \Gamma_n = \Gamma$ and $(m'' + m_n + m'_0 + 1, e'' + e_n + e'_0) = (m' + m_n + 1, e' + e_n + 1) = (m + 1, e)$.

– *Root step for* $\to_{\mathtt{e}_{\mathrm{cbv}}}$ *i.e.* $t \coloneqq V\langle\langle x\rangle\rangle[x\leftarrow S\langle v\rangle] \mapsto_{\mathtt{e}_{\mathrm{cbv}}} S\langle V\langle\langle v\rangle\rangle[x\leftarrow v]\rangle \eqqcolon t'$ with $S \coloneqq [y_1\leftarrow s_1] \dots [y_n\leftarrow s_n]$ for some $n \geq 0$. We proceed by induction on $n \in \mathbb{N}$. If $n = 0$ then $S = \langle\cdot\rangle$ and so $t = S\langle v\rangle = v$ and $t' = S\langle V\langle\langle v\rangle\rangle[x\leftarrow v]\rangle = V\langle\langle v\rangle\rangle[x\leftarrow v]$. Hence, Φ' has the form

$$\dfrac{\Psi_0 \triangleright_{\mathrm{cbv}} \ \Gamma_0, x : N \vdash^{(m_0,e_0)} V\langle\langle v\rangle\rangle : M \qquad \Theta_1 \triangleright_{\mathrm{cbv}} \ \Gamma_1 \vdash^{(m_1,e_1)} v : N}{\Gamma \vdash^{(m,e)} V\langle\langle v\rangle\rangle[x\leftarrow v] : M}\ \mathtt{ES}$$

where $\Gamma \coloneqq \Gamma_0 \uplus \Gamma_1$, and $m \coloneqq m_0 + m_1$ and $e \coloneqq e_0 + e_1$. By linear removal (Lemma 5), there are a multi type N', two type contexts Γ'_0 and Π and two derivations $\Theta' \triangleright_{\mathrm{cbv}} \Gamma'_0 \vdash^{(m'_0,m'_0)} v : N'$ and $\Psi \triangleright_{\mathrm{cbv}} \Pi, x : N \uplus N' \vdash^{(m'',e'')} V\langle\langle v\rangle\rangle : M$ such that $\Gamma_0 = \Gamma'_0 \uplus \Pi$ and $(m_0, e_0) = (m'_0 + m'', e'_0 + e'' - 1)$. Note that $x \notin \mathtt{dom}(\Gamma'_0)$ by Lemma 12, since $x \notin \mathtt{fv}(v)$. By merging of multitypes (Lemma 15.2), there is a derivation $\Theta \triangleright_{\mathrm{cbv}} \ \Gamma'_0 \uplus \Gamma_1 \vdash^{(m'_0+m_1,e'_0+e_1)} v : N \uplus N'$. We can construct the following derivation Φ:

$$\dfrac{\Psi \triangleright_{\mathrm{cbv}} \ \Pi, x : N \uplus N' \vdash^{(m'',e'')} V\langle\langle x\rangle\rangle : M \qquad \Theta \triangleright_{\mathrm{cbv}} \ \Gamma'_0 \uplus \Gamma_1 \vdash^{(m'_0+m_1,e'_0+e_1)} v : N \uplus N'}{\Pi \uplus \Gamma'_0 \uplus \Gamma_1 \vdash^{(m''+m'_0+m_1,e''+e'_0+e_1)} V\langle\langle x\rangle\rangle[x\leftarrow S\langle v\rangle] : M}\ \mathtt{ES}$$

where $\Pi \uplus \Gamma'_0 \uplus \Gamma_1 = \Gamma_0 \uplus \Gamma_1 = \Gamma$ and $(m'' + m'_0 + m_1, e'' + e'_0 + e_1) = (m_0 + m_1, e_0 + 1 + e_1) = (m, e + 1)$.
Suppose now $n > 0$. Let $S' \coloneqq [y_1\leftarrow s_1] \dots [y_{n-1}\leftarrow s_{n-1}]$: then, $t = S\langle v\rangle = S'\langle v\rangle[y_n\leftarrow s_n]$ and $t' = S\langle V\langle\langle v\rangle\rangle[x\leftarrow v]\rangle = S'\langle V\langle\langle v\rangle\rangle[x\leftarrow v]\rangle[y_n\leftarrow s_n]$. Hence, Φ' has the form

$$\dfrac{\Psi' \triangleright_{\mathrm{cbv}} \ \Gamma_0, y_n : N_n \vdash^{(m_0,e_0)} S'\langle V\langle\langle v\rangle\rangle[x\leftarrow v]\rangle : M \qquad \Theta_n \triangleright_{\mathrm{cbv}} \ \Gamma_n \vdash^{(m_n,e_n)} s_n : N_n}{\Gamma \vdash^{(m,e)} S\langle V\langle\langle v\rangle\rangle[x\leftarrow v]\rangle : M}\ \mathtt{ES}$$

where $\Gamma \coloneqq \Gamma_0 \uplus \Gamma_n$ and $(m, e) = (m_0 + m_n, e_0 + e_n)$. By *i.h.* applied to Ψ' (since $V\langle\langle x\rangle\rangle[x\leftarrow S'\langle v\rangle] \mapsto_{\mathtt{e}_{\mathrm{cbv}}} S'\langle V\langle\langle v\rangle\rangle[x\leftarrow v]\rangle$), there exists a derivation with conclusion $\Gamma_0, y_n : N_n \vdash^{(m_0,e_0+1)} V\langle\langle x\rangle\rangle[x{\leftarrow}S'\langle v\rangle] : M$, which necessarily has the form (as $y_n \notin \mathtt{dom}(\Gamma_0')$ by Lemma 12, since $y_n \notin \mathtt{fv}(V\langle\langle x\rangle\rangle)$)

$$\dfrac{\Psi \rhd_{\text{cbv}} \ \Gamma_0', x : N \vdash^{(m_0',e_0')} V\langle\langle x\rangle\rangle : M \qquad \Psi'' \rhd_{\text{cbv}} \ y_n : N_n, \Gamma_0'' \vdash^{(m_0'',e_0'')} S'\langle v\rangle : N}{\Gamma_0, y_n : N_n \vdash^{(m_0,e_0+1)} V\langle\langle x\rangle\rangle[x{\leftarrow}S'\langle v\rangle] : M} \ \mathtt{ES}$$

where $\Gamma_0 = \Gamma_0' \uplus \Gamma_0''$ and $(m_0, e_0+1) = (m_0' + m_0'', e_0' + e_0'')$. Therefore, we can construct the following derivation Φ:

$$\dfrac{\Psi \rhd_{\text{cbv}} \ \Gamma_0', x : N \vdash^{(m_0',e_0')} V\langle\langle x\rangle\rangle : M \qquad \dfrac{\Psi'' \rhd_{\text{cbv}} y_n : N_n, \Gamma_0'' \vdash^{(m_0'',e_0'')} S'\langle v\rangle : N \quad \Theta_n \rhd_{\text{cbv}} \Gamma_n \vdash^{(m_n,e_n)} s_n : N_n}{\Gamma_0'' \uplus \Gamma_n \vdash^{(m_0''+m_n, e_0''+e_n)} S\langle v\rangle : N} \ \mathtt{ES}}{\Gamma_0' \uplus \Gamma_0'' \uplus \Gamma_n \vdash^{(m_0'+m''+m_n, e_0'+e''+e_n)} V\langle\langle x\rangle\rangle[x{\leftarrow}S\langle v\rangle] : M} \ \mathtt{ES}$$

where $\Gamma_0' \uplus \Gamma_0'' \uplus \Gamma_n = \Gamma_0 \uplus \Gamma_n = \Gamma$ and $(m_0' + m_0'' + m_n, e_0' + e_0'' + e_n) = (m_0 + m_n, e_0 + 1 + e_n) = (m, e+1)$.

– *Application left, i.e.* $t \coloneqq su \to_{\mathsf{r}} s'u =: t'$ with $s \to_{\mathsf{r}} s'$ and $\mathsf{r} \in \{\mathtt{m}_{\text{cbv}}, \mathtt{e}_{\text{cbv}}\}$. So, Φ' has the form

$$\dfrac{\Psi_1 \rhd_{\text{cbv}} \ \Gamma_1 \vdash^{(m',e')} s' : [N \multimap M] \qquad \Psi_2 \rhd_{\text{cbv}} \ \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m'+m_2+1, e'+e_2)} t' : M} \ \mathtt{app}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $m \coloneqq m' + m_2 + 1$ and $e \coloneqq e' + e_2$. By *i.h.* applied to Ψ_1, there is a derivation $\Psi \rhd_{\text{cbv}} \ \Gamma_1 \vdash^{(m_1,e_1)} s : [N \multimap M]$ where
- $m_1 \coloneqq m' + 1$ and $e_1 \coloneqq e'$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_1 \coloneqq m_1$ and $e_1 \coloneqq e' + 1$ if $\mathsf{r} = \mathtt{e}_{\text{cbv}}$.

Thus, we have the derivation Φ

$$\dfrac{\Psi \rhd_{\text{cbv}} \ \Gamma_1 \vdash^{(m_1,e_1)} s : [N \multimap M] \qquad \Psi_2 \rhd_{\text{cbv}} \ \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m_2+1, e_1+e_2)} t : M} \ \mathtt{app}$$

where
- $m_1 + m_2 + 1 = m' + 1 + m_2 + 1 = m + 1$ and $e_1 + e_2 = e' + e_2 = e$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_1 + m_2 + 1 = m' + m_2 + 1 = m$ and $e_1 + e_2 = e' + 1 + e_2 = e + 1$ if $\mathsf{r} = \mathtt{e}_{\text{cbv}}$.

– *Application right, i.e.* $t \coloneqq us \to_{\mathsf{r}} us' =: t'$ with $s \to_{\mathsf{r}} s'$ and $\mathsf{r} \in \{\mathtt{m}_{\text{cbv}}, \mathtt{e}_{\text{cbv}}\}$. So, Φ' has the form

$$\dfrac{\Psi_1 \rhd_{\text{cbv}} \ \Gamma_1 \vdash^{(m_1,e_1)} u : [N \multimap M] \qquad \Psi_2 \rhd_{\text{cbv}} \ \Gamma_2 \vdash^{(m',e')} s' : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m'+1, e_1+e')} t' : M} \ \mathtt{app}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $m \coloneqq m_1 + m' + 1$ and $e \coloneqq e_1 + e'$. By *i.h.* applied to d_2', there is a derivation $\Psi \rhd_{\text{cbv}} \ \Gamma_2 \vdash^{(m_2,e_2)} s : N$ where
- $m_2 \coloneqq m' + 1$ and $e_2 \coloneqq e'$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_2 \coloneqq m'$ and $e_2 \coloneqq e' + 1$ if $\mathsf{r} = \mathtt{e}_{\text{cbv}}$.

Thus, we have the derivation Φ

$$\dfrac{\Psi_1 \rhd_{\text{cbv}} \; \Gamma_1 \vdash^{(m_1,e_1)} u : [N \multimap M] \qquad \Psi \rhd_{\text{cbv}} \; \Gamma_2 \vdash^{(m_2,e_2)} s : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m_2+1,e_1+e_2)} t : M}\;\mathtt{app}$$

where

- $m_1 + m_2 + 1 = m_1 + m' + 1 + 1 = m + 1$ and $e_1 + e_2 = e_1 + e' = e$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_1 + m_2 + 1 = m_1 + m' + 1 = m$ and $e_1 + e_2 = e_1 + e' + 1 = e + 11$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$.

– *Left explicit substitution*, *i.e.* $t \coloneqq s[x{\leftarrow}u] \to_{\mathsf{r}} s'[x{\leftarrow}u] =: t'$ with $s \to_{\mathsf{r}} s'$ and $\mathsf{r} \in \{\mathtt{m}_{\text{cbv}}, \mathsf{e}_{\text{cbv}}\}$. So, Φ' has the form

$$\dfrac{\Psi_1 \rhd_{\text{cbv}} \; \Gamma_1, x : N \vdash^{(m',e')} s' : M \qquad \Psi_2 \rhd_{\text{cbv}} \; \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m'+m_2,e'+e_2)} t' : M}\;\mathtt{ES}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $m \coloneqq m' + m_2$ and $e \coloneqq e' + e_2$. By *i.h.* applied to Ψ_1, there is a derivation $\Psi \rhd_{\text{CbV}} \; \Gamma_1, x : N \vdash^{(m_1,e_1)} s : M$ where

- $m_1 \coloneqq m' + 1$ and $e_1 \coloneqq e'$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_1 \coloneqq m'$ and $e_1 \coloneqq e' + 1$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$.

Thus, we have the derivation Φ

$$\dfrac{\Psi \rhd_{\text{cbv}} \; \Gamma_1, x : N \vdash^{(m_1,e_1)} s : M \qquad \Psi_2 \rhd_{\text{cbv}} \; \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m_2,e_1+e_2)} t : M}\;\mathtt{ES}$$

where

- $m_1 + m_2 = m' + 1 + m_2 = m + 1$ and $e_1 + e_2 = e' + e_2 = e$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_1 + m_2 = m' + m_2 = m$ and $e_1 + e_2 = e' + 1 + e_2 = e + 1$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$.

– *Right explicit substitution*, *i.e.* $t \coloneqq s[x{\leftarrow}u] \to_{\mathsf{r}} s[x{\leftarrow}u'] =: t'$ with $u \to_{\mathsf{r}} u'$ and $\mathsf{r} \in \{\mathtt{m}_{\text{cbv}}, \mathsf{e}_{\text{cbv}}\}$. So, Φ' has the form

$$\dfrac{\Psi_1 \rhd_{\text{cbv}} \; \Gamma_1, x : N \vdash^{(m_1,e_1)} s : M \qquad \Psi_2 \rhd_{\text{cbv}} \; \Gamma_2 \vdash^{(m',e')} u' : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m',e_1+e')} t' : M}\;\mathtt{ES}$$

where $\Gamma \coloneqq \Gamma_1 \uplus \Gamma_2$ and $m \coloneqq m_1 + m'$ and $e \coloneqq e_1 + e'$. By *i.h.* applied to Ψ_2, there is a derivation $\Psi \rhd_{\text{CbV}} \; \Gamma_2 \vdash^{(m_2,e_2)} u : M$ where

- $m_2 \coloneqq m' + 1$ and $e_2 \coloneqq e'$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_2 \coloneqq m'$ and $e_2 \coloneqq e' + 1$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$.

Thus, we have the derivation Φ

$$\dfrac{\Psi_1 \rhd_{\text{cbv}} \; \Gamma_1, x : N \vdash^{(m_1,e_1)} s : M \qquad \Psi \rhd_{\text{cbv}} \; \Gamma_2 \vdash^{(m_2,e_2)} u : N}{\Gamma_1 \uplus \Gamma_2 \vdash^{(m_1+m_2,e_1+e_2)} t : M}\;\mathtt{ES}$$

where

- $m_1 + m_2 = m_1 + m' + 1 = m + 1$ and $e_1 + e_2 = e_1 + e' = e$ if $\mathsf{r} = \mathtt{m}_{\text{cbv}}$,
- $m_1 + m_2 = m_1 + m' = m$ and $e_1 + e_2 = e_1 + e' + 1 = e + 1$ if $\mathsf{r} = \mathsf{e}_{\text{cbv}}$. $\square$

Theorem 11 (CbV tight completeness). *Let t be a closed term. If $d\colon t \to_{\text{cbv}}^* s$ with $\mathsf{normal}_{\text{cbv}}(s)$, then there is a tight derivation $\Phi \rhd_{\text{cbv}} \;\vdash^{(|d|_{\mathtt{m}},|d|_{\mathtt{e}})} t : \mathbf{0}$.*

Proof. By induction on the length $k := |d|$ of the evaluation $d\colon t \to_{\text{cbv}}^* s$.

