Quantitative Weak Linearisation*

Sandra Alves[1][0000−0001−8840−5587] and Daniel Ventura[2][0000−0001−5141−0438]

[1] DCC-FCUP & CRACS, Univ. Porto, Portugal
[2] INF, Univ. Federal de Goiás, Brazil

Abstract. Weak linearisation was defined years ago through a static characterization of the intuitive notion of virtual redex, based on (legal) paths computed from the (syntactical) term tree. Weak-linear terms impose a linearity condition only on functions that are applied (consumed by reduction) and functions that are not applied (therefore persist in the term along any reduction) can be non-linear. This class of terms was shown to be strongly normalising with deciding typability in polynomial time. We revisit this notion through non-idempotent intersection types (also called quantitative types). By using an effective characterisation of minimal typings, based on the notion of tightness, we are able to distinguish between *"consumed"* and *"persistent"* term constructors, which allows us to define an expansion relation, between general λ-terms and weak-linear λ-terms, whilst preserving normal forms by reduction.

Keywords: Intersection Types, Linearisation, Minimal typings

1 Introduction

The notion of linearisation of the λ-calculus, as a process of transforming (or simulating) non-linear terms into (or using) *"equivalent"* linear terms, was first introduced in [27], and has since been used as a transformation technique in different settings [19, 20, 31]. From the computational point of view, linear terms have nice properties: there are no infinite reduction sequences starting from linear terms, there is no duplication of terms upon function evaluation, and every linear term is typable in the simple type system [22] thus it is typable in polynomial time. Regarding implementation issues of linear programs, both inline expansion and updating of structures in place, can be done safely. Several abstract machines for functional calculi have been designed to take into account and optimise the computation in the case of linear functions (see for instance [30, 33, 28, 5]). Linear functions are also naturally occurring in hardware compilation [19]. Circuits are static (i.e., they cannot be copied at run-time), so linear computations are more naturally compiled into hardware.

In this paper we revisit the notion of weak-linearisation, as defined in [6], through a notion of expansion based on non-idempotent intersection types and

* First author supported by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020. Second author partially supported by CNPq Universal 430667/2016-7 grant.

the notion of minimal typings [1]. Non-idempotent intersections were independently introduced by Gardner [18] and Kfoury [27], and turned out to be crucial in several resource aware consumption investigations [2, 3, 7, 9, 12, 24, 26]. The foundation for resource aware reasoning dates back to Girard's seminal work on resource management in proof theory that lead to the definition of linear logic [21]. The quantitative feature of the non-idempotent intersections was highlighted in De Carvalho's thesis [16] where its relation with linear logic [21] and quantitative relational models has been deeply explored. In the intersection types discipline [14], objects are allowed to be typed with distinct types combined by means of an intersection operator. If this intersection operator ($\cap$) does not enjoy idempotency (that is, it distinguishes $\tau \cap \tau$ from τ), then besides providing qualitative information on how objects are used, it also gives quantitative information on how many times they are used. For that reason, these type systems are called (nowadays) quantitative type systems.

Kfoury [27], used non-idempotent intersections to define a linear λ-calculus and the notion of linearisation. Kfoury's linear calculus, denoted $\Lambda^{\wedge}$, came with a new notion of reduction, called $\beta^{\wedge}$. Linearisation was defined indirectly by means of the notion of *contraction*, such that well-formed terms of the new calculus were those for which there was a contracted term in the λ-calculus. The relation between linearisation and intersection types was further explored by Florido and Damas in [17], introducing a notion of expansion of terms in the λ-calculus into linear terms, such that, any term typable using intersection types [14] is related with a term typable using simple types [15]. In other words, any typable term can be expanded to a simply typable linear term. For instance, the term $\delta\mathtt{I}$ –where $\delta \equiv \lambda x.xx$ and $\mathtt{I} \equiv \lambda x.x$– is typable in such an intersection type system and is related with term $(\lambda x_1 x_2.x_1 x_2)\,\mathtt{I}\,\mathtt{I}$, also said to be a linear expansion of the former. Expansion is preserved by weak-head reduction[3], in the sense that reduction commutes with expansion.