If $k = 0$ then $t = s$ and $\mathsf{normal}_{\text{cbv}}(t)$. Proposition 8 gives the existence of a tight derivation $\Phi \rhd_{\text{cbv}} \;\vdash^{(0,0)} t : \mathbf{0}$, that satisfies the statement because $|d|_{\mathtt{m}} = |d|_{\mathtt{e}} = 0$.

If $k > 0$ then $d\colon t \to_{\text{cbv}} u \to_{\text{cbv}}^{k-1} s$. Let d' be the evaluation $u \to_{\text{cbv}}^{k-1} s$. Thus, if $t \to_{\mathtt{m}_{\text{cbv}}} u$ then $|d|_{\mathtt{m}} = |d'|_{\mathtt{m}} + 1$ and $|d|_{\mathtt{e}} = |d'|_{\mathtt{e}}$; otherwise $t \to_{\mathtt{e}_{\text{cbv}}} u$ then $|d|_{\mathtt{m}} = |d'|_{\mathtt{m}}$ and $|d|_{\mathtt{e}} = |d'|_{\mathtt{e}} + 1$ By *i.h.*, there exists a tight derivation $\Psi \rhd_{\text{cbv}} \;\vdash^{(|d'|_{\mathtt{m}},|d'|_{\mathtt{e}})} u : \mathbf{0}$. By quantitative subject expansion (Proposition 9), there exists a derivation $\Phi \rhd_{\text{cbv}} \;\vdash^{(|d|_{\mathtt{m}},|d|_{\mathtt{e}})} u : \mathbf{0}$, in particular Φ is tight and with indices $(|d|_{\mathtt{m}}, |d|_{\mathtt{e}})$. □

D Types by Need (Sect. 6)

D.1 CbNeed Correctness

Lemma 17 (Basic properties of derivations in CbNeed). *Let $\Phi \rhd_{CbNeed} \Gamma \vdash^{(m,e)} t : M$ be a derivation. Then,*

1. *if $x \notin \mathtt{fv}(t)$ then $x \notin \mathtt{dom}(\Gamma)$,*
2. *if $t = E\langle\langle x\rangle\rangle$ and $M \neq \mathbf{0}$ then $x \in \mathtt{dom}(\Gamma)$.*

Proof. 1. We prove that $\mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(t)$ by induction on Φ:

- Rules ax and normal satisfy the statement, as can be observed by a simple analysis of the typing rules.
- Rules app, app_{gc} and many satisfy the statement by a trivial application of the *i.h.*.
- Rule $\mathtt{fun}$: By *i.h.*, $x \cup \mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(t)$, and so $\mathtt{dom}(\Gamma) = \mathtt{dom}(x : M; \Gamma) \setminus \{x\} \subseteq \mathtt{fv}(t) \setminus \{x\} = \mathtt{fv}(\lambda x.t)$.
- *Rule ES*: We first apply *i.h.* on the left-hand side premise to obtain that $\mathtt{dom}(x : M; \Gamma) \subseteq \mathtt{fv}(t)$, which in turn implies that $\mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(t) \setminus \{x\}$. We then apply *i.h.* on the right-hand side premise to obtain that $\mathtt{dom}(\Pi) \subseteq \mathtt{fv}(s)$. Hence, $\mathtt{dom}(\Gamma) \cup \mathtt{dom}(\Pi) \subseteq (\mathtt{fv}(t) \setminus \{x\}) \cup \mathtt{fv}(s) = \mathtt{fv}(t[x{\leftarrow}s])$.
- *Rule ES_{gc}*: By *i.h.*, $\mathtt{dom}(\Gamma) \subseteq \mathtt{fv}(t)$. But $x \notin \mathtt{dom}(\Gamma)$, and so $\mathtt{dom}(\Gamma) = \mathtt{dom}(\Gamma) \setminus \{x\} \subseteq \mathtt{fv}(t) \setminus \{x\} \subseteq (\mathtt{fv}(t) \setminus \{x\}) \cup \mathtt{fv}(s) = \mathtt{fv}(t[x{\leftarrow}s])$.

2. By induction on the construction of E:
   - Let $E = \langle\cdot\rangle$. Then $t = x$ and so Φ is of the form

$$\frac{}{x : M \vdash^{(m,e)} x : M}\ \mathsf{ax}$$

   Clearly, if $M \neq \mathbf{0}$ then $x \in \mathtt{dom}(\Gamma)$.
   - If $E = E_1 s$, then $t = E_1\langle\langle x\rangle\rangle s$. Then Φ can only have either app or app_{gc} as the last typing rule.
     - Let Φ be of the form

$$\frac{\Phi_{E_1\langle\langle x\rangle\rangle} \rhd_{\text{CbNeed}} \ \Pi \vdash^{(m',e')} E_1\langle\langle x\rangle\rangle : [N \multimap M] \qquad \Delta \vdash^{(m'',e'')} s : N}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} E_1\langle\langle x\rangle\rangle s : M}\ \mathsf{app}$$

Then by application of *i.h.* on $\Phi_{E_1\langle\langle x\rangle\rangle}$ we obtain that $x \in \mathtt{dom}(\Pi)$, finally obtaining $x \in \mathtt{dom}(\Pi \uplus \Delta)$.

- If Φ has $\mathtt{app}_{\mathrm{gc}}$ as its last typing rule instead, then $x \in \mathtt{dom}(\Gamma)$ simply by *i.h.*.

– If $E = E_1[y\leftarrow s]$, then $t = E\langle\langle x\rangle\rangle = E_1\langle\langle x\rangle\rangle[y\leftarrow s]$, with $x \neq y$. Then Φ can only have either ES or $\mathrm{ES}_{\mathrm{gc}}$ as the last typing rule.

- Let Φ be of the form

$$\dfrac{\Phi_{E_1\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \; y : N; \Pi \vdash^{(m',e')} E_1\langle\langle x\rangle\rangle : M \quad \Delta \vdash^{(m'',e'')} s : N}{\Pi \uplus \Delta \vdash^{(m'+e'',e'+e'')} E_1\langle\langle x\rangle\rangle[y\leftarrow s] : M}\,\text{ES}$$

Then by application of *i.h.* on $\Phi_{E_1\langle\langle x\rangle\rangle}$ we obtain that $x \in \mathtt{dom}(y : N; \Pi)$, which implies that $x \in \mathtt{dom}(\Pi)$ and so $x \in \mathtt{dom}(\Pi \uplus \Delta)$.

- If Φ has $\mathrm{ES}_{\mathrm{gc}}$ as its last typing rule instead, then $x \in \mathtt{dom}(\Gamma)$ simply by *i.h.*.

– Let $E = E_1\langle\langle y\rangle\rangle[y\leftarrow E_2]$, and so $t = E\langle\langle x\rangle\rangle = E_1\langle\langle y\rangle\rangle[y\leftarrow E_2\langle\langle x\rangle\rangle]$, with $x \neq y$ because x is a free variable of t while y is a bound variable of t, and we are working up to α-equivalence. Suppose now that $\mathrm{ES}_{\mathrm{gc}}$ was the last typing rule of Φ. This means that Φ is of the form

$$\dfrac{\Phi_{E_1\langle\langle y\rangle\rangle} \triangleright_{\text{CbNeed}} \; \Gamma \vdash^{(m,e)} E_1\langle\langle y\rangle\rangle : M \quad y \notin \mathtt{dom}(\Gamma)}{\Gamma \vdash^{(m,e)} E_1\langle\langle y\rangle\rangle[y\leftarrow E_2\langle\langle x\rangle\rangle] : M}\,\mathrm{ES}_{\mathrm{gc}}$$

However, by applying *i.h.* on $\Phi_{E_1\langle\langle y\rangle\rangle}$ we obtain that $y \in \mathtt{dom}(\Gamma)$, which is in contradiction with the constraint of rule $\mathrm{ES}_{\mathrm{gc}}$.
Hence, Φ can only have ES as the last typing rule. Thus, Φ is of the form

$$\dfrac{y : N; \Pi \vdash^{(m',e')} E_1\langle\langle y\rangle\rangle : M \quad \Phi_{E_2\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \; \Delta \vdash^{(m'',e'')} E_2\langle\langle x\rangle\rangle : N \quad N \neq \mathbf{0}}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} E_1\langle\langle y\rangle\rangle[y\leftarrow E_2\langle\langle x\rangle\rangle] : M}\,\text{ES}$$

We can then apply *i.h.* on $\Phi_{E_2\langle\langle x\rangle\rangle}$ to obtain that $x \in \mathtt{dom}(\Delta)$ and so $x \in \mathtt{dom}(\Pi \uplus \Delta)$. □

Lemma 18 (Splitting of multi-sets with respect to derivations). *Let v be a value and $\Phi \triangleright_{CbNeed} \Gamma \vdash^{(m,e)} v : M$ be a derivation such that $|M| \geq 2$. Then, for every splitting $M = N \uplus O$ such that $|N|, |O| \geq 1$ there are type contexts Γ_N and Γ_O and derivations $\Phi_N \triangleright_{CbNeed} \Gamma_N \vdash^{(m_N,e_N)} v : N$ and $\Phi_O \triangleright_{CbNeed} \Gamma_O \vdash^{(m_O,e_O)} v : O$ such that*
- *$\Gamma = \Gamma_N \uplus \Gamma_O$,*
- *$m = m_N + m_O$, and*
- *$e = e_N + e_O$.*

Proof. By a simple observation of the typing rules with an abstraction as the term of the final type judgement, we note that Φ can only be of the form

$$\dfrac{(\Gamma_i \vdash^{(m_i,e_i)} v : L_i)_{i\in I}}{\biguplus_{i\in I} \Gamma_i \vdash^{(\sum_{i\in I} m_i, \sum_{i\in I} e_i)} v : [L_i]_{i\in I}}\,\mathsf{many}$$

We then appropriately define $J = \{i \in I : L_i \in N\}$ and $K = \{i \in I : L_i \in O\}$; *i.e.*, $[L_j]_{j\in J} = N$, $[L_k]_{k\in K} = O$, and making sure that $J \cap K = \emptyset$. Note that $J, K \neq \emptyset$, since $N, O \neq \mathbf{0}$. Thus, we obtain Φ_N as

$$\frac{(\Gamma_j \vdash^{(m_j,e_j)} v : L_j)_{j\in J}}{\biguplus_{j\in J} \Gamma_j \vdash^{(\sum_{j\in J} m_j, \sum_{j\in J} e_j)} v : [L_j]_{j\in J}}\ \mathsf{many}$$

and Φ_O as

$$\frac{(\Gamma_k \vdash^{(m_k,e_k)} v : L_k)_{k\in K}}{\biguplus_{k\in K} \Gamma_k \vdash^{(\sum_{k\in K} m_k, \sum_{k\in K} e_k)} v : [L_k]_{k\in K}}\ \mathsf{many}$$

where $\biguplus_{j\in J} \Gamma_j = \Gamma_N$, $\biguplus_{k\in K} \Gamma_k = \Gamma_O$, $(\sum_{j\in J} m_j, \sum_{j\in J} e_j) = (m_N, e_N)$, and $(\sum_{k\in K} m_k, \sum_{k\in K} e_k) = (m_O, e_O)$. □

Lemma 19 (CbNeed linear substitution). *Let $\Phi_{E\langle\langle x\rangle\rangle} \triangleright_{\text{need}} x : M; \Gamma \vdash^{(m,e)} E\langle\langle x\rangle\rangle : O$ be a derivation and v a value such that $O \neq \mathbf{0}$ and E does not capture the free variables of v. Then there exists a splitting $M = M_1 \uplus M_2$, with $M_1 \neq \mathbf{0}$, such that for every derivation $\Psi \triangleright_{\text{need}} \Pi \vdash^{(m',e')} v : M_1$ there exists a derivation $\Phi_{E\langle\langle v\rangle\rangle} \triangleright_{\text{need}} x : M_2; \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} E\langle\langle v\rangle\rangle : O$.*

Proof. We prove this by induction on N:

– *Empty context,* i.e. $N = \langle\cdot\rangle$. Then $\Gamma = \emptyset$, $O = M$, and $\Phi_{E\langle\langle x\rangle\rangle}$ is of the form

$$\frac{}{x \colon M \vdash^{(0,1)} x : M}\ \mathsf{ax}$$

Therefore, by defining $M_1 \coloneqq M$ and $M_2 \coloneqq \mathbf{0}$, the statement holds for every $\Psi \triangleright_{\text{CbNeed}}\ \Pi \vdash^{(m',e')} v : M_1$ by taking $\Phi_{E\langle\langle v\rangle\rangle} \coloneqq \Psi$. In particular, note that $(m + m', e + e' - 1) = (0 + m', 1 + e' - 1) = (m', e')$.

– *Left of an application,* i.e. $N = N_1 s$. There are two possible last rules in $\Phi_{E\langle\langle x\rangle\rangle}$, namely app_b or app_{gc}.

  • Let $\Phi_{E\langle\langle x\rangle\rangle}$ be of the form

$$\frac{x \colon M_\Delta; \Delta \vdash^{(m_\Delta,e_\Delta)} N_1\langle\langle x\rangle\rangle : [O' \multimap O] \quad x \colon M_\Sigma; \Sigma \vdash^{(m_\Sigma,e_\Sigma)} s : O'}{x \colon (M_\Delta \uplus M_\Sigma); (\Delta \uplus \Sigma) \vdash^{(m_\Delta+m_\Sigma+1, e_\Delta+e_\Sigma)} E_1\langle\langle x\rangle\rangle s : O}\ \mathsf{app}_b$$

where $\Gamma = \Delta \uplus \Sigma$, $\Delta(x) = \Sigma(x) = \mathbf{0}$ and $M = M_\Delta \uplus M_\Sigma$.
By applying the *i.h.* on the left-hand side premise we obtain that there exists a splitting $M_\Delta = M_{\Delta,1} \uplus M_{\Delta,2}$, with $M_{\Delta,1} \neq \mathbf{0}$, such that for every derivation $\Psi \triangleright_{\text{ CbNeed}} \Pi \vdash^{(m',e')} v : M_{\Delta,1}$ there exists a derivation $\Phi_{E_1\langle\langle v\rangle\rangle} \triangleright_{\text{CbNeed}}\ x \colon M_{\Delta,2}; \Delta \uplus \Pi \vdash^{(m_\Delta+m',e_\Delta+e'-1)} E_1\langle\langle v\rangle\rangle : [O' \multimap O]$.
We can then construct $\Phi_{E\langle\langle v\rangle\rangle}$ for such a Ψ as follows

$$\frac{x \colon M_{\Delta,2}; \Delta \uplus \Pi \vdash^{(m_\Delta+m',e_\Delta+e'-1)} E_1\langle\langle v\rangle\rangle : [O' \multimap O] \quad x \colon M_\Sigma; \Sigma \vdash^{(m_\Sigma,e_\Sigma)} s : O'}{x \colon M_{\Delta,2} \uplus M_\Sigma; \Delta \uplus \Pi \uplus \Sigma \vdash^{(m_\Delta+m'+m_\Sigma+1, e_\Delta+e'-1+e_\Sigma)} E_1\langle\langle v\rangle\rangle s : O}\ \mathsf{app}_b$$

Note that $\Phi_{E\langle\langle v\rangle\rangle}$ is as desired by splitting M into $M_1 \coloneqq M_{\Delta,1}$ and $M_2 \coloneqq M_{\Delta,2} \uplus M_\Sigma$.

- Let $\Phi_{E\langle\langle x\rangle\rangle}$ be of the form

$$\dfrac{\Phi_{E_1\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \ x\colon M; \Gamma \vdash^{(m-1,e)} E_1\langle\langle x\rangle\rangle : [\mathbf{0} \multimap O]}{x\colon M; \Gamma \vdash^{(m,e)} E_1\langle\langle x\rangle\rangle s : O}\ \mathtt{app}_{\text{gc}}$$

By applying the *i.h.* on $\Phi_{E_1\langle\langle x\rangle\rangle}$ we obtain that there exists a splitting $M = M_1 \uplus M_2$, with $M_1 \neq \mathbf{0}$, such that for every derivation $\Psi \triangleright_{\text{CbNeed}} \ \Pi \vdash^{(m',e')} v : M_1$ there exists a derivation $\Phi_{E_1\langle\langle v\rangle\rangle} \triangleright_{\text{CbNeed}} x\colon M_2; \Gamma \uplus \Pi \vdash^{(m-1+m',e+e'-1)} E_1\langle\langle v\rangle\rangle : [\mathbf{0} \multimap O]$. We can then construct $\Phi_{E\langle\langle v\rangle\rangle}$ for such a Ψ as follows

$$\dfrac{x\colon M_2; \Gamma \uplus \Pi \vdash^{(m-1+m',e+e'-1)} E_1\langle\langle v\rangle\rangle : [\mathbf{0} \multimap O]}{x\colon M_2; \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} E_1\langle\langle v\rangle\rangle s : O}\ \mathtt{app}_{\text{gc}}$$

– *Left of a substitution;* i.e. $E = E_1[y\leftarrow s]$. Note that $x \neq y$, because the hypothesis $E\langle\langle x\rangle\rangle$ impies that E does not capture x. There are two possible last rules in $\Phi_{E\langle\langle x\rangle\rangle}$, namely ES and ES_{gc}.

- Let $\Phi_{E\langle\langle x\rangle\rangle}$ be of the form

$$\dfrac{x\colon M_\Delta; \Delta \vdash^{(m_\Delta,e_\Delta)} E_1\langle\langle x\rangle\rangle : O \quad x\colon M_\Sigma; \Sigma \vdash^{(m_\Sigma,e_\Sigma)} s : \Delta(y) \quad \Delta(y) \neq \mathbf{0}}{x\colon (M_\Delta \uplus M_\Sigma); (\Delta \setminus\!\!\setminus y) \uplus \Sigma \vdash^{(m_\Delta+m_\Sigma,e_\Delta+e_\Sigma)} E_1\langle\langle x\rangle\rangle[y\leftarrow s] : O}\ \text{ES}$$

where $M = M_\Delta \uplus M_\Sigma$ and $\Gamma = (\Delta \setminus\!\!\setminus y) \uplus \Sigma$.
By applying the *i.h.* on the leftmost premise we obtain a splitting $M_\Delta = M_{\Delta,1} \uplus M_{\Delta,2}$, with $M_{\Delta,1} \neq \mathbf{0}$, such that for every derivation $\Psi \triangleright_{\text{CbNeed}} \ \Pi \vdash^{(m',e')} v : M_{\Delta,1}$ there exists a derivation $\Phi_{E_1\langle\langle v\rangle\rangle} \triangleright_{\text{CbNeed}} x\colon M_{\Delta,2}; \Delta \uplus \Pi \vdash^{(m_\Delta+m',e_\Delta+e'-1)} E_1\langle\langle v\rangle\rangle : O$. Note however that if $y \in \mathtt{dom}(\Pi)$, then Lemma 17 applied on Ψ would imply that $y \in \mathtt{fv}(v)$, which contradicts the hypothesis that E does not capture the free variables of v; *i.e.*, $y \notin \mathtt{dom}(\Pi)$, and so $(\Pi \uplus \Delta)(y) = \Delta(y)$. We can then construct $\Phi_{E\langle\langle v\rangle\rangle}$ for such a Ψ as follows

$$\dfrac{x\colon M_{\Delta,2}; \Delta \uplus \Pi \vdash^{(m_\Delta+m',e_\Delta+e'-1)} E_1\langle\langle v\rangle\rangle : O \quad x\colon M_\Sigma; \Sigma \vdash^{(m_\Sigma,e_\Sigma)} s : \Delta(y) \quad \Delta(y) \neq \mathbf{0}}{x\colon M_{\Delta,2} \uplus M_\Sigma; ((\Delta \uplus \Pi) \setminus\!\!\setminus y) \uplus \Sigma \vdash^{(m_\Delta+m'+m_\Sigma,e_\Delta+e'-1+e_\Sigma)} E_1\langle\langle v\rangle\rangle[y\leftarrow s] : O}\ \text{ES}$$

by splitting M into $M_1 \coloneqq M_{\Delta,1}$ and $M_2 \coloneqq M_{\Delta,2} \uplus M_\Sigma$. Since $y \notin \mathtt{dom}(\Pi)$, then $((\Delta \uplus \Pi) \setminus\!\!\setminus y) \uplus \Sigma = (\Delta \setminus\!\!\setminus y) \uplus \Pi \uplus \Sigma = \Gamma \uplus \Pi$.

- Let $\Phi_{E\langle\langle x\rangle\rangle}$ be of the form

$$\dfrac{x\colon M; \Gamma \vdash^{(m,e)} E_1\langle\langle x\rangle\rangle : O \quad \Gamma(y) = \mathbf{0}}{x\colon M; \Gamma \vdash^{(m,e)} E_1\langle\langle x\rangle\rangle[y\leftarrow s] : O}\ \text{ES}_{\text{gc}}$$

By applying *i.h.* on the premise we obtain a splitting $M = M_1 \uplus M_2$, with $M_1 \neq \mathbf{0}$, such that for every derivation $\Psi \triangleright_{\text{CbNeed}} \ \Pi \vdash^{(m',e')} v : M_1$ there exists a derivation $\Phi_{E_1\langle\langle v\rangle\rangle} \triangleright_{\text{CbNeed}} \ x\colon M_2; \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)}$

$E_1\langle\!\langle v\rangle\!\rangle : O$. Note that $y \notin \mathtt{dom}(\Pi)$, because applying Lemma 17 on Ψ would otherwise imply that $y \in \mathtt{fv}(v)$, which contradicts the hypothesis that E does not capture the free variables of v. Hence, we can then construct $\Phi_{E\langle\!\langle v\rangle\!\rangle}$ for such a Ψ as follows

$$\dfrac{x\colon M_2; \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} E\langle\!\langle v\rangle\!\rangle : O \quad (\Gamma \uplus \Pi)(y) = \Gamma(y) = \mathbf{0}}{x\colon M_2; \Gamma \uplus \Pi \vdash^{(m+m',e+e'-1)} E_1\langle\!\langle v\rangle\!\rangle[y\leftarrow s] : O}\ \mathrm{ES}_{\mathrm{gc}}$$

– Let $E = E_1\langle\!\langle y\rangle\!\rangle[y\leftarrow E_2]$. We can safely assume that $x \neq y$, since we are working up to α-equivalence. Lemma 17 implies $y \in \mathtt{dom}(x\colon M; \Gamma)$, and so $\Phi_{E\langle\!\langle x\rangle\!\rangle}$ can only have ES as the final type judgement and be of the form

$$\dfrac{x\colon M_\Delta; \Delta \vdash^{(m_\Delta,e_\Delta)} E_1\langle\!\langle y\rangle\!\rangle : O \quad x\colon M_\Sigma; \Sigma \vdash^{(m_\Sigma,e_\Sigma)} E_2\langle\!\langle x\rangle\!\rangle : \Delta(y) \quad \Delta(y) \neq \mathbf{0}}{x\colon (M_\Delta \uplus M_\Sigma); (\Delta \setminus\!\!\setminus y) \uplus \Sigma \vdash^{(m_\Delta+m_\Sigma,e_\Delta+e_\Sigma)} E_1\langle\!\langle y\rangle\!\rangle[y\leftarrow E_2\langle\!\langle x\rangle\!\rangle] : O}\ \mathrm{ES}$$

where $M = M_\Delta \uplus M_\Sigma$, $\Gamma = (\Delta \setminus\!\!\setminus y) \uplus \Sigma$, and $(m, e) = (m_\Delta + m_\Sigma, e_\Delta + e_\Sigma)$. We can then apply the *i.h.* on the premise in the middle to obtain a splitting $M_\Sigma = M_{\Sigma,1} \uplus M_{\Sigma,2}$, with $M_{\Sigma,1} \neq \mathbf{0}$, such that for every derivation $\Psi \rhd_{\text{CbNeed}} \Pi \vdash^{(m',e')} v : M_{\Sigma,1}$ there exists a derivation $\Phi_{E_2\langle\!\langle v\rangle\!\rangle} \rhd_{\text{CbNeed}} x\colon M_{\Sigma,2}; \Sigma \uplus \Pi \vdash^{(m_\Sigma+m',e_\Sigma+e'-1)} E_2\langle\!\langle v\rangle\!\rangle : \Delta(y)$. We can then construct $\Phi_{E\langle\!\langle v\rangle\!\rangle}$ for such Ψ as follows

$$\dfrac{x\colon M_\Delta; \Delta \vdash^{(m_\Delta,e_\Delta)} E_1\langle\!\langle y\rangle\!\rangle : O \quad x\colon M_{\Sigma,2}; \Sigma \uplus \Pi \vdash^{(m_\Sigma+m',e_\Sigma+e'-1)} E_2\langle\!\langle v\rangle\!\rangle : \Delta(y) \quad \Delta(y) \neq \mathbf{0}}{x\colon (M_\Delta \uplus M_{\Sigma,2}); (\Delta \setminus\!\!\setminus y) \uplus \Sigma \uplus \Pi \vdash^{(m_\Delta+m_\Sigma+m',e_\Delta+e_\Sigma+e'-1)} E_1\langle\!\langle y\rangle\!\rangle[y\leftarrow E_2\langle\!\langle v\rangle\!\rangle] : O}\ \mathrm{ES}$$

where we take $M_1 \coloneqq M_{\Sigma,1}$ and $M_2 \coloneqq M_\Delta \uplus M_{\Sigma,2}$. □

Proposition 23 (Quantitative subject reduction for CbNeed). *Let $\Phi \rhd_{\text{need}} \Gamma \vdash^{(m,e)} t : M$ be a derivation such that $M \neq \mathbf{0}$.*

– Multiplicative*: if $t \to_{\mathtt{m}_{\text{need}}} s$ then $m \geq 1$ and there is a derivation $\Phi' \rhd_{\text{need}} \Gamma \vdash^{(m-1,e)} t : M$.*

– Exponential*: if $t \to_{\mathtt{e}_{\text{need}}} s$ then $e \geq 1$ and there exists a derivation $\Phi' \rhd_{\text{need}} \Gamma \vdash^{(m,e-1)} t : M$.*

Proof. By induction on the reduction relation $\to_{\text{need}}$, with $\mapsto_{\mathtt{m}_{\text{need}}}$ and $r \to_{\mathtt{e}_{\text{need}}}$ as the base cases, and the closure by CbNeed contexts of $\mapsto_{\mathtt{m}_{\text{need}}} \cup \mapsto_{\mathtt{e}_{\text{need}}}$ as the inductive one.

– *Root step for* $\to_{\mathtt{m}_{\text{need}}}$. Let us assume that $t = S\langle \lambda x.u\rangle r \mapsto_{\mathtt{m}} S\langle u[x\leftarrow r]\rangle = s$, and proceed by induction on S:
  - Let $S = \langle\,\rangle$. Then $t = (\lambda x.u)r$ and so the last rule of Φ is either $\mathtt{app}_b$ or $\mathtt{app}_{\mathrm{gc}}$, because they are the only rules whose term in the conclusion type judgement is an application.
    * If $\mathtt{app}_b$ is the last rule of Φ, then the latter is of the form

$$\dfrac{\dfrac{\dfrac{\Pi \vdash^{(m',e')} u : M}{\Pi \setminus\!\!\setminus x \vdash^{(m',e')} \lambda x.u : \Pi(x) \multimap M}\ \mathtt{fun}}{\Pi \setminus\!\!\setminus x \vdash^{(m',e')} \lambda x.u : [\Pi(x) \multimap M]}\ \mathtt{many} \qquad \Delta \vdash^{(m'',e'')} r : \Pi(x) \quad \Pi(x) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus x) \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} (\lambda x.u)r : M}\ \mathtt{app}_b$$

Therefore, $m \geq 1$. Since $\Pi(x) \neq \mathbf{0}$, then we can construct Φ' as follows:

$$\dfrac{\Pi \vdash^{(m',e')} u : M \quad \Delta \vdash^{(m'',e'')} r : \Pi(x) \quad \Pi(x) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus x) \uplus \Delta \vdash^{(m'+m'',e'+e'')} u[x{\leftarrow}r] : M}\ \text{ES}$$

Note that $(m' + m'', e' + e') = (m - 1, e)$.

* If $\mathtt{app}_{\text{gc}}$ is the las typing rule of Φ, then the latter is of the form

$$\dfrac{\dfrac{\dfrac{\Pi \vdash^{(m',e')} u : M}{\Pi \vdash^{(m',e')} \lambda x.u : \mathbf{0} \multimap M}\ \mathtt{fun}}{\Pi \vdash^{(m',e')} \lambda x.u : [\mathbf{0} \multimap M]}\ \mathsf{many}}{\Pi \vdash^{(m'+1,e')} (\lambda x.u)r : M}\ \mathtt{app}_{\text{gc}}$$

with $(m, e) = (m' + 1, e')$ and $\Gamma = \Pi \setminus\!\!\setminus x$. Note that $x \notin \mathtt{dom}(\Pi)$, because u is typed with M and $\lambda x.u$ is typed with $\mathbf{0} \multimap M$, so we can construct Φ' as follows:

$$\dfrac{\Pi \vdash^{(m',e')} u : M \quad \Pi(x) = \mathbf{0}}{\Pi \vdash^{(m',e')} u[x{\leftarrow}r] : M}\ \text{ES}_{\text{gc}}$$

- Let $S = S'[y{\leftarrow}q]$. Then $t = S\langle\lambda x.u\rangle r = ((S'\langle\lambda x.u\rangle)[y{\leftarrow}q])r \mapsto_{\mathtt{m}} (S'\langle u[x{\leftarrow}r]\rangle)[y{\leftarrow}q] = S\langle u[x{\leftarrow}r]\rangle = s$. Since we are working up to α-equivalence, it is safe to assume that $y \notin \mathtt{fv}(q)$ and $y \notin \mathtt{fv}(r)$. There are several possible forms of Φ, namely:
  * If the last rule is $\mathtt{app}_{\text{gc}}$ and $[y{\leftarrow}q]$ is appended through rule ES_{gc}, then Φ is of the form

$$\dfrac{\dfrac{\Gamma \vdash^{(m',e')} S'\langle\lambda x.u\rangle : [\mathbf{0} \multimap N] \quad \Gamma(y) = \mathbf{0}}{\Gamma \vdash^{(m',e')} S\langle\lambda x.u\rangle : [\mathbf{0} \multimap N]}\ \text{ES}_{\text{gc}}}{\Gamma \vdash^{(m'+1,e')} S\langle\lambda x.u\rangle r : N}\ \mathtt{app}_{\text{gc}}$$

We then construct the following derivation

$$\dfrac{\Gamma \vdash^{(m',e')} S'\langle\lambda x.u\rangle : [\mathbf{0} \multimap N]}{\Gamma \vdash^{(m'+1,e')} S'\langle\lambda x.u\rangle r : N}\ \mathtt{app}_{\text{gc}}$$

and apply *i.h.* on it to obtain $m = m' + 1 \geq 1$. Moreover, the *i.h.* also yields a derivation $\Phi'' \rhd_{\text{CbNeed}} \Gamma \vdash^{(m',e')} S'\langle u[x{\leftarrow}r]\rangle : N$, with which we can then construct Φ' as follows:

$$\dfrac{\Phi'' \rhd_{\text{CbNeed}} \Gamma \vdash^{(m',e')} S'\langle u[x{\leftarrow}r]\rangle : N \quad \Gamma(y) \neq \mathbf{0}}{\Gamma \vdash^{(m',e')} S\langle u[x{\leftarrow}r]\rangle : N}\ \text{ES}_{\text{gc}}$$

Finally, note that $(m', e') = (m - 1, e)$.