An operational notion of linearisation, called weak-linearisation, was defined by Alves and Florido [6], based on the notion of legal paths. Legal paths were introduced by Asperti and Laneve [8] as a characterisation based on paths of Lévy's redex families in the context of optimal reductions for the λ-calculus [29], providing a static characterisation of the intuitive notion of virtual redex. Weak linear λ-terms are strongly normalising, typable in polynomial time, and the transformation of general terms into weak linear terms was shown to preserve normal forms. A term t is weak linear if in any reduction sequence from t, when there is a contraction of a β-redex $(\lambda x.u)v$, then x occurs free in u at most once, therefore only functions that are applied to an argument in some reduction process are required to be linear. Although this class of terms is often called *affine*, whereas linear is used for terms $\lambda x.t$ where x occurs free exactly once in t, the definition of linear function used here follows [27, 17]. Notice that the definition of weak linear term refers to all the abstractions $\lambda x.u$ that are going to play the function part of a β-redex along the reduction of t, and legal

[3] a notion of reduction used in functional programming languages implementation, which does not evaluate inside values, *i.e.* under abstractions.

paths were used to identify these virtual redexes. For example $t \equiv (\mathtt{I}\,\delta)k$ is not weak linear, since $(\delta\,k)$ is a redex in a reduction sequence from t, whereas $u \equiv ((\lambda xy.yx)\,\delta)k$ is weak-linear, since the only duplicating abstraction δ is never applied during (any) reduction from u. Note also that we require the capability to distinguish functions that are going to be applied, therefore *"consumed"* by (any) reduction, from the ones that are going to *"persist"* in the term. For example in $\delta\,\mathtt{I}$, only one of the copies of $\mathtt{I}$ is going to play the function part of a β-redex, whereas the other is going to remain in the normal form. This dichotomy of consumed/persistent terms has been highlighted in [25] for the $\lambda\mu$-calculus, and adapted to a pattern-calculus in [7], using a quantitative typing system extended with the notion of tight types. Tight-typings, which provide an effective characterisation of minimal typings, have been applied to the λ-calculus [1] to extract exact bounds for different evaluation strategies, and in particular for a maximal reduction strategy.

In this paper we combine the notion of expansion relation, as defined by Florido and Damas in [17], with the notion of tight types in a non-idempotent intersection type system following [1], and define a new expansion relation that captures the operational notion of weak-linearisation. We prove that expansion commutes with β-reduction and that expanded terms are weak-linear in the sense of [6], and therefore typable in polynomial time. Furthermore, because we base our expansion relation on a quantitative type system that characterises strong normalisation, we are able to show that all strong-normalising λ-terms can be simulated by a weak-linear λ-term, with the same β-normal form. This is a major breakthrough with respect to the approach based on legal paths [6, 4], where such relation was left as a conjecture. Note that Kfoury also conjectured that strongly normalisable λ-terms are those that can be simulated by a corresponding well-formed term in $\Lambda^{\wedge}$ [27]. Additionally, using tight-types we are able to give a typing characterisation of expanded terms, by introducing a linear restriction of the typing system in [1], such that weak linear λ-terms are exactly the terms typed in this system.

Overview: The rest of the paper is structured as follows. In Section 2 we give some preliminary notions on the λ-calculus and discuss some properties of the non-deterministic maximal reduction strategy that we are using in this paper. In Section 3, we recall the non-idempotent type system for a maximal reduction strategy given in [1], together with some properties that we adapt to our non-deterministic strategy. Our expansion relation is developed in Section 4, where we prove its relation with our maximal reduction strategy. In Section 5, we give a typing characterisation of weak-linear terms, by means of a linear system with tight-typings. We discuss some related work in Section 6 and finally conclude and discuss future work in Section 7.

2 Terms and Reductions

We start by recalling some basic definitions on the λ-calculus. For a detailed reference see [10]. The **set of λ-terms**, denoted by Λ, is inductively defined by

means of the following grammar: $t, u ::= x \mid \lambda x.t \mid t\,u$; here $x, y, z, w \ldots$ range over an infinite countable set of variables. The term x is called a *variable*, $\lambda x.t$ is an *abstraction* and tu is an *application*. We write $\mathtt{abs}(t)$, to denote that t is a term of the form $\lambda x.u$. The **size** $|t|$ of a term is defined by: $|x| = 0$; $|\lambda x.u| = 1+|u|$ and $|uv| = |u| + |v| + 1$. **Free variables** of terms are defined as: $\mathtt{fv}(x) := \{x\}$, $\mathtt{fv}(\lambda x.t) := \mathtt{fv}(t) \setminus \{x\}$ and $\mathtt{fv}(tu) := \mathtt{fv}(t) \cup \mathtt{fv}(u)$.

Terms are considered modulo α-conversion, so that for example $\lambda xy.xz =_\alpha \lambda x'y'.x'z$, but $\lambda x.xz \neq_\alpha \lambda z.zz$. We use $t\{x/u\}$ to denote the **meta-level substitution** operation which replaces all the free occurrences of x in t by the term u. As usual, this operation is performed modulo α-conversion so that capture of free variables is avoided. Throughout the paper, whenever necessary, we will assume Barendregt's Variable Convention (BVC), regarding the use of free and bound variables (see [10]).