* If the last rule is $\mathtt{app}_{\mathrm{gc}}$ and $[y{\leftarrow}q]$ is appended through rule ES, then Φ is

$$\cfrac{\cfrac{\Pi \vdash^{(m',e')} S'\langle \lambda x.u\rangle : [\mathbf{0} \multimap M] \quad \Delta \vdash^{(m'',e'')} r : N \quad \Pi(y) = N \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus y) \uplus \Delta \vdash^{(m'+m'',e'+e'')} S\langle \lambda x.u\rangle : [\mathbf{0} \multimap M]}\ \mathrm{ES}}{(\Pi \setminus\!\!\setminus y) \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} S\langle \lambda x.u\rangle r : M}\ \mathtt{app}_{\mathrm{gc}}$$

We can then construct the following derivation

$$\frac{\Pi \vdash^{(m',e')} S'\langle \lambda x.u\rangle : [\mathbf{0} \multimap M]}{\Pi \vdash^{(m'+1,e')} S'\langle \lambda x.u\rangle r : M}\ \mathtt{app}_{\mathrm{gc}}$$

Applying *i.h.* on it yields a derivation $\Phi'' \triangleright_{\mathrm{CbNeed}} \Pi \vdash^{(m',e')} S'\langle u[x{\leftarrow}r]\rangle : M$ and implies the fact that $m = m' + m'' + 1 \geq m' + 1 \geq 1$. Finally, we construct Φ' as follows:

$$\frac{\Phi'' \triangleright_{\mathrm{CbNeed}} \ \Pi \vdash^{(m',e')} S'\langle u[x{\leftarrow}r]\rangle : M \quad \Delta \vdash^{(m'',e'')} r : N \quad \Pi(y) = N \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus y) \uplus \Delta \vdash^{(m'+m'',e'+e'')} S\langle u[x{\leftarrow}r]\rangle : M}\ \mathrm{ES}$$

Note that $(m' + m'', e' + e'') = (m - 1, e)$.

* If the last rule is $\mathtt{app}_b$ and $[y{\leftarrow}q]$ is appended through rule $\mathrm{ES}_{\mathrm{gc}}$, then Φ is

$$\cfrac{\cfrac{\Pi \vdash^{(m',e')} S'\langle \lambda x.u\rangle : [M \multimap N] \quad \Pi(y) = \mathbf{0}}{\Pi \vdash^{(m',e')} S\langle \lambda x.u\rangle : [M \multimap N]}\ \mathtt{ES}_{\mathrm{gc}} \qquad \Delta \vdash^{(m'',e'')} r : M}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} S\langle \lambda x.u\rangle r : N}\ \mathtt{app}\ M \neq \mathbf{0}$$

We are now able to give the following derivation

$$\frac{\Pi \vdash^{(m',e')} S'\langle \lambda x.u\rangle : [M \multimap N] \quad \Delta \vdash^{(m'',e'')} r : M}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} S'\langle \lambda x.u\rangle r : N}\ \mathtt{app}$$

on which application of *i.h.* gives that $m = m' + m'' + 1 \geq 1$, and yields a derivation $\Phi'' \triangleright_{\mathrm{CbNeed}} \ \Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} S'\langle u[x{\leftarrow}r]\rangle : N$, thus allowing us to construct Φ' as follows:

$$\frac{\Phi'' \triangleright_{\mathrm{CbNeed}} \ \Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} S'\langle u[x{\leftarrow}r]\rangle : N \quad (\Pi \uplus \Delta)(y) = \mathbf{0}}{\Phi'' \triangleright_{\mathrm{CbNeed}} \ \Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} S\langle u[x{\leftarrow}r]\rangle : N}\ \mathtt{ES}_{\mathrm{gc}}$$

Note that $y \notin \mathtt{fv}(q)$ and so via Lemma 17 we know that $\Delta(y) = \mathbf{0}$; hence, the use of rule $\mathtt{ES}_{\mathrm{gc}}$ in Φ' is correct. Moreover, note that $(m' + m'', e' + e'') = (m, e)$.

* If the last rule is $\mathtt{app}$ and $[y{\leftarrow}q]$ is appended through rule $\mathtt{ES}$, then Φ is

$$\cfrac{\cfrac{\Pi \vdash^{(m',e')} S'\langle \lambda x.u\rangle : [M \multimap N] \quad \Delta \vdash^{(m'',e'')} q : \Pi(y) \quad \Pi(y) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus y) \uplus \Delta \vdash^{(m'+m'',e'+e'')} S\langle \lambda x.u\rangle : [M \multimap N]}\ \mathtt{ES} \qquad \Sigma \vdash^{(m''',e''')} r : M}{((\Pi \setminus\!\!\setminus y) \uplus \Delta) \uplus \Sigma \vdash^{((m'+m'')+m'''+1,(e'+e'')+e''')} S\langle \lambda x.u\rangle r : N}\ \mathtt{app}$$

Since $y \notin \mathtt{fv}(r) \cup \mathtt{fv}(q)$ then we know through Lemma 17 that $y \notin \mathtt{dom}(\Delta)$ and $y \notin \mathtt{dom}(\Sigma)$. Now, applying *i.h.* on the following derivation

$$\dfrac{\Pi \vdash^{(m',e')} S'\langle \lambda x.u\rangle : [M \multimap N] \quad \Sigma \vdash^{(m''',e''')} r : M}{\Pi \uplus \Sigma \vdash^{(m'+m'''+1,e'+e''')} S'\langle \lambda x.u\rangle r : N}\ \mathtt{app}$$

yields a derivation $\Phi'' \triangleright_{\text{CbNeed}} \Pi \uplus \Sigma \vdash^{(m'+m''',e'+e''')} L'\langle u[x{\leftarrow}r]\rangle : N$ and implies $m = (m'+m'') + m''' + 1 \geq m' + m''' + 1 \geq 1$. Finally, given that $(\Pi \uplus \Sigma)(y) = \Pi(y)$, we can finally construct Φ' as follows:

$$\dfrac{\Phi'' \triangleright_{\text{CbNeed}}\ \Pi \uplus \Sigma \vdash^{(m'+m''',e'+e''')} S'\langle u[x{\leftarrow}r]\rangle : N \quad \Delta \vdash^{(m'',e'')} q : \Pi(y) \quad \Pi(y) \neq \mathbf{0}}{((\Pi \uplus \Sigma) \setminus\!\!\setminus y) \uplus \Delta \vdash^{((m'+m''')+m'',(e'+e''')+e'')} S\langle u[x{\leftarrow}r]\rangle : N}\ \text{ES}$$

Note that $((m'+m''')+m'', (e'+e''')+e'') = (m-1, e)$, and that since $y \notin \mathtt{dom}(\Sigma)$ then $((\Pi \uplus \Sigma) \setminus\!\!\setminus y) \uplus \Delta = ((\Pi \setminus\!\!\setminus y) \uplus \Delta) \uplus \Sigma$.

All other typing rules are not possible as the final rule in Φ. In particular, rule many is not possible because the term in its final judgement has to be an abstraction, not an application term.

– *Root step for* $\to_{\mathtt{e}}$. Let $t = E\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S\langle v\rangle] \mapsto_{\mathtt{e}} S\langle E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle$. We can infer from Lemma 17 that Φ can only have ES as its last typing rule, and so can only be of the form

$$\dfrac{\Phi_{E\langle\!\langle x\rangle\!\rangle} \triangleright_{\text{CbNeed}}\ x\colon O; \Pi \vdash^{(m',e')} E\langle\!\langle x\rangle\!\rangle : M \quad \Phi_{S\langle v\rangle} \triangleright_{\text{CbNeed}}\ \Delta \vdash^{(m'',e'')} S\langle v\rangle : O \quad O \neq \mathbf{0}}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} E\langle\!\langle x\rangle\!\rangle[x{\leftarrow}S\langle v\rangle] : M}\ \text{ES}$$

Note that $x \notin \mathtt{dom}(\Delta)$, because otherwise Lemma 17 would imply $x \in \mathtt{fv}(S\langle v\rangle)$ and this cannot be the case, given that we are working up to α-equivalence. We now proceed to prove by induction on S that whenever we have $\Phi_{E\langle\!\langle x\rangle\!\rangle}$ and $\Phi_{S\langle v\rangle}$ we can derive $\Phi' \triangleright_{\text{CbNeed}}\ \Pi \biguplus \Delta \vdash^{(m'+m'',e'+e''-1)} S\langle E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle : M$.

- Let $S \coloneqq \langle\cdot\rangle$. First of all, applying Lemma 6 on $\Phi_{E\langle\!\langle x\rangle\!\rangle}$ yields a splitting $O = O_1 \uplus O_2$ such that for every derivation $\Psi \triangleright_{\text{CbNeed}}\ \Sigma \vdash^{(m''',e''')} v : O_1$ there exists a derivation $x\colon O_2; \Pi \biguplus \Sigma \vdash^{(m'+m''',e'+e'''-1)} E\langle\!\langle v\rangle\!\rangle : M$. In particular, if $O_2 = \mathbf{0}$, then we can construct the desired derivation Φ' as follows

$$\dfrac{\dfrac{x\colon O; \Pi \vdash^{(m',e')} E\langle\!\langle x\rangle\!\rangle : M \qquad \Delta \vdash^{(m'',e'')} v : O}{\Pi \biguplus \Delta \vdash^{(m'+m'',e'+e''-1)} E\langle\!\langle v\rangle\!\rangle : M}\ \textit{Lemma 6}}{\Pi \biguplus \Delta \vdash^{(m'+m'',e'+e''-1)} E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v] : M}\ \text{ES}_{\text{gc}}$$

On the other hand, if $O_2 \neq \mathbf{0}$, we can then apply Lemma 18 on $\Phi_{S\langle v\rangle}$ to yield derivations $\Phi_{O_1} \triangleright_{\text{CbNeed}}\ \Delta_{O_1} \vdash^{(m''_{O_1},e''_{O_1})} v : O_1$ and $\Phi_{O_2} \triangleright_{\text{CbNeed}}\ \Delta_{O_2} \vdash^{(m''_{O_2},e''_{O_2})} v : O_2$ such that $\Delta = \Delta_{O_1} \uplus \Delta_{O_2}$ and $(m'', e'') = (m''_{O_1} + m''_{O_2}, e''_1 + e''_{O_2})$. Thus, we are now able to combine all these derivations to construct Φ' as follows

$$\dfrac{\dfrac{x\colon O;\Pi \vdash^{(m',e')} E\langle\!\langle x\rangle\!\rangle : M \quad \Delta_{O_1} \vdash^{(m''_{O_1},e''_{O_1})} v : O_1}{x\colon O_2;\Pi \uplus \Delta_{O_1} \vdash^{(m'+m''_{O_1},e'+e''_{O_1}-1)} E\langle\!\langle v\rangle\!\rangle : M}\,L.6 \qquad \Delta_{O_2} \vdash^{(m''_{O_2},e''_{O_2})} S\langle v\rangle : O_2}{\Pi \uplus \Delta_{O_1} \uplus \Delta_{O_2} \vdash^{(m'+m''_{O_1}+m''_{O_2},e'+e''_{O_1}-1+e''_{O_2})} E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v] : M}\,\text{ES}$$

- Let $S := S'[y{\leftarrow}t]$. There are two possible final typing rules in $\Phi_{S\langle v\rangle}$, namely ES and ES_{gc}.
  - ∗ Let $\Phi_{S\langle v\rangle}$ be of the form

    $$\dfrac{\Delta \vdash^{(m'',e'')} S'\langle v\rangle : O \quad \Delta(y) = \mathbf{0}}{\Delta \vdash^{(m'',e'')} S\langle v\rangle : O}\,\text{ES}_{\text{gc}}$$

    Note that since we are working up to α-equivalence we can safely assume that $y \notin \mathtt{fv}(E\langle\!\langle x\rangle\!\rangle)$, and so via Lemma 17 we have that $y \notin \mathtt{dom}(\Pi)$. We can then construct Φ' by application of the *i.h.* as follows

    $$\dfrac{\dfrac{x\colon O;\Pi \vdash^{(m',e')} E\langle\!\langle x\rangle\!\rangle : M \quad \Delta \vdash^{(m'',e'')} S'\langle v\rangle : O}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} S'\langle E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle : M}\,\textit{i.h.}}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} S\langle E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle : M}\,\text{ES}_{\text{gc}}$$

  - ∗ Let $\Phi_{S\langle v\rangle}$ be of the form

    $$\dfrac{y\colon P;\Delta_1 \vdash^{(m''_1,e''_1)} S'\langle v\rangle : O \quad \Delta_2 \vdash^{(m''_2,e''_2)} t : P \quad P \neq \mathbf{0}}{\Delta_1 \uplus \Delta_2 \vdash^{(m''_1+m''_2,e''_1+e''_2)} S\langle v\rangle : O}\,\text{ES}$$

    where $\Delta = \Delta_1 \uplus \Delta_2$ and $(m'',e'') = (m''_1+m''_2, e''_1+e''_2)$. Note that $y \notin \mathtt{fv}(E\langle\!\langle x\rangle\!\rangle)$, and so via Lemma 17 we have that $y \notin \mathtt{dom}(\Pi)$.
    We can then construct Φ' by application of the *i.h.* and a rearranging of Φ as follows

    $$\dfrac{\dfrac{x\colon O;\Pi \vdash^{(m',e')} E\langle\!\langle x\rangle\!\rangle : M \quad y\colon P;\Delta_1 \vdash^{(m''_1,e''_1)} S'\langle v\rangle : O}{y\colon P;\Pi \uplus \Delta_1 \vdash^{(m'+m''_1,e'+e''_1-1)} S'\langle E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle : M}\,\textit{i.h.} \qquad \Delta_2 \vdash^{(m''_2,e''_2)} t : P}{\Pi \uplus \Delta_1 \uplus \Delta_2 \vdash^{(m'+m''_1+m''_2,e'+e''_1-1+e''_2)} S\langle E\langle\!\langle v\rangle\!\rangle[x{\leftarrow}v]\rangle : M}\,\text{ES}$$

– *Contextual closure.* We proceed by induction on the derivation of $t = E\langle t_1\rangle \rightarrow_{\mathtt{nd}} E\langle t_2\rangle = s$:
- If $E = \langle\,\rangle$, then $t \mapsto_{\mathtt{m}} s$ or $t \mapsto_{\mathtt{e}} s$, and the statement holds as we have just proved.
- Let $E = E_1 u$. This implies that the last typing rule in Φ is either $\mathtt{app}_{\text{gc}}$ or $\mathtt{app}_b$. We will only cover the case where $E\langle t_1\rangle \rightarrow_{\mathtt{m}} E\langle t_2\rangle$ and Φ ends in rule $\mathtt{app}_b$, leaving the rest of the (analogous) cases to the reader.
  Now, Φ is of the form

  $$\dfrac{\Pi \vdash^{(m',e')} E_1\langle t_1\rangle : [N \multimap M] \quad \Delta \vdash^{(m'',e'')} u : N}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} E_1\langle t_1\rangle u : M}\,\mathtt{app}_b$$

Then we apply *i.h.* on the left premise of the last rule, obtaining a type derivation whose final judgement is $\Pi \vdash^{(m'-1,e')} E_1\langle t_2\rangle : [N \multimap M]$, thus allowing us to construct Φ' as folows:

$$\dfrac{\Pi \vdash^{(m'-1,e')} E_1\langle t_2\rangle : [N \multimap M] \quad \Delta \vdash^{(m'',e'')} u : N}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} E_1\langle t_2\rangle u : M}\ \texttt{app}_b$$

Note that $(m' + m'', e' + e'') = (m - 1, e)$.