We now present an inductive definition of a non-deterministic maximal reduction strategy, denoted $\to_{\mathsf{nmx}}$, that is closely related to the deterministic maximal reduction strategy $\to_{\mathsf{mx}}$ in [1]. A maximal strategy computes the longest reduction sequence to a normal form, whenever such a reduction exists, obtaining an infinite reduction sequence otherwise. The $\to_{\mathsf{mx}}$ reduction strategy is based on a maximal strategy defined by [11] and further developed in [32]. This reduction was used in [32] to reason about strong normalisation, in [6] to prove strong normalization of weak-linear λ-terms and in [1] to reason about quantitative measures for reduction.

The $\to_{\mathsf{nmx}}$ **reduction strategy** on λ-terms is defined in Figure 1. Superscript e in $\xrightarrow{e}_{\mathsf{nmx}}$ denotes the size of a term erased in reduction and is mostly omitted from notation throughout the paper. The definition of $\to_{\mathsf{nmx}}$ relies on the notion of **head-neutral normal forms**. An inductive definition of head-neutral normal forms, denoted by $t \in \mathcal{N}_{\mathsf{hd}}$, and (nmx)**-normal forms**, denoted by $t \in \mathcal{M}$, is given by the following grammars:

$$\mathcal{N}_{\mathsf{hd}} ::= x \mid \mathcal{N}_{\mathsf{hd}}\,\Lambda \qquad\qquad \mathcal{N} ::= x \mid \mathcal{N}\,\mathcal{M} \qquad\qquad \mathcal{M} ::= \mathcal{N} \mid \lambda x.\mathcal{M}$$

A term $t \in \mathcal{N}$ is called a neutral normal form. Note that $\mathcal{N} \subseteq \mathcal{N}_{\mathsf{hd}}$.

The **reflexive-transitive closure** of $\to_{\mathsf{nmx}}$, is defined by:

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

The only difference between $\to_{\mathsf{mx}}$ and $\to_{\mathsf{nmx}}$ is that $\to_{\mathsf{mx}}$ last rule is defined with $t \in \mathcal{N}$, *i.e.* in a term of the form $xu_1 \ldots u_n$, while u_i will be reduced only when $xu_1 \ldots u_{i-1} \in \mathcal{N}$ in the former, *i.e.* $u_j \in \mathcal{M}$ for each $1 \leq j < i$, any u_i is reduced independently in the latter. For instance, let $t = x\,(\mathtt{I}\,\mathtt{I})\,(\mathtt{I}\,\mathtt{I})$, then t can only be $\to_{\mathsf{mx}}$-reduced to $x\,\mathtt{I}\,(\mathtt{I}\,\mathtt{I})$ while it can also be $\to_{\mathsf{nmx}}$-reduced to $x\,(\mathtt{I}\,\mathtt{I})\,\mathtt{I}$. Therefore, $\to_{\mathsf{mx}} \subseteq \to_{\mathsf{nmx}}$.

We establish some necessary properties about $\to_{\mathsf{nmx}}$ and normal forms.

Proposition 1. *Let t be a λ-term:*

$$\frac{x \in \mathtt{fv}(t)}{(\lambda x.t)u \xrightarrow{0}_{\mathsf{nmx}} t\{x/u\}} \qquad \frac{u \xrightarrow{r}_{\mathsf{nmx}} u' \quad x \notin \mathtt{fv}(t)}{(\lambda x.t)u \xrightarrow{r}_{\mathsf{nmx}} (\lambda x.t)u'} \qquad \frac{u \not\to_{\mathsf{nmx}} \quad x \notin \mathtt{fv}(t)}{(\lambda x.t)u \xrightarrow{|u|}_{\mathsf{nmx}} t}$$

$$\frac{t \xrightarrow{r}_{\mathsf{nmx}} t'}{\lambda x.t \xrightarrow{r}_{\mathsf{nmx}} \lambda x.t'} \qquad \frac{\neg\mathtt{abs}(t) \quad t \xrightarrow{r}_{\mathsf{nmx}} t'}{tu \xrightarrow{r}_{\mathsf{nmx}} t'u} \qquad \frac{t \in \mathcal{N}_{\mathsf{hd}} \quad u \xrightarrow{r}_{\mathsf{nmx}} u'}{tu \xrightarrow{r}_{\mathsf{nmx}} tu'}$$

Fig. 1. Non-deterministic maximal reduction strategy

1. *If* $t \in \mathcal{N}_{\mathsf{hd}}$ *then* $\neg\mathtt{abs}(t)$. *In particular,* $t \in \mathcal{N}$ *implies that* $\neg\mathtt{abs}(t)$.
2. $t \in \mathcal{M}$ *iff* $t \not\to_{\mathsf{nmx}}$.
3. *If* $t \in \mathcal{N}_{\mathsf{hd}}$ *and* $t \to_{\mathsf{nmx}} t'$ *then* $t' \in \mathcal{N}_{\mathsf{hd}}$.