- Let $E = E_1[x\leftarrow u]$. This implies that the last typing rule in Φ is either $\mathrm{ES_{gc}}$ or ES. We will only cover the case where $E\langle t_1\rangle \rightarrow_{\mathtt{m}} E\langle t_2\rangle$ and Φ ends in rule ES, leaving the rest of the (analogous) cases to the reader.
Now, Φ is of the form

$$\dfrac{\Pi \vdash^{(m',e')} E_1\langle t_1\rangle : M \quad \Delta \vdash^{(m'',e'')} u : \Pi(x) \quad \Pi(x) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus x) \uplus \Delta \vdash^{(m'+m'',e'+e'')} E_1\langle t_1\rangle[x\leftarrow u] : M}\ \mathrm{ES}$$

Applying *i.h.* on the left premise of the last rule yields a derivation whose final judgement is $\Pi \vdash^{(m'-1,e')} E_1\langle t_2\rangle : M$, thus allowing us to construct Φ' as follows:

$$\dfrac{\Pi \vdash^{(m'-1,e')} E_1\langle t_2\rangle : M \quad \Delta \vdash^{(m'',e'')} u : \Pi(x) \quad \Pi(x) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus x) \uplus \Delta \vdash^{(m'-1+m'',e'+e'')} E_1\langle t_2\rangle[x\leftarrow u] : M}\ \mathrm{ES}$$

Note that $(m' - 1 + m'', e' + e'') = (m - 1, e)$
- Let $E = E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2]$. We will only consider the case where

$$E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2\langle t_1\rangle] \rightarrow_{\mathtt{m}} E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2\langle t_2\rangle]$$

leaving the other (analogous) case to the reader.
First of all, Lemma 17 implies that the last rule in Φ is $\mathtt{ES}$; *i.e.*, Φ is of the form

$$\dfrac{\Pi \vdash^{(m',e')} E_1\langle x\rangle : M \quad \Delta \vdash^{(m'',e'')} E_2\langle t_1\rangle : \Pi(x) \quad \Pi(x) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus x) \uplus \Delta \vdash^{(m'+m'',e'+e'')} E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2\langle t_1\rangle] : M}\ \mathrm{ES}$$

Applying now the *i.h.* on the premise in the middle of the last rule yields a derivation with conclusion $\Delta \vdash^{(m''-1,e'')} E_2\langle t_2\rangle : \Pi(x)$, thus allowing us to construct Φ' as follows

$$\dfrac{\Pi \vdash^{(m',e')} E_1\langle\!\langle x\rangle\!\rangle : M \quad \Delta \vdash^{(m''-1,e'')} E_2\langle t_2\rangle : \Pi(x) \quad \Pi(x) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus x) \uplus \Delta \vdash^{(m'+m''-1,e'+e'')} E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2\langle t_2\rangle] : M}\ \mathrm{ES}$$

verifying that $(m' + m'' - 1, e' + e'') = (m - 1, e)$. □

Proposition 24 ($[\mathsf{normal}]$ typings for normal forms for CbNeed). *Let $\Phi \triangleright_{\text{need}} \Gamma \vdash^{(m,e)} t : [\mathsf{normal}]$ be a derivation, with $\mathsf{normal}(t)$. Then Γ is empty, and so Φ is tight, and $m = e = 0$.*

Proof. By induction on $\mathsf{normal}(t)$.

– If $\mathsf{normal}(t)$ because $t = \lambda x.s$ then Φ can only be of the form

$$\dfrac{\dfrac{}{\vdash^{(0,0)} \lambda x.s : \mathsf{normal}}\ \mathsf{normal}}{\vdash^{(0,0)} \lambda x.s : [\mathsf{normal}]}\ \mathsf{many}$$

– If $\mathsf{normal}(t)$ because $t = s[y \leftarrow u]$ and $\mathsf{normal}(s)$ then, in principle, there are two possible last typing rules to Φ, namely ES and ES_{gc}. If we assume that it is ES, then Φ is of the form

$$\dfrac{y : O; \Pi_1 \vdash^{(m_1',e_1')} s : [\mathsf{normal}] \quad \Pi_2 \vdash^{(m_2',e_2')} u : O \quad O \neq \mathbf{0}}{\Pi_1 \uplus \Pi_2 \vdash^{(m_1'+m_2',e_1'+e_2')} s[y \leftarrow u] : [\mathsf{normal}]}\ \text{ES}$$

with $\Pi = \Pi_1 \uplus \Pi_2$ and $(m', e') = (m_1' + m_2', e_1' + e_2')$. However, application of the *i.h.* on the left-hand side premise gives that $y : O; \Pi_1$ is empty, in turn implying that $O = \mathbf{0}$, which is in contradiction with the constraints of the ES typing rule.

Therefore, ES could not be the last typing rule of Φ, and so the latter can only be of the form

$$\dfrac{\Pi \vdash^{(m',e')} s : [\mathsf{normal}] \quad y \notin \mathtt{dom}(\Pi)}{\Pi \vdash^{(m',e')} s[y \leftarrow u] : [\mathsf{normal}]}\ \text{ES}_{\text{gc}}$$

Finally, it suffices to apply *i.h.* on the premise to obtain that Π is empty and $m' = e' = 0$. □

Theorem 12 (CbNeed tight correctness). *Let t be a closed term. If $\Phi \triangleright_{\text{need}} \vdash^{(m,e)} t : M$ then there is s such that $d : t \rightarrow^*_{\text{need}} s$, $\mathsf{normal}(s)$, $|d|_{\mathtt{m}} \leq m$, $|d|_{\mathtt{e}} \leq e$. Moreover, if Φ is tight then $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$.*

Proof. By induction on $m + e$ and case analysis on whether t reduces or not. If t is in $\rightarrow_{\text{need}}$-normal form, then we only have to prove the *moreover* part, which states that if Φ is $\mathtt{tight}$ then $m = e = 0$, which follows from Proposition 11.

Otherwise, there are 2 cases:

1. *Multiplicative steps*: If $t \rightarrow_{\mathtt{m}_{\text{need}}} u$, then by Quantitative Subject Reduction For CbNeed (Proposition 10) there exists a typing derivation $\Psi \triangleright_{\text{CbNeed}} \Gamma \vdash^{(m-1,e)} u : M$. By *i.h.* there exist s and d' such that $\mathsf{normal}(s)$, $d' : u \rightarrow^*_{\text{need}} s$, $|d'|_m \leq m - 1$, and $|d'|_e \leq e$. Just note that $t \rightarrow_{\mathtt{m}_{\text{need}}} u$ and so, since d' is preceeded by such a step, then we have $|d|_m = |d'|_m + 1 \leq m$ and $|d|_e = |d'|_e \leq e$.
   If Φ is $\mathtt{tight}$ then so is Ψ. Then by *i.h.* $|d'|_m = m - 1$ and $|d'|_e = e$, which finally implies that $|d|_m = |d'|_m + 1 = m$ and $|d|_e = |d'|_e = e$.
2. *Exponential steps*: If $t \rightarrow_{\mathtt{e}_{\text{need}}} u$, then by Quantitative Subject Reduction (Proposition 10) there exists a typing derivation $\Psi \triangleright_{\text{CbNeed}} \Gamma \vdash^{(m,e-1)} u : M$. By *i.h.* there exists s and d' such that $\mathsf{normal}(s)$, $d' : u \rightarrow^*_{\text{need}} s$, $|d'|_m \leq m$, and $|d'|_e \leq e - 1$. Just note that $t \rightarrow_{\mathtt{e}_{\text{need}}} u$ and so, since d' is preceeded by such a step, we have $|d|_m = |d'|_m \leq m$ and $|d|_e = |d'|_e + 1 \leq e$.
   If Φ is $\mathtt{tight}$ then so is Ψ. Then by *i.h.* $|d'|_m = m$ and $|d'|_e = e - 1$, which finally implies that $|d|_m = |d'|_m = m$ and $|d|_e = |d'|_e + 1 = e$. □

D.2 CbNeed Completeness

Proposition 25 (Normal forms are tightly typable for CbNeed). *Let t be such that $\mathsf{normal}(t)$. Then there is a tight derivation $\Phi \triangleright_{\text{need}} \vdash^{(0,0)} t : [\mathsf{normal}]$.*

Proof. We can easily prove by induction on $\mathsf{normal}()$ that if $\mathsf{normal}(t)$ then $t = S\langle \lambda x.s\rangle$, for some abstraction $\lambda x.s$ and substitution context $S = \langle\cdot\rangle[x_1{\leftarrow}t_1]...[x_n{\leftarrow}t_n]$, with $n \geq 0$.

Therefore, we can derive Φ as follows

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\vdash^{(0,0)} \lambda x.s : \mathsf{normal}}\ \mathsf{normal}}{\vdash^{(0,0)} \lambda x.s : [\mathsf{normal}]}\ \mathsf{many}}{\vdash^{(0,0)} \lambda x.s[x_1{\leftarrow}t_1] : [\mathsf{normal}]}\ \mathrm{ES_{gc}}}{\cdots}\ \mathrm{ES_{gc}}}{\vdash^{(0,0)} \lambda x.s[x_1{\leftarrow}t_1][x_n{\leftarrow}t_n] : [\mathsf{normal}]}\ \mathrm{ES_{gc}}$$

□

Lemma 20 (Merging of multi-sets with respect to derivations). *Given a value v, for any two derivations $\Phi_N \triangleright_{CbNeed} \Gamma_N \vdash^{(m_N,e_N)} v : N$ and $\Phi_O \triangleright_{CbNeed} \Gamma_O \vdash^{(m_O,e_O)} v : O$, there is a derivation $\Phi_{N \uplus O} \triangleright_{CbNeed} \Gamma_N \uplus \Gamma_O \vdash^{(m_N+m_O, e_N+e_O)} v : N \uplus O$.*

Proof. Among the different rules that type abstractions (namely normal, $\mathtt{fun}$ and many), only rule many types them with a multi type. Thus, by properly defining J and K such that $N = [L_j]_{j\in J}$ and $O = [L_k]_{k\in K}$, we have that Φ_N is of the form

$$\cfrac{(\Gamma_j \vdash^{(m_j,e_j)} v : L_j)_{j\in J}}{\biguplus_{j\in J} \Gamma_j \vdash^{(\sum_{j\in J} m_j, \sum_{j\in J} e_j)} v : [L_j]_{j\in J}}\ \mathsf{many}$$

and Φ_O is of the form

$$\cfrac{(\Gamma_k \vdash^{(m_k,e_k)} v : L_k)_{k\in K}}{\biguplus_{k\in K} \Gamma_k \vdash^{(\sum_{k\in K} m_k, \sum_{k\in K} e_k)} v : [L_k]_{k\in K}}\ \mathsf{many}$$

Therefore, we define $I = J \cup K$ and finally obtain $\Phi_{N\uplus O}$ as follows

$$\cfrac{(\Gamma_i \vdash^{(m_i,e_i)} v : L_i)_{i\in I}}{\biguplus_{i\in I} \Gamma_i \vdash^{(\sum_{i\in I} m_i, \sum_{i\in I} e_i)} v : [L_i]_{i\in I}}\ \mathsf{many}$$

□

Lemma 21 (Linear removal for CbNeed). *Let $\Phi \triangleright_{\text{need}} \Gamma \vdash^{(m,e)} E\langle\langle v\rangle\rangle : O$ be a derivation, with $O \neq \mathbf{0}$ and $x \notin \mathtt{fv}(v)$. Then there exist*
- *a multi type M,*
- *a derivation $\Phi_v \triangleright_{\text{need}} \Gamma_v \vdash^{(m_v,e_v)} v : M$, and*
- *a derivation $\Phi_{E\langle\langle x\rangle\rangle} \triangleright_{\text{need}} \Gamma' \uplus \{x : M\} \vdash^{(m',e')} E\langle\langle x\rangle\rangle : O$*

such that
- Type contexts*:* $\Gamma = \Gamma' \uplus \Gamma_v$*.*
- Indices*:* $(m, e) = (m' + m_v, e' + e_v - 1)$*.*

Proof. We prove this by induction on the context E:
- Let $E = \langle\,\rangle$. Note that $O \neq \mathbf{0}$, a fact that is verifiable simply by checking the typing rules for abstractions. Now, by taking $\Gamma_v \coloneqq \Gamma$, $\Gamma' \coloneqq \emptyset$, $M \coloneqq O$, $(m_v, e_v) \coloneqq (m, e)$, and $(m', e') \coloneqq (0, 1)$, we can then take $\Phi_v \coloneqq \Phi$ and construct $\Phi_{E\langle\!\langle x\rangle\!\rangle}$ as follows:

$$\frac{}{x\colon M \vdash^{(0,1)} x : M}\,\mathsf{ax}$$

verifying that $(m, e) = (m_v, e_v) = (0 + m_v, 1 + e_v - 1) = (m' + m_v, e' + e_v - 1)$ and $\Gamma = \emptyset \uplus \Gamma = \Gamma' \uplus \Gamma_v$.
- Let $E = E_1 t$, and so $E\langle\!\langle v\rangle\!\rangle = E_1\langle\!\langle v\rangle\!\rangle t$. There are two possible last rules in Φ, namely app_b or $\mathsf{app}_{\mathrm{gc}}$.
Let us assume Φ is of the form

$$\frac{\Phi_{E_1\langle\!\langle v\rangle\!\rangle} \triangleright_{\mathrm{CbNeed}} \Pi \vdash^{(m_\Pi, e_\Pi)} E_1\langle\!\langle v\rangle\!\rangle : [O' \multimap O] \quad \Delta \vdash^{(m_\Delta, e_\Delta)} t : O' \quad O' \neq \mathbf{0}}{\Pi \uplus \Delta \vdash^{(m_\Pi + m_\Delta + 1, e_\Pi + e_\Delta)} E_1\langle\!\langle v\rangle\!\rangle t : O}\,\mathsf{app}_b$$