Proof. 1. Follows directly from the definition of $\mathcal{N}_{\mathsf{hd}}$.
2. ($\Rightarrow$): By induction on $t \in \mathcal{M}$. ($\Leftarrow$): A a slightly stronger result is proved by induction on t : If $t \not\to_{\mathsf{nmx}}$ then $\neg\mathtt{abs}(t)$ implies $t \in \mathcal{N}$ otherwise $t \in \mathcal{M}$.
3. By induction on $t \in \mathcal{N}_{\mathsf{hd}}$.

We prove that $\to_{\mathsf{nmx}}$ has the diamond property (DP) thus is also maximal.

Lemma 1 (nmx-reduction has DP). *Let* t *be a term. If* $u \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; t \xrightarrow{e}_{\mathsf{nmx}} v$, *with* $u \neq v$, *then there is* w *s.t.* $u \xrightarrow{e}_{\mathsf{nmx}} w \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; v$.

Proof. By induction on t. We present the case for $t = t_1t_2$ where $\neg\mathtt{abs}(t_1)$:

- If $t_1 \in \mathcal{N}$, in which case (by Prop. 1.2) $t_1 \not\to_{\mathsf{nmx}}$, then $u = t_1u_2$ and $v = t_1v_2$ where $u_2 \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; t_2 \xrightarrow{e}_{\mathsf{nmx}} v_2$. By the *i.h.* there is w_2 s.t. $u_2 \xrightarrow{e}_{\mathsf{nmx}} w_2 \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; v_2$. Therefore $u \xrightarrow{e}_{\mathsf{nmx}} w \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; v$ for $w = t_1w_2$.
- If $t_1 \notin \mathcal{N}$, but $t_1 \in \mathcal{N}_{\mathsf{hd}}$ then there are two subcases:
  - If $u = u_1t_2$ where $t_1 \xrightarrow{e'}_{\mathsf{nmx}} u_1$, then $u_1 \in \mathcal{N}_{\mathsf{hd}}$ (Prop. 1.3) and there are two possibilities:
    - If $v = t_1v_2$ where $t_2 \xrightarrow{e}_{\mathsf{nmx}} v_2$ then, for $w = u_1v_2$ we have that $u \xrightarrow{e}_{\mathsf{nmx}} w \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; v$.
    - If $v = v_1t_2$ where $t_1 \xrightarrow{e}_{\mathsf{nmx}} v_1$ then by the *i.h.* there is w_1 s.t. $u_1 \xrightarrow{e}_{\mathsf{nmx}} w_1 \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; v_1$. Since $v_1 \in \mathcal{N}_{\mathsf{hd}}$ (Prop. 1.3), we have that $u \xrightarrow{e}_{\mathsf{nmx}} w \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; v$ for $w = w_1t_2$.
  - If $u = t_1u_2$ where $t_2 \xrightarrow{e'}_{\mathsf{nmx}} u_2$ then the cases are similar as above.
- If $t_1 \notin \mathcal{N}_{\mathsf{hd}}$ then $u = u_1t_2$ and $v = v_1t_2$ where $u_1 \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; t_1 \xrightarrow{e}_{\mathsf{nmx}} v_1$. By the *i.h.* there is w_1 s.t. $u_1 \xrightarrow{e}_{\mathsf{nmx}} w_1 \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; v_1$. If $\mathtt{abs}(u_1)$ then either $t_1 = (\lambda x.t_{11})t_{12}$ and $u_1 = t_{11}\{x/t_{12}\}$ or $t_1 = (\lambda x.t_{11})t_{12}$ and $u_1 = t_{11}$. In either case we would have that $u = v$. Then, the only diverging case is when $\neg\mathtt{abs}(u_1)$ and $\neg\mathtt{abs}(v_1)$, therefore $u \xrightarrow{e}_{\mathsf{nmx}} w \;{}_{\mathsf{nmx}}\!\xleftarrow{e'}\; v$ for $w = w_1t_2$.

Corollary 1. $\to_{\mathsf{nmx}}$ *is maximal.*

3 Types

In this section we recall the non-idempotent type system for a maximal reduction strategy given in [1], which we will refer to as $\mathcal{MX}$. This can be seen as an extension of Gardner's system [18], but in which tight types are used to identify persistent term constructors, that is constructors that are not going to be consumed by reduction. The **sets of types** ($\mathcal{T}$) and **multi-types** are given by the following grammars:

$$\begin{array}{lll} (\textbf{tight-types})\ \mathtt{t} ::= \bullet_{\mathcal{M}} \mid \bullet_{\mathcal{N}} & (\textbf{types}) & \sigma,\tau ::= \mathtt{t} \mid \mathcal{A}\to\sigma \\ & (\textbf{multi-types}) & \mathcal{A} ::= [\sigma_k]_{k\in K} \end{array}$$

where $\mathtt{t}$ denotes **tight-types**: the constants $\bullet_{\mathcal{M}}$ and $\bullet_{\mathcal{N}}$ denote the type of any term reducing to a term in $\mathcal{M}$ and $\mathcal{N}$, respectively. A multiset type (multi-types) is an unordered list of (not necessarily different) elements where K denotes a (possibly empty) finite set of indexes. We use $[\,]$ to denote the *empty* multiset. We write $|\mathcal{A}|$ to denote the number of elements of the multiset $\mathcal{A}$. For example $[\bullet_{\mathcal{N}}, [\bullet_{\mathcal{N}}] \to \bullet_{\mathcal{N}}, \bullet_{\mathcal{N}}]$ is a multi-type of 3 elements, representing the intersection type $\bullet_{\mathcal{N}} \cap ([\bullet_{\mathcal{N}}] \to \bullet_{\mathcal{N}}) \cap \bullet_{\mathcal{N}}$, where $\cap$ is an associative, commutative and non-idempotent intersection type constructor. We write $\sqcup$ to denote multiset union. As usual the arrow constructor is right-associative.

A **typing context** Γ is a map from variables to multi-types, such that only finitely many variables are not mapped to the empty multiset $[\,]$. We write $\mathtt{dom}(\Gamma)$ to denote the domain of Γ, which is the set $\{x \mid \Gamma(x) \neq [\,]\}$. We may write $\Gamma\#\Delta$ if and only if $\mathtt{dom}(\Gamma)$ and $\mathtt{dom}(\Delta)$ are disjoint. Given typing contexts $\{\Gamma_k\}_{k\in K}$ we write $+_{k\in K}\Gamma_k$ for the context that maps x to $\sqcup_{k\in K}\Gamma_k(x)$. One particular case is $\Gamma + \Delta$. We sometimes write $\Gamma;\Delta$ instead of $\Gamma + \Delta$, when $\Gamma\#\Delta$, and we do not distinguish $\Gamma; x : [\,]$ from Γ. **Context inclusion relation** is defined by: $\Delta \subseteq \Gamma$ iff $\exists \Gamma', \Delta + \Gamma' = \Gamma$. The typing context $\Gamma\!\setminus\! x$ is defined by $(\Gamma\!\setminus\! x)(y) = \Gamma(y)$ if $y \neq x$ and $[\,]$ otherwise.

We write $\mathtt{tight}(\sigma)$, if σ is of the form $\bullet_{\mathcal{M}}$ or $\bullet_{\mathcal{N}}$. We extend this notion to multi-types and typing contexts as expected, that is, $\mathtt{tight}([\sigma_i]_{i\in I})$ if $\mathtt{tight}(\sigma_i)$ for all $i \in I$, and $\mathtt{tight}(\Gamma)$ if $\mathtt{tight}(\Gamma(x))$, for all $x \in \mathtt{dom}(\Gamma)$.

The typing assignment system $\mathcal{MX}$ is given in Figure 2. Indexes in the pair (b, r) in a typing $\Gamma \vdash_{\mathsf{mx}}^{(b,r)} t : \tau$ are related with the number of $\to_{\mathsf{nmx}}$ steps and the size of corresponding normal form, respectively, (see Ex. 1) and will be mostly ommitted throughout the paper. We use $\Phi \rhd \Gamma \vdash_{\mathsf{mx}} t : \sigma$ (resp. $\Phi \rhd \Gamma \vdash_{\mathsf{mx}} t : \mathcal{A}$) to denote a **type derivation** ending with the sequent $\Gamma \vdash_{\mathsf{mx}} t : \sigma$ (resp. $\Gamma \vdash_{\mathsf{mx}} t : \mathcal{A}$). The size of a derivation Φ, denoted by $\mathtt{sz}(\Phi)$, is the number of all typing rules but $\mathtt{many}_{>0}$ and $\mathtt{none}$ used in Φ. Note that rule ($\mathtt{none}$) demands typability of the premise, not allowing arguments of erasing-functions to be untyped thus preventing divergent terms as arguments.

System $\mathcal{MX}$ is **relevant**: if $\Phi \rhd \Gamma \vdash_{\mathsf{mx}} t : \tau$ then $\mathtt{dom}(\Gamma) = \mathtt{fv}(t)$. In other words, no weakening is allowed in the typing system. Given a derivation $\Phi \rhd \Gamma \vdash_{\mathsf{mx}} t : \tau$, we say Φ is **garbage-tight** if $\mathtt{tight}(\sigma)$ in every application of rule ($\mathtt{none}$); and is **(maximal) tight**, denoted by $\mathtt{tight}(\Phi)$, if it is garbage-tight,