Then we can apply the *i.h.* on $\Phi_{E_1\langle\!\langle v\rangle\!\rangle}$ to obtain a type M and typing derivations

$$\Phi_v \triangleright_{\mathrm{CbNeed}} \Pi_v \vdash^{(m_v, e_v)} v : M$$

and

$$\Phi_{E_1\langle\!\langle x\rangle\!\rangle} \triangleright_{\mathrm{CbNeed}} \Pi' \biguplus \{x\colon M\} \vdash^{(m'', e'')} E_1\langle\!\langle x\rangle\!\rangle : [O' \multimap O]$$

such that $\Pi = \Pi' \uplus \Pi_v$ and $(m_\Pi, e_\Pi) = (m'' + m_v, e'' + e_v - 1)$.
Thus, we are able to construct $\Phi_{E\langle\!\langle x\rangle\!\rangle}$ as follows

$$\frac{\Phi_{E_1\langle\!\langle x\rangle\!\rangle} \triangleright_{\mathrm{CbNeed}} \Pi' \uplus \{x\colon M\} \vdash^{(m'', e'')} E_1\langle\!\langle x\rangle\!\rangle : [O' \multimap O] \quad \Delta \vdash^{(m_\Delta, e_\Delta)} t : O' \quad O' \neq \mathbf{0}}{\Pi' \uplus \{x\colon M\} \uplus \Delta \vdash^{(m'' + m_\Delta + 1, e'' + e_\Delta)} E_1\langle\!\langle x\rangle\!\rangle t : O}\,\mathsf{app}_b$$

and, by taking $\Gamma' \coloneqq \Pi' \uplus \Delta$, $\Gamma_v \coloneqq \Pi_v$, and $(m', e') \coloneqq (m'' + m_\Delta + 1, e'' + e_\Delta)$, then we verify that

$$\Gamma = \Pi \biguplus \Delta = \Pi' \biguplus \Pi_v \biguplus \Delta = \Gamma' \biguplus \Gamma_v$$

and

$$(m, e) = (m_\Pi + m_\Delta + 1, e_\Pi + e_\Delta) = (m'' + m_v + m_\Delta + 1, e'' + e_v - 1 + e_\Delta) = (m' + m_v, e' + e_v - 1)$$

Now, let us assume Φ is of the form

$$\frac{\Phi_{E_1\langle\!\langle v\rangle\!\rangle} \triangleright_{\mathrm{CbNeed}} \Gamma \vdash^{(m-1, e)} E_1\langle\!\langle v\rangle\!\rangle : [\mathbf{0} \multimap O]}{\Gamma \vdash^{(m, e)} E_1\langle\!\langle v\rangle\!\rangle t : O}\,\mathsf{app}_{\mathrm{gc}}$$

We then apply the *i.h.* on $\Phi_{E_1\langle\langle v\rangle\rangle}$ to obtain type M and typing derivations

$$\Phi_v \triangleright_{\text{CbNeed}} \Gamma_v \vdash^{(m_v,e_v)} v : M$$

and

$$\Phi_{N_1\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \Gamma' \biguplus \{x\colon M\} \vdash^{(m'',e'')} N_1\langle\langle x\rangle\rangle : [O' \multimap O]$$

such that $\Gamma = \Gamma' \uplus \Gamma_v$ and $(m-1, e) = (m'' + m_v, e'' + e_v - 1)$.
Thus, we are able to construct $\Phi_{E\langle\langle x\rangle\rangle}$ as follows

$$\dfrac{\Phi_{E_1\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \Gamma' \uplus \{x\colon M\} \vdash^{(m'',e'')} E_1\langle\langle x\rangle\rangle : [O' \multimap O]}{\Gamma' \uplus \{x\colon M\} \vdash^{(m''+1,e'')} E_1\langle\langle x\rangle\rangle t : O}\ \mathtt{app}_{\text{gc}}$$

and, by taking $(m', e') = (m'' + 1, e'')$, then verify that

$$(m, e) = (m'' + m_v + 1, e'' + e_v - 1) = (m' + m_v, e' + e_v - 1)$$

Finally, if Φ is $\mathtt{tight}$

– Let $E = E_1[y{\leftarrow}t]$, and so $E\langle\langle v\rangle\rangle = E_1\langle\langle v\rangle\rangle[y{\leftarrow}t]$. Note that we can safely assume that $x \neq y$, since we are working up to α-equivalence and y has a binding occurrence in E while x represents a free variable. Moreover, note that $y \notin \mathtt{fv}(v)$, since otherwise $E\langle\langle v\rangle\rangle$ would not be well-defined.
There are two possible last rules in Φ, namely ES or ES$_{\text{gc}}$.
Let Φ be of the form

$$\dfrac{\Phi_{E_1\langle\langle v\rangle\rangle} \triangleright_{\text{CbNeed}} \Pi \vdash^{(m_\Pi,e_\Pi)} E_1\langle\langle v\rangle\rangle : O \quad \Delta \vdash^{(m_\Delta,e_\Delta)} t : \Pi(y) \quad \Pi(y) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus y) \uplus \Delta \vdash^{(m_\Pi+m_\Delta,e_\Pi+e_\Delta)} E_1\langle\langle v\rangle\rangle[y{\leftarrow}t] : O}\ \text{ES}$$

where $\Gamma = (\Pi \setminus\!\!\setminus y) \uplus \Delta$, $(m, e) = (m_\Pi + m_\Delta, e_\Pi + e_\Delta)$.
Then we can apply the *i.h.* on $\Phi_{E_1\langle\langle v\rangle\rangle}$ to obtain type M and typing derivations

$$\Phi_v \triangleright_{\text{CbNeed}} \Pi_v \vdash^{(m_v,e_v)} v : M$$

and

$$\Phi_{E_1\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \Pi' \biguplus \{x\colon M\} \vdash^{(m'',e'')} E_1\langle\langle x\rangle\rangle : O$$

such that $\Pi = \Pi' \uplus \Pi_v$ and $(m_\Pi, e_\Pi) = (m'' + m_v, e'' + e_v - 1)$.
Moreover, since $y \neq x$ and $y \notin \mathtt{dom}(\Pi_v)$ (otherwise Lemma 17 would imply that $y \in \mathtt{fv}(v)$, which we already know not to be the case), then $(\Pi' \uplus \{x\colon M\})(y) = \Pi(y)$ and so we are able to construct $\Phi_{E\langle\langle x\rangle\rangle}$ as follows

$$\dfrac{\Phi_{E_1\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \Pi' \uplus \{x\colon M\} \vdash^{(m'',e'')} E_1\langle\langle x\rangle\rangle : O \quad \Delta \vdash^{(m_\Delta,e_\Delta)} t : \Pi(y) \quad \Pi(y) \neq \mathbf{0}}{((\Pi' \uplus \{x\colon M\}) \setminus\!\!\setminus y) \uplus \Delta \vdash^{(m''+m_\Delta,e''+e_\Delta)} E_1\langle\langle x\rangle\rangle[y{\leftarrow}t] : O}\ \text{ES}$$

Now, by taking $\Gamma' := (\Pi' \setminus\!\!\setminus y) \uplus \Delta$, $\Gamma_v := \Pi_v$, and $(m', e') := (m'' + m_\Delta, e'' + e_\Delta)$, we can verify that

$$\Gamma = (\Pi \setminus\!\!\setminus y) \biguplus \Delta = ((\Pi' \biguplus \Pi_v) \setminus\!\!\setminus y) \biguplus \Delta = (\Pi' \setminus\!\!\setminus y) \biguplus \Pi_v \biguplus \Delta = \Gamma' \biguplus \Gamma$$

and

$$(m, e) = (m_\Pi{+}m_\Delta, e_\Pi{+}e_\Delta) = ((m''{+}m_v){+}m_\Delta, (e''{+}e_v{-}1){+}e_\Delta) = (m'{+}m_v, e'{+}e_v)$$

If Φ is instead of the form

$$\dfrac{\Phi_{E_1\langle\langle v\rangle\rangle} \triangleright_{\text{CbNeed}} \ \Gamma \vdash^{(m,e)} E_1\langle\langle v\rangle\rangle : O \quad \Gamma(y) = \mathbf{0}}{\Gamma \vdash^{(m,e)} E_1\langle\langle v\rangle\rangle[y{\leftarrow}t] : O} \ \text{ES}_{\text{gc}}$$

then we can apply the *i.h.* on $\Phi_{E_1\langle\langle v\rangle\rangle}$ to obtain a type M and typing derivations

$$\Phi_v \triangleright_{\text{CbNeed}} \ \Pi_v \vdash^{(m_v,e_v)} v : M$$

and

$$\Phi_{E_1\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \ \Gamma' \uplus \{x\colon M\} \vdash^{(m',e')} E_1\langle\langle x\rangle\rangle : O$$

such that $\Gamma = \Gamma' \uplus \Gamma_v$ and $(m, e) = (m' + m_v, e' + e_v - 1)$. Note that this type context and these indices are exactly as desired, and so we can finally construct $\Phi_{E\langle\langle x\rangle\rangle}$ as follows:

$$\dfrac{\Phi_{E_1\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \ \Gamma' \uplus \{x\colon M\} \vdash^{(m',e')} E_1\langle\langle x\rangle\rangle : O}{\Gamma' \uplus \{x\colon M\} \vdash^{(m',e')} E_1\langle\langle x\rangle\rangle[y{\leftarrow}t] : O} \ \text{ES}_{\text{gc}}$$

– Let $E = E_1\langle\langle y\rangle\rangle[y{\leftarrow}E_2]$, and so $E\langle\langle v\rangle\rangle = E_1\langle\langle y\rangle\rangle[y{\leftarrow}E_2\langle\langle v\rangle\rangle]$. Once again, we will assume $x \neq y$. Lemma 17 implies there is only one possible form of Φ, namely:

$$\dfrac{\Pi \vdash^{(m_\Pi,e_\Pi)} E_1\langle\langle y\rangle\rangle : O \quad \Phi_{E_2\langle\langle v\rangle\rangle} \triangleright_{\text{CbNeed}} \ \Delta \vdash^{(m_\Delta,e_\Delta)} E_2\langle\langle v\rangle\rangle : \Pi(y) \quad \Pi(y) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus y) \uplus \Delta \vdash^{(m_\Pi+m_\Delta,e_\Pi+e_\Delta)} E_1\langle\langle y\rangle\rangle[y{\leftarrow}E_2\langle\langle v\rangle\rangle] : O} \ \text{ES}$$

where $\Gamma = (\Pi \setminus\!\!\setminus y) \uplus \Delta$, $(m, e) = (m_\Pi + m_\Delta, e_\Pi + e_\Delta)$.
Then we can apply the *i.h.* on $\Phi_{E_2\langle\langle v\rangle\rangle}$ to obtain type M and typing derivations

$$\Phi_v \triangleright_{\text{CbNeed}} \ \Delta_v \vdash^{(m_v,e_v)} v : M$$

and

$$\Phi_{E_2\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \ \Delta' \uplus \{x\colon M\} \vdash^{(m'',e'')} E_2\langle\langle x\rangle\rangle : \Pi(y)$$

such that $\Delta = \Delta' \uplus \Delta_v$ and $(m_\Delta, e_\Delta) = (m'' + m_v, e'' + e_v - 1)$.
We can then construct $\Phi_{E\langle\langle x\rangle\rangle}$ as follows

$$\dfrac{\Pi \vdash^{(m_\Pi,e_\Pi)} E_1\langle\langle y\rangle\rangle : O \quad \Phi_{E_2\langle\langle x\rangle\rangle} \triangleright_{\text{CbNeed}} \ \Delta' \uplus \{x\colon M\} \vdash^{(m'',e'')} E_2\langle\langle x\rangle\rangle : \Pi(y) \quad \Pi(y) \neq \mathbf{0}}{(\Pi \setminus\!\!\setminus y) \uplus \Delta' \vdash^{(m_\Pi+m'',e_\Pi+e'')} E_1\langle\langle y\rangle\rangle[y{\leftarrow}E_2\langle\langle x\rangle\rangle] : O} \ \text{ES}$$

and, by taking $\Gamma' := (\Pi \setminus\!\!\setminus y) \uplus \Delta'$, $\Gamma_v := \Delta_v$, $(m', e') = (m_\Pi + m'', e_\Pi + e'')$, then verify that

$$\Gamma = (\Pi \setminus\!\!\setminus y) \uplus \Delta = (\Pi \setminus\!\!\setminus y) \uplus (\Delta' \uplus \Delta_v) = \Gamma' \uplus \Gamma_v$$

and

$$(m,e) = (m_\Pi{+}m_\Delta, e_\Pi{+}e_\Delta) = (m_\Pi{+}(m''{+}m_v), e_\Pi{+}(e''{+}e_v{-}1)) = (m'{+}m_v, e'{+}e_v{-}1)$$

□

Proposition 26 (Quantitative subject expansion for CbNeed). *Let $\Phi \triangleright_{\text{need}} \Gamma \vdash^{(m,e)} s : M$ be a derivation such that $M \neq \mathbf{0}$. Then,*

- Multiplicative: *if $t \to_{\mathtt{m}_{\text{need}}} s$ then there is a derivation $\Phi' \triangleright_{\text{need}} \Gamma \vdash^{(m+1,e)} t : M$,*
- Exponential*: if $t \to_{\mathtt{e}_{\text{need}}} s$ then there is a derivation $\Phi' \triangleright_{\text{need}} \Gamma \vdash^{(m,e+1)} t : M$.*

Proof. By induction on the derivation $t \to_{\mathtt{nd}} s$, with the root rules $\mapsto_{\mathtt{m}}$ and $\mapsto_{\mathtt{e}}$ as the base case, and the closure by CbNeed contexts of $\mapsto_{\mathtt{nd}}$ as the inductive one.

- *Root step for* $\to_{\mathtt{m}}$. Let $t = S\langle \lambda x.u\rangle r \mapsto_{\mathtt{m}} S\langle u[x{\leftarrow}r]\rangle = s$, and proceed by induction on S:
  - Let $S \coloneqq \langle\cdot\rangle$. Then $t = (\lambda x.u)r$ and $s = u[x{\leftarrow}r]$, and so Φ has either ES or ES$_{\text{gc}}$ as its last typing rule.
    - If Φ is of the form

$$\dfrac{\Gamma \vdash^{(m,e)} u : M \quad \Gamma(x) = \mathbf{0}}{\Gamma \vdash^{(m,e)} u[x{\leftarrow}r] : M}\ \text{ES}_{\text{gc}}$$

then we can construct Φ' as follows

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash^{(m,e)} u : M}{\Gamma \vdash^{(m,e)} \lambda x.u : \mathbf{0} \multimap M}\ \mathtt{fun}}{\Gamma \vdash^{(m,e)} \lambda x.u : [\mathbf{0} \multimap M]}\ !}{\Gamma \vdash^{(m+1,e)} (\lambda x.u)r : M}\ \mathtt{app}_{\text{gc}}$$

    - Let Φ be of the form

$$\dfrac{x \colon O; \Pi \vdash^{(m',e')} u : M \quad \Delta \vdash^{(m'',e'')} r : O \quad O \neq \mathbf{0}}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} u[x{\leftarrow}r] : M}\ \text{ES}$$

where $\Gamma = \Pi \uplus \Delta$ and $(m,e) = (m'+m'', e'+e'')$.
We can then construct Φ' as follows

$$\dfrac{\dfrac{\dfrac{x \colon O; \Pi \vdash^{(m',e')} u : M}{\Pi \vdash^{(m',e')} \lambda x.u : O \multimap M}\ \mathtt{fun}}{\Pi \vdash^{(m',e')} \lambda x.u : [O \multimap M]}\ ! \qquad \Delta \vdash^{(m'',e'')} r : O}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} (\lambda x.u)r : M}\ \mathtt{app}_b$$

  - Let $S \coloneqq S'[y{\leftarrow}q]$. Then $t = (S'\langle \lambda x.u\rangle[y{\leftarrow}q])r$ and $s = S'\langle u[x{\leftarrow}r]\rangle[y{\leftarrow}q]$. Note that since we are working up to α-equivalence, $y \notin \mathtt{fv}(r)$. There are two possible last typing rules of Φ, namely ES and ES$_{\text{gc}}$.