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

$$\frac{\Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \tau}{\Gamma \setminus\!\setminus x \vdash^{(b+1,r)}_{\mathsf{mx}} \lambda x.t : \Gamma(x) \to \tau}\ (\mathtt{fun}_b) \qquad \frac{\Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \mathtt{t} \qquad \mathtt{tight}(\Gamma(x))}{\Gamma \setminus\!\setminus x \vdash^{(b,r+1)}_{\mathsf{mx}} \lambda x.t : \bullet_{\mathcal{M}}}\ (\mathtt{fun}_r)$$

$$\frac{\Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \mathcal{A} \to \tau \qquad \Delta \vdash^{(b',r')}_{\mathsf{mx}} u : \mathcal{A}}{\Gamma + \Delta \vdash^{(b+b'+1,r+r')}_{\mathsf{mx}} t\,u : \tau}\ (\mathtt{app}_b) \qquad \frac{\Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \bullet_{\mathcal{N}} \qquad \Delta \vdash^{(b',r')}_{\mathsf{mx}} u : \mathtt{t}}{\Gamma + \Delta \vdash^{(b+b',r+r'+1)}_{\mathsf{mx}} t\,u : \bullet_{\mathcal{N}}}\ (\mathtt{app}^{lo}_r)$$

$$\frac{(\Delta_k \vdash^{(b_k,r_k)}_{\mathsf{mx}} t : \sigma_k)_{k\in K} \qquad |K| > 0}{+_{k\in K}\Delta_k \vdash^{(+_{k\in K} b_k, +_{k\in K} r_k)}_{\mathsf{mx}} t : [\sigma_k]_{k\in K}}\ (\mathtt{many}_{>0}) \qquad \frac{\Delta \vdash^{(b,r)}_{\mathsf{mx}} t : \sigma}{\Delta \vdash^{(b,r)}_{\mathsf{mx}} t : [\,]}\ (\mathtt{none})$$

Fig. 2. System $\mathcal{MX}$

$\mathtt{tight}(\Gamma)$ and $\mathtt{tight}(\tau)$. **Tightness** is the key notion in the typing system, in which persistent and consumable (term) constructors are discriminated. Such classification allows one to give precise measures on both number of steps and size of the corresponding normal form.

Example 1. Consider $t \equiv (\lambda x.x\mathtt{I}x)\Delta$, with $\mathtt{I} \equiv \lambda z.z$ and $\Delta \equiv \lambda y.yy$. Let $\mathcal{B} = [\underbrace{[[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}}_{\tau_1}, \underbrace{[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}}_{\tau_2}]$ and $\mathcal{A} = [\bullet_{\mathcal{M}}, \mathcal{B} \to [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]$. Let Φ be:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : [\bullet_{\mathcal{N}}] \vdash^{(0,0)} x : \bullet_{\mathcal{N}}} \quad \dfrac{}{x : [\bullet_{\mathcal{N}}] \vdash^{(0,0)} x : \bullet_{\mathcal{N}}}}{x : [\bullet_{\mathcal{N}}, \bullet_{\mathcal{N}}] \vdash^{(0,1)} xx : \bullet_{\mathcal{N}}}}{\vdash^{(0,2)} \Delta : \bullet_{\mathcal{M}}} \qquad \dfrac{\dfrac{\dfrac{}{x : [\tau_1] \vdash^{(0,0)} x : \tau_1} \quad \dfrac{\dfrac{}{x : [\tau_2] \vdash^{(0,0)} x : \tau_2}}{x : [\tau_2] \vdash^{(0,0)} x : [\tau_2]}}{x : \mathcal{B} \vdash^{(1,0)} xx : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}}}{\vdash^{(2,0)} \Delta : \mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}}}{\vdash^{(2,2)} \Delta : \mathcal{A}}$$

and $\Phi_{\mathtt{I}}$ be:

$$\dfrac{\dfrac{x : [\tau_2] \vdash^{(0,0)} x : \tau_2}{\vdash^{(1,0)} \mathtt{I} : \tau_1} \quad \dfrac{y : [\bullet_{\mathcal{M}}] \vdash^{(0,0)} y : \bullet_{\mathcal{M}}}{\vdash^{(1,0)} \mathtt{I} : \tau_2}}{\vdash^{(2,0)} \mathtt{I} : \mathcal{B}}$$

We have the following tight derivation for t:

$$\dfrac{\dfrac{\dfrac{x : [\mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}] \vdash^{(0,0)} x : \mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}} \quad (\Phi_{\mathtt{I}})_{(2,0)}}{x : [\mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}] \vdash^{(3,0)} x\mathtt{I} : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}} \quad \dfrac{x : [\bullet_{\mathcal{M}}] \vdash^{(0,0)} x : \bullet_{\mathcal{M}}}{x : [\bullet_{\mathcal{M}}] \vdash^{(0,0)} x : [\bullet_{\mathcal{M}}]}}{x : \mathcal{A} \vdash^{(4,0)} x\mathtt{I}x : \bullet_{\mathcal{M}}}}{\vdash^{(5,0)} (\lambda x.x\mathtt{I}x) : \mathcal{A}\to\bullet_{\mathcal{M}}} \quad \Phi_{(2,2)}}{\vdash^{(8,2)} (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}}$$