* Let Φ be of the form

$$\dfrac{y\colon O; \Pi \vdash^{(m',e')} S'\langle u[x\leftarrow r]\rangle : M \quad \Delta \vdash^{(m'',e'')} q : O \quad O \neq \mathbf{0}}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} S'\langle u[x\leftarrow r]\rangle[y\leftarrow q] : M}\,\text{ES}$$

where $\Gamma = \Pi \uplus \Delta$ and $(m,e) = (m'+m'', e'+e'')$. We can then apply the *i.h.* on the leftmost premise to obtain a typing derivation $\Phi'_{i.h.} \rhd_{\text{CbNeed}} y\colon O; \Pi \vdash^{(m'+1,e')} S'\langle\lambda x.u\rangle r : M$. We then analyze the two possibilities of the last typing rule in $\Phi'_{i.h.}$, namely $\mathtt{app}_b$ or $\mathtt{app}_{\text{gc}}$

  · Let $\Phi'_{i.h.}$ be of the form

$$\dfrac{y\colon O; \Pi_1 \vdash^{(m'_1,e'_1)} S'\langle\lambda x.u\rangle : [P \multimap M] \quad \Pi_2 \vdash^{(m'_2,e'_2)} r : P \quad P \neq \mathbf{0}}{y\colon O; \Pi \vdash^{(m'+1,e')} S'\langle\lambda x.u\rangle r : M}\,\mathtt{app}_b$$

  where $(m',e') = (m'_1+m'_2, e'_1+e'_2)$, $\Pi = \Pi_1 \uplus \Pi_2$, and $y \notin \mathtt{dom}(\Pi_2)$, since otherwise Lemma 17 would imply $y \in \mathtt{fv}(r)$.
  We can then construct Φ' as follows

$$\dfrac{\dfrac{y\colon O; \Pi_1 \vdash^{(m'_1,e'_1)} S'\langle\lambda x.u\rangle : [P \multimap M] \quad \Delta \vdash^{(m'',e'')} q : O}{\Delta \uplus \Pi_1 \vdash^{(m'_1+m'',e'_1+e'')} S\langle\lambda x.u\rangle : [P \multimap M]}\,\text{ES} \quad \Pi_2 \vdash^{(m'_2,e'_2)} r : P}{\Delta \uplus \Pi_1 \uplus \Pi_2 \vdash^{(m'_1+m''+m'_2+1,e'_1+e''+e'_2)} S\langle\lambda x.u\rangle r : M}\,\mathtt{app}_b$$

  · Let $\Phi'_{i.h.}$ be of the form

$$\dfrac{y\colon O; \Pi \vdash^{(m',e')} S'\langle\lambda x.u\rangle : [\mathbf{0} \multimap M]}{y\colon O; \Pi \vdash^{(m'+1,e')} S'\langle\lambda x.u\rangle r : M}\,\mathtt{app}_{\text{gc}}$$

  We can then construct Φ' as follows

$$\dfrac{\dfrac{y\colon O; \Pi \vdash^{(m',e')} S'\langle\lambda x.u\rangle : [\mathbf{0} \multimap M] \quad \Delta \vdash^{(m'',e'')} q : O}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} S\langle\lambda x.u\rangle : [\mathbf{0} \multimap M]}\,\text{ES}}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} S\langle\lambda x.u\rangle r : M}\,\mathtt{app}_{\text{gc}}$$

* Let Φ be of the form

$$\dfrac{\Gamma \vdash^{(m,e)} S'\langle u[x\leftarrow r]\rangle : M \quad \Gamma(y) = \mathbf{0}}{\Gamma \vdash^{(m,e)} S'\langle u[x\leftarrow r]\rangle[y\leftarrow q] : M}\,\text{ES}_{\text{gc}}$$

We then apply the *i.h.* on the leftmost premise to obtain a typing derivation $\Phi'_{i.h.} \rhd_{\text{CbNeed}} \Gamma \vdash^{(m+1,e)} S'\langle\lambda x.u\rangle r : M$ for which there are two possible last typing rules, namely $\mathtt{app}_b$ and $\mathtt{app}_{\text{gc}}$.

  · Let $\Phi'_{i.h.}$ be of the form

$$\dfrac{\Pi \vdash^{(m',e')} S'\langle\lambda x.u\rangle : [O \multimap M] \quad \Delta \vdash^{(m'',e'')} r : O \quad O \neq \mathbf{0}}{\Gamma \vdash^{(m+1,e)} S'\langle\lambda x.u\rangle r : M}\,\mathtt{app}_b$$

where $\Gamma = \Pi \uplus \Delta$ and $(m, e) = (m' + m'', e' + e'')$.
We can then construct Φ' as follows

$$\dfrac{\dfrac{\Pi \vdash^{(m',e')} S'\langle \lambda x.u\rangle : [O \multimap M]}{\Pi \vdash^{(m',e')} S\langle \lambda x.u\rangle : [O \multimap M]}\ \mathrm{ES}_{\mathrm{gc}} \qquad \Delta \vdash^{(m'',e'')} r : O}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} S\langle \lambda x.u\rangle r : M}\ \mathtt{app}_b$$

· Let $\Phi'_{i.h.}$ be of the form

$$\dfrac{\Gamma \vdash^{(m,e)} S'\langle \lambda x.u\rangle : [\mathbf{0} \multimap M]}{\Gamma \vdash^{(m+1,e)} S'\langle \lambda x.u\rangle r : M}\ \mathtt{app}_{\mathrm{gc}}$$

We can then construct Φ' as follows

$$\dfrac{\dfrac{\Gamma \vdash^{(m,e)} S'\langle \lambda x.u\rangle : [\mathbf{0} \multimap M]}{\Gamma \vdash^{(m,e)} S\langle \lambda x.u\rangle : [\mathbf{0} \multimap M]}\ \mathrm{ES}_{\mathrm{gc}}}{\Gamma \vdash^{(m+1,e)} S\langle \lambda x.u\rangle r : M}\ \mathtt{app}_{\mathrm{gc}}$$

– *Root step for* $\rightarrow_{\mathtt{e}}$. Let $t = E\langle\langle x\rangle\rangle[x{\leftarrow}S\langle v\rangle] \mapsto_{\mathtt{e}} S\langle E\langle\langle v\rangle\rangle[x{\leftarrow}v]\rangle = s$, and proceed by induction on S:
 • Let $S \coloneqq \langle\cdot\rangle$.Then $t = E\langle\langle x\rangle\rangle[x{\leftarrow}v] \mapsto_{\mathtt{e}} E\langle\langle v\rangle\rangle[x{\leftarrow}v] = s$, and so Φ has either ES or $\mathrm{ES}_{\mathrm{gc}}$ as its last typing rule.
  ∗ Let Φ be of the form

$$\dfrac{\Phi_{E\langle\langle v\rangle\rangle} \triangleright_{\mathrm{CbNeed}}\ \Gamma \vdash^{(m,e)} E\langle\langle v\rangle\rangle : M \quad \Gamma(x) = \mathbf{0}}{\Gamma \vdash^{(m,e)} E\langle\langle v\rangle\rangle[x{\leftarrow}v] : M}\ \mathrm{ES}_{\mathrm{gc}}$$

We apply Lemma 7 on $\Phi_{E\langle\langle v\rangle\rangle}$ to obtain typing derivations $\Phi_v \triangleright_{\mathrm{CbNeed}} \Gamma_v \vdash^{(m_v,e_v)} v : O$ and $\Phi_{E\langle\langle x\rangle\rangle} \triangleright_{\mathrm{CbNeed}}\ \Gamma' \uplus \{x\colon O\} \vdash^{(m',e')} E\langle\langle x\rangle\rangle : M$ such that $\Gamma = \Gamma' \uplus \Gamma_v$ and $(m, e) = (m' + m_v, e' + e_v - 1)$. We can then construct Φ' with such derivations as follows

$$\dfrac{\Gamma' \uplus \{x\colon O\} \vdash^{(m',e')} E\langle\langle x\rangle\rangle : M \quad \Gamma_v \vdash^{(m_v,e_v)} v : O}{\Gamma' \uplus \Gamma_v \vdash^{(m'+m_v,e'+e_v)} E\langle\langle x\rangle\rangle[x{\leftarrow}v] : M}\ \mathrm{ES}$$

In particular, note that $(m' + m_v, e' + e_v) = (m, e + 1)$.
  ∗ Let Φ be of the form

$$\dfrac{x\colon O; \Pi \vdash^{(m',e')} E\langle\langle v\rangle\rangle : M \quad \Delta \vdash^{(m'',e'')} v : O \quad O \neq \mathbf{0}}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} E\langle\langle v\rangle\rangle[x{\leftarrow}v] : M}\ \mathrm{ES}$$

We can then apply Lemma 7 on the leftmost premise with respect to x to obtain a multi type P and typing derivations $\Phi_v \triangleright_{\mathrm{CbNeed}} \Pi_v \vdash^{(m_v,e_v)} v : P$ and $\Phi_{E\langle\langle x\rangle\rangle} \triangleright_{\mathrm{CbNeed}}\ \Pi_{E\langle\langle x\rangle\rangle} \uplus \{x\colon P\} \vdash^{(m_{E\langle\langle x\rangle\rangle},e_{E\langle\langle x\rangle\rangle})} E\langle\langle x\rangle\rangle : M$ such that $x\colon O; \Pi = \Pi_v \uplus \Pi_{E\langle\langle x\rangle\rangle}$ and $(m', e') = (m_{E\langle\langle x\rangle\rangle} + m_v, e_{E\langle\langle x\rangle\rangle} + e_v - 1)$. Note how Lemma 17 implies that $x \notin \mathtt{dom}(\Pi_v)$ - given that $x \notin \mathtt{fv}(v)$-, and so $\Phi_{E\langle\langle x\rangle\rangle}$ can be rewritten as $\Phi_{E\langle\langle x\rangle\rangle} \triangleright_{\mathrm{CbNeed}}$

$x\colon O \uplus P; \Pi'_{E\langle\langle x\rangle\rangle} \vdash^{(m_{E\langle\langle x\rangle\rangle}, e_{E\langle\langle x\rangle\rangle})} E\langle\langle x\rangle\rangle : M$, where $\Pi_{E\langle\langle x\rangle\rangle} = x\colon O; \Pi'_{E\langle\langle x\rangle\rangle}$ and so $\Pi = \Pi'_{E\langle\langle x\rangle\rangle} \uplus \Pi_v$.

Furthermore, we can apply Lemma 20 on $\Delta \vdash^{(m'', e'')} v : O$ and Φ_v to obtain a typing derivation $\Phi_{O \uplus P} \triangleright_{\text{CbNeed}} \Delta \uplus \Pi_v \vdash^{(m''+m_v, e''+e_v)} v : O \uplus P$.

Finally, we can construct Φ' as follows

$$\dfrac{x\colon O \uplus P; \Pi'_{E\langle\langle x\rangle\rangle} \vdash^{(m_{E\langle\langle x\rangle\rangle}, e_{E\langle\langle x\rangle\rangle})} E\langle\langle x\rangle\rangle : M \quad \Delta \uplus \Pi_v \vdash^{(m''+m_v, e''+e_v)} v : O \uplus P}{\Pi'_{E\langle\langle x\rangle\rangle} \uplus \Delta \uplus \Pi_v \vdash^{(m_{E\langle\langle x\rangle\rangle}+m''+m_v, e_{E\langle\langle x\rangle\rangle}+e''+e_v)} E\langle\langle x\rangle\rangle[x{\leftarrow}v] : M}\ \text{ES}$$

Note that $(m_{E\langle\langle x\rangle\rangle}+m''+m_v, e_{E\langle\langle x\rangle\rangle}+e''+e_v) = (m'+m'', e'+1+e'') = (m, e+1)$.

- Let $S \coloneqq S[y{\leftarrow}r]$. Then $t = E\langle\langle x\rangle\rangle[x{\leftarrow}S'\langle v\rangle[y{\leftarrow}r]] \mapsto_{\mathtt{e}} S'\langle E\langle\langle v\rangle\rangle[x{\leftarrow}v]\rangle[y{\leftarrow}r] = s$. Note that $y \notin \mathtt{fv}(E\langle\langle x\rangle\rangle)$, since y is bound in $S\langle v\rangle$ and we are working up to α-equivalence. Then, Φ has either ES or ES_{gc} as its last typing rule.
  - $*$ Let Φ be of the form

$$\dfrac{\Gamma \vdash^{(m,e)} S'\langle E\langle\langle v\rangle\rangle[x{\leftarrow}v]\rangle : M \quad \Gamma(y) = \mathbf{0}}{\Gamma \vdash^{(m,e)} S'\langle E\langle\langle v\rangle\rangle[x{\leftarrow}v]\rangle[y{\leftarrow}r] : M}\ \text{ES}_{\text{gc}}$$

We can then apply the *i.h.* on the premise to obtain a typing derivation $\Phi'_{i.h.} \triangleright_{\text{CbNeed}} \Gamma \vdash^{(m,e+1)} E\langle\langle x\rangle\rangle[x{\leftarrow}S'\langle v\rangle] : M$. Moreover, note that $\Phi'_{i.h.}$ can only have ES as its last typing rule, by application of Lemma 17. $\Phi'_{i.h.}$ is hence of the form

$$\dfrac{x\colon O; \Pi \vdash^{(m_{E\langle\langle x\rangle\rangle}, e_{E\langle\langle x\rangle\rangle})} E\langle\langle x\rangle\rangle : M \quad \Phi_{S'\langle v\rangle} \triangleright_{\text{CbNeed}} \Delta \vdash^{(m_{S'\langle v\rangle}, e_{S'\langle v\rangle})} S'\langle v\rangle : O \quad O \neq \mathbf{0}}{\Gamma \vdash^{(m,e+1)} E\langle\langle x\rangle\rangle[x{\leftarrow}S'\langle v\rangle] : M}\ \text{ES}$$

where $\Pi \uplus \Delta = \Gamma$ and $(m, e+1) = (m_{E\langle\langle x\rangle\rangle} + m_{S'\langle v\rangle}, e_{E\langle\langle x\rangle\rangle} + e_{S'\langle v\rangle})$. Since $\Gamma(y) = \mathbf{0}$ then $\Delta(y) = \mathbf{0}$ and we can construct Φ' as follows

$$\dfrac{x\colon O; \Pi \vdash^{(m_{E\langle\langle x\rangle\rangle}, e_{E\langle\langle x\rangle\rangle})} E\langle\langle x\rangle\rangle : M \quad \dfrac{\Delta \vdash^{(m_{S'\langle v\rangle}, e_{S'\langle v\rangle})} S'\langle v\rangle : O \quad \Delta(y) = \mathbf{0}}{\Delta \vdash^{(m_{S'\langle v\rangle}, e_{S'\langle v\rangle})} S'\langle v\rangle[y{\leftarrow}r] : O}\ \text{ES}_{\text{gc}}}{\Pi \uplus \Delta \vdash^{(m_{E\langle\langle x\rangle\rangle}+m_{S'\langle v\rangle}, e_{E\langle\langle x\rangle\rangle}+e_{S'\langle v\rangle})} E\langle\langle x\rangle\rangle[x{\leftarrow}S'\langle v\rangle[y{\leftarrow}r]] : M}\ \text{ES}$$