Moreover, $t \to_{\mathsf{nmx}} \Delta\mathtt{I}\Delta \to_{\mathsf{nmx}} \mathtt{I}\mathtt{I}\Delta \to_{\mathsf{nmx}} \mathtt{I}\Delta \to_{\mathsf{nmx}} \Delta$. In other words, $t \to_{\mathsf{nmx}}$-normalization has $4 = 8/2$ steps with normal form $|\Delta| = 2$. Notice that the Δ

copy typed with $\bullet_{\mathcal{M}}$ in Φ characterizes its abstraction constructor as persistent, not consumed during normalisation, *i.e.* its λ is not applied and, in this case, occurs in t's normal form.

We now recall some properties from [1] that are relevant to our work. Properties regarding normal forms[4], such as typability with a tight derivation, and (Anti) Substitution Lemmas do not depend on the reduction strategy and thus trivially holds in the current presentation. Below we present one of such properties, which play a role when considering nmx-reductions.

Lemma 2 (Tight spreading on neutral terms for $\mathcal{MX}$[1]). *If $t \in \mathcal{N}_{\mathsf{hd}}$ and $\Phi \rhd \Gamma \vdash_{\mathsf{mx}} t : \tau$ s.t. $\mathtt{tight}(\Gamma)$ then $\mathtt{tight}(\tau)$.*

Proposition 2 (Quantitative Subject Reduction). *If $\Phi \rhd \Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \tau$ is tight and $t \xrightarrow{e}_{\mathsf{nmx}} t'$ then there is a $\Gamma' \subseteq \Gamma$ and a tight derivation Φ' s.t. $\Gamma' \vdash^{(b-2,r-e)}_{\mathsf{mx}} t' : \tau$.*

Proof. As in [1], the proof is by induction on $t \xrightarrow{e}_{\mathsf{nmx}} t'$ of a stronger property: Let $t \xrightarrow{e}_{\mathsf{nmx}} t'$, $\Phi \rhd \Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \tau$ garbage-tight, $\mathtt{tight}(\Gamma)$ and either $\mathtt{tight}(\tau)$ or $\neg\mathtt{abs}(t)$. Then there are a $\mathtt{tight}(\Gamma')$ s.t. $\Gamma' \subseteq \Gamma$ and a garbage-tight typing $\Phi' \rhd \Gamma' \vdash^{(b-2,r-e)}_{\mathsf{mx}} t' : \tau$. It is sufficient to analyse the only case in which reduction $\rightarrow_{\mathsf{nmx}}$ differs from the deterministic strategy in [1]. Lemma 2 is used in this case, which holds for terms in $\mathcal{N}_{\mathsf{hd}}$. Therefore, the proof steps are the same as in [1].

We can then establish the corresponding subject expansion property with a similar consideration regarding Lemma 2.

Proposition 3 (Quantitative Subject Expansion). *If $\Phi \rhd \Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \tau$ is tight and $t' \xrightarrow{e}_{\mathsf{nmx}} t$ then there is a $\Gamma \subseteq \Gamma'$ and a tight derivation Φ' s.t. $\Gamma' \vdash^{(b+2,r+e)}_{\mathsf{mx}} t' : \tau$.*

Given that (quantitative) subject reduction and expansion properties still hold for nmx-reduction, tight correctness and completeness is proved as in [1].

Theorem 1 (Tight Correctness for nmx). *Let $\Phi \rhd \Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \tau$ be a tight derivation. Then there is an integer e and a term $u \in \mathcal{M}$ s.t. $t \xrightarrow{e}{}^{b/2}_{\mathsf{nmx}} u$ and $|u| + e = r$. Moreover, if $\tau = \bullet_{\mathcal{N}}$ then $u \in \mathcal{N}$.*

Theorem 2 (Tight Completeness for nmx). *If $t \xrightarrow{e}{}^{k}_{\mathsf{nmx}} u$ with $u \in \mathcal{M}$, then there exists a tight typing $\Phi \rhd \Gamma \vdash^{(2k,|u|+e)}_{\mathsf{mx}} t : \tau$. Moreover, if $u \in \mathcal{N}$ then $\tau = \bullet_{\mathcal{N}}$, and if $\mathtt{abs}(u)$ then $\tau = \bullet_{\mathcal{M}}$.*

Remark that, by the correctness and completeness results, an abstraction typed with $\bullet_{\mathcal{M}}$ in a tight derivation is in fact persistent. In other words, it occurs either in the normal form –contributing with $|u|$– or in some normal form erased along the reduction –contributing with e in the corresponding step.