  - $*$ Let Φ be of the form

$$\dfrac{y\colon O; \Pi \vdash^{(m_1,e_1)} S'\langle E\langle\langle v\rangle\rangle[x{\leftarrow}v]\rangle : M \quad \Delta \vdash^{(m_2,e_2)} r : O}{\Pi \uplus \Delta \vdash^{(m_1+m_2, e_1+e_2)} S'\langle E\langle\langle v\rangle\rangle[x{\leftarrow}v]\rangle[y{\leftarrow}r] : M}\ \text{ES}$$

where $\Pi \uplus \Delta = \Gamma$ and $(m_1 + m_2, e_1 + e_2) = (m, e)$. We can then apply the *i.h.* on the leftmost premise to obtain a typing derivation $\Phi'_{i.h.} \triangleright_{\text{CbNeed}} y\colon O; \Pi \vdash^{(m_1, e_1+1)} E\langle\langle x\rangle\rangle[x{\leftarrow}S'\langle v\rangle] : M$ which has to have ES as its last typing rule -via Lemma 17-, as follows

$$\dfrac{x\colon P; \Pi_1 \vdash^{(m_{1,1}, e_{1,1})} E\langle\langle x\rangle\rangle : M \quad y\colon O; \Pi_2 \vdash^{(m_{1,2}, e_{1,2})} S'\langle v\rangle : P}{y\colon O; \Pi_1 \uplus \Pi_2 \vdash^{(m_{1,1}+m_{1,2}, e_{1,1}+e_{1,2})} E\langle\langle x\rangle\rangle[x{\leftarrow}S'\langle v\rangle] : M}\ \text{ES}$$

where $\Pi_1 \uplus \Pi_2 = \Pi$, $(m_{1,1} + m_{1,2}, e_{1,1} + e_{1,2}) = (m_1, e_1 + 1)$. We then construct Φ' as follows

$$\dfrac{x\colon P; \Pi_1 \vdash^{(m_{1,1},e_{1,1})} E\langle\langle x\rangle\rangle : M \quad \dfrac{y\colon O; \Pi_2 \vdash^{(m_{1,2},e_{1,2})} S'\langle v\rangle : P \quad \Delta \vdash^{(m_2,e_2)} r : O}{\Pi_2 \uplus \Delta \vdash^{(m_{1,2}+m_2,e_{1,2}+e_2)} S'\langle v\rangle[y\leftarrow r] : P}\,\mathrm{ES}}{\Pi_1 \uplus \Pi_2 \uplus \Delta \vdash^{(m_{1,1}+m_{1,2}+m_2,e_{1,1}+e_{1,2}+e_2)} E\langle\langle x\rangle\rangle[x\leftarrow S'\langle v\rangle[y\leftarrow r]] : M}\,\mathrm{ES}$$

Note that $\Pi_1 \uplus \Pi_2 \uplus \Delta = \Pi \uplus \Delta = \Gamma$ and $(m_{1,1} + m_{1,2} + m_2, e_{1,1} + e_{1,2} + e_2) = (m_1 + m_2, e_1 + e_2) = (m, e + 1)$.

– *Contextual closure.* We proceed by induction on the derivation of $t = E\langle t'\rangle \to_{\mathtt{nd}} E\langle s'\rangle = s$:
  - Let $E = \langle\cdot\rangle$. Then $t \mapsto_{\mathtt{m}} s$ or $t \mapsto_{\mathtt{e}} s$ and in either case the statement holds, as we have just proved.
  - Let $E = E_1 u$. Then $\Phi \triangleright_{\mathrm{CbNeed}} \Gamma \vdash^{(m,e)} E_1\langle s'\rangle u : M$ and its last typing rule is either $\mathtt{app}_b$ or $\mathtt{app}_{\mathrm{gc}}$. We will only cover the case where $E\langle t'\rangle \to_{\mathtt{m}} E\langle s'\rangle$ and Φ ends in rule $\mathtt{app}_b$, leaving the rest of the (analogous) cases to the reader.
    Let Φ be of the form

$$\dfrac{\Phi_{E_1\langle s\rangle} \triangleright_{\mathrm{CbNeed}} \ \Pi \vdash^{(m',e')} E_1\langle s'\rangle : [O \multimap M] \quad \Delta \vdash^{(m'',e'')} u : O}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} E_1\langle s'\rangle u : M}\,\mathtt{app}_b$$

where $\Gamma = \Pi \uplus \Delta$, $(m' + m'' + 1, e' + e'') = (m, e)$, and $O \neq \mathbf{0}$.
We can then apply the *i.h.* on $\Phi_{E_1\langle s\rangle}$ to obtain a typing derivation $\Phi'_{i.h.} \triangleright_{\mathrm{CbNeed}} \ \Pi \vdash^{(m+1,e)} E_1\langle t'\rangle : [O \multimap M]$ with which we construct Φ' as follows

$$\dfrac{\Pi \vdash^{(m+1,e)} E_1\langle t'\rangle : [O \multimap M] \quad \Delta \vdash^{(m'',e'')} u : O}{\Pi \uplus \Delta \vdash^{(m'+1+m''+1,e'+e'')} E_1\langle t'\rangle u : M}\,\mathtt{app}_b$$

Note that $(m' + 1 + m'' + 1, e' + 1 + e'') = (m + 1, e)$.
  - Let $E = E_1[x\leftarrow u]$. Then $\Phi \triangleright_{\mathrm{CbNeed}} \ \Gamma \vdash^{(m,e)} E_1\langle s'\rangle[x\leftarrow u] : M$ and its last typing rule is either ES or $\mathrm{ES}_{\mathrm{gc}}$. We will only cover the case where $E\langle t_1\rangle \to_{\mathtt{m}} E\langle s_1\rangle$ and Φ ends in rule ES, leaving the rest of the (analogous) cases to the reader.
    Let Φ be of the form

$$\dfrac{\Phi_{E_1\langle s'\rangle} \triangleright_{\mathrm{CbNeed}} \ x\colon O; \Pi \vdash^{(m',e')} E_1\langle s'\rangle : M \quad \Delta \vdash^{(m'',e'')} u : O}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} E_1\langle s'\rangle[x\leftarrow u] : M}\,\mathrm{ES}$$

where $\Pi \uplus \Delta = \Gamma$, $(m' + m'', e' + e'') = (m, e)$, and $O \neq \mathbf{0}$.
We can then apply the *i.h.* on $\Phi_{E_1\langle s'\rangle}$ to obtain a typing derivation $\Phi'_{i.h.} \triangleright_{\mathrm{CbNeed}} \ x\colon O; \Pi \vdash^{(m'+1,e')} E_1\langle t'\rangle : M$ with which Φ' goes as follows

$$\dfrac{x\colon O; \Pi \vdash^{(m'+1,e')} E_1\langle t'\rangle : M \quad \Delta \vdash^{(m'',e'')} u : O}{\Pi \uplus \Delta \vdash^{(m'+1+m'',e'+e'')} E_1\langle t'\rangle[x\leftarrow u] : M}\,\mathrm{ES}$$

Note that $(m' + 1 + m'', e' + e'') = (m + 1, e)$.

- Let $E = E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2]$. We will only consider the case where

$$t = E\langle t'\rangle = E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2\langle t'\rangle] \to_{\mathtt{m}} E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2\langle s'\rangle] = E\langle s'\rangle = s$$

leaving the (analogous) case when $t \to_{\mathtt{e}} s$ to the reader.
Therefore, Φ is of the form

$$\frac{\Pi \vdash^{(m',e')} E_1\langle\!\langle x\rangle\!\rangle : M \quad \Phi_{E_2\langle s'\rangle} \triangleright_{\text{CbNeed}} \Delta \vdash^{(m'',e'')} E_2\langle s'\rangle : O}{\Pi \uplus \Delta \vdash^{(m'+m'',e'+e'')} E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2\langle s'\rangle] : M}\ \text{ES}$$

where $\Pi \uplus \Delta = \Gamma$, $(m'+m'', e'+e'') = (m,e)$, and $O \neq \mathbf{0}$.
We can then apply the *i.h.* on $\Phi_{E_2\langle s'\rangle}$ to obtain a typing derivation $\Phi'_{i.h.} \triangleright_{\text{CbNeed}} \Delta \vdash^{(m''+1,e'')} E_2\langle t'\rangle : O$, with which we can finally construct Φ' as follows

$$\frac{\Pi \vdash^{(m',e')} E_1\langle\!\langle x\rangle\!\rangle : M \quad \Delta \vdash^{(m''+1,e'')} E_2\langle t'\rangle : O}{\Pi \uplus \Delta \vdash^{(m'+m''+1,e'+e'')} E_1\langle\!\langle x\rangle\!\rangle[x\leftarrow E_2\langle t'\rangle] : M}\ \text{ES}$$

Note that $(m'+m''+1, e'+e'') = (m+1, e)$. □

Theorem 13 (CbNeed tight completeness). *Let t be a closed term. If $d\colon t \to^*_{\text{need}} s$ and $\mathsf{normal}(s)$ then there exists a tight derivation $\Phi \triangleright_{\text{need}} \ \vdash^{(|d|_{\mathtt{m}},|d|_{\mathtt{e}})} t : [\mathsf{normal}]$.*

Proof. By induction on $|d|$; *i.e.*, on k such that $d\colon t \to^k_{\text{need}} s$.

- If $k = 0$, then $t = s$ and Proposition 1 implies that $\mathsf{normal}(t)$. We then obtain Φ as desired via application of Proposition 12.
- If $k > 0$ then $d\colon t \to_{\text{need}} u \to^{k-1}_{\text{need}} s$. We can then apply the *i.h.* on $d'\colon u \to^{k-1}_{\text{need}} s$ to obtain a `tight` derivation $\Phi_{i.h.} \triangleright_{\text{need}} \ \vdash^{(|d'|_m,|d'|_e)} u : [\mathsf{normal}]$.
  Now, if $t \to_{\mathtt{m}_{\text{need}}} u$ then Proposition 13 implies that there exists a derivation as desired, namely $\Phi \triangleright_{\text{need}} \ \vdash^{(|d'|_m+1,|d'|_e)} t : [\mathsf{normal}]$, since $(|d'|_m + 1, |d'|_e) = (|d|_m, |d|_e)$.
  If $t \to_{\mathtt{e}_{\text{need}}} u$ then Proposition 13 implies that there exists a derivation as desired, namely $\Phi \triangleright_{\text{need}} \ \vdash^{(|d'|_m,|d'|_e+1)} t : [\mathsf{normal}]$, since $(|d'|_m, |d'|_e + 1) = (|d|_m, |d|_e)$. □

E A New Fundamental Theorem for Call-by-Need (Sect. 7)

Corollary 3 (CbV correctness wrt CbNeed). *Let t be a closed term and $\Phi \triangleright_{\text{cbv}} \ \vdash^{(m,e)} t : M$ be a derivation. Then there exists s such that $d\colon t \to^*_{\text{need}} s$ and $\mathsf{normal}(s)$, with $|d|_{\mathtt{m}} \leq m$ and $|d|_{\mathtt{e}} \leq e$.*

Proof. By induction on $m + e$ and case analysis on whether t reduces or not. If $\mathsf{normal}(t)$ then the statement holds with $s \coloneqq t$ and d the empty evaluation, so that $|d|_{\mathtt{m}} = 0 = |d|_{\mathtt{e}}$.

Otherwise $\neg\mathsf{normal}(t)$, then $t \to_{\mathrm{need}} u$ according to the syntactic characterization of closed need-normal forms (Proposition 1), since t is closed. As $t \to_{\mathrm{need}} u$ means either $t \to_{\mathtt{m}_{\mathrm{need}}} u$ or $t \to_{\mathtt{e}_{\mathrm{need}}} u$, by quantitative subject reduction for the CbV multi type system with respect to CbNeed evaluation (Proposition **??**) there is $\Psi \triangleright_{\mathrm{cbv}} \Gamma \vdash^{(m',e')} u : M$ with:

- $m' \coloneqq m - 1$ and $e' = e$ if $t \to_{\mathtt{m}_{\mathrm{need}}} u$,
- $m' \coloneqq m$ and $e' = e - 1$ if $t \to_{\mathtt{e}_{\mathrm{need}}} u$.

By *i.h.* (since $m' + e' = m + e - 1$), there is a term s such that $d' \colon u \to^*_{\mathrm{need}} s$ and $\mathsf{normal}(s)$, with $|d'|_{\mathtt{m}} \leq m'$ and $|d'|_{\mathtt{e}} \leq e'$. The evaluation $d \colon t \to^*_{\mathrm{need}} s$ obtained by prefixing d' with the step $t \to_{\mathrm{need}} u$ verifies $|d|_{\mathtt{m}} \leq m$ and $|d|_{\mathtt{e}} \leq e$ because:

- if $t \to_{\mathtt{m}_{\mathrm{need}}} u$ then $|d|_{\mathtt{m}} = |d'|_{\mathtt{m}} + 1 \leq m' + 1 = m$ and $|d|_{\mathtt{e}} = |d'|_{\mathtt{e}} \leq e' = e$,
- if $t \to_{\mathtt{e}_{\mathrm{need}}} u$ then $|d|_{\mathtt{m}} = |d'|_{\mathtt{m}} \leq m' = m$ and $|d|_{\mathtt{e}} = |d'|_{\mathtt{e}} + 1 \leq e' + 1 = e$. ☐

Corollary 4 (CbNeed duplicates as wisely as CbV). *Let* $d \colon t \to^*_{\mathrm{cbv}} u$ *with* $\mathsf{normal}_{\mathrm{cbv}}(u)$. *Then there is* $d' \colon t \to^*_{\mathrm{need}} s$ *with* $\mathsf{normal}(s)$ *and* $|d'|_{\mathtt{m}} \leq |d|_{\mathtt{m}}$ *and* $|d'|_{\mathtt{e}} \leq |d|_{\mathtt{e}}$.

Proof. By tight completeness for CbV (Theorem 4), there exists a tight type derivation $\Phi \triangleright_{\mathrm{cbv}} \; \vdash^{(m,e)} t : \mathbf{0}$ with $|d|_{\mathtt{m}} = m$ and $|d|_{\mathtt{e}} = e$, because by hypothesis t is CbV normalisable. Correctness of CbV with respect to CbNeed (Corollary 1) then gives $d' \colon t \to^*_{\mathrm{need}} s$ with $\mathsf{normal}(s)$, $|d'|_{\mathtt{m}} \leq m = |d|_{\mathtt{m}}$ and $|d'|_{\mathtt{e}} \leq e = |d|_{\mathtt{e}}$. ☐