[4] both deterministic and non-deterministic strategies have the same normal forms

$$
\begin{aligned}
\mathscr{E}(x:\sigma) &\triangleleft (y, \{x:[y:\sigma]\}),\ y \text{ fresh}\\
\mathscr{E}(x:\mathtt{t}) &\triangleleft (x, \{x:[x:\mathtt{t}]\})\\
\mathscr{E}(\lambda x.t:[\tau_i]_{i=1\dots n}\to\sigma) &\triangleleft (\lambda x_1\dots x_n.t^*, A), \text{if for some } n>0 \text{ and fresh } x_1,\dots,x_n\\
&\qquad \mathscr{E}(t:\sigma)\triangleleft(t^*, A; \{x:[x_1:\tau_1,\dots,x_n:\tau_n]\})\\
\mathscr{E}(\lambda x.t:[\,]\to\sigma) &\triangleleft (\lambda x.t^*, A),\ \text{if } x\notin \mathtt{fv}(t) \text{ and}\\
&\qquad \mathscr{E}(t:\sigma)\triangleleft(t^*, A)\\
\mathscr{E}(\lambda x.t:\bullet_{\mathcal{M}}) &\triangleleft (\lambda x.t^*, A), \text{if for some tight type } \mathtt{t} \text{ and } n\geq 0\\
&\qquad \mathscr{E}(t:\mathtt{t})\triangleleft(t^*, A; \{x:[x:\mathtt{t}_1,\dots,x:\mathtt{t}_n]\})\\
\mathscr{E}(tu:\sigma) &\triangleleft (t_0u_1\dots u_m, +_{j=0\dots m}A_j),\ \text{if for some } m>0 \text{ and } \tau_1,\dots,\tau_m\\
&\qquad \mathscr{E}(t:[\tau_j]_{j=1\dots m}\to\sigma)\triangleleft(t_0,A_0) \text{ and } (\mathscr{E}(u:\tau_j)\triangleleft(u_j,A_j))_{j=1\dots m}\\
\mathscr{E}(tu:\sigma) &\triangleleft (t^*u^*, A_1+A_2),\ \text{if for some type } \tau\\
&\qquad \mathscr{E}(t:[\,]\to\sigma)\triangleleft(t^*,A_1) \text{ and } \mathscr{E}(u:\tau)\triangleleft(u^*,A_2)\\
\mathscr{E}(tu:\bullet_{\mathcal{N}}) &\triangleleft (t^*u^*, A_1+A_2),\ \textit{if for some tight type } \mathtt{t}\\
&\qquad \mathscr{E}(t:\bullet_{\mathcal{N}})\triangleleft(t^*,A_1) \text{ and } \mathscr{E}(u:\mathtt{t})\triangleleft(u^*,A_2)
\end{aligned}
$$

Fig. 3. Expansion $\mathcal{E}$

4 Weak-linearisation for strongly normalising terms

In this section we present our expansion relation, following the relation by Florido and Damas [17]. The expansion relation $\mathscr{E}$, associates to a λ-term t typed with a non-idempotent type σ in $\mathcal{MX}$, an expanded (weak-linear) λ-term u and an expansion context A.

An **expansion context** A is a map from variables to multisets of the form $[y_1 : \sigma_1, \dots, y_m : \sigma_m]$, where y_i is variable and σ_i is a type, and such that only finitely many variables are not mapped to the empty multiset $[\,]$. As for typing contexts, we define $\mathtt{dom}(A)$ to denote the domain of A, which is the set $\{x \mid A(x) \neq [\,]\}$. The notions of $+_{k\in K}A_k$, and $A_1; A_2$, are defined as in the case of typing contexts but taking multisets of pairs $x:\sigma$, instead of multisets of types. The **expansion** relation $\mathscr{E}(t:\sigma)\triangleleft(u,A)$ is inductively defined in Figure 3.

Example 2. Considering $t \equiv (\lambda x.x\mathtt{I}x)\Delta$ from Example 1, with (tight) typing $\Psi \rhd \vdash^{(8,2)}_{\mathsf{mx}} (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}$. See Figure 4 for the complete expansion of t, where $\mathcal{A} = [\bullet_{\mathcal{M}}, \mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]$ and $\mathcal{B} = [[[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}, [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]$.

We now prove some basic properties on $\mathscr{E}$.

Lemma 3 (Basic properties of $\mathscr{E}$). *If $\mathscr{E}(t:\sigma)\triangleleft(u,A)$, then:*

1. *If $\neg\mathtt{abs}(t)$ then $\neg\mathtt{abs}(u)$.*
2. *If $t\in\mathcal{N}_{\mathsf{hd}}$ then $u\in\mathcal{N}_{\mathsf{hd}}$.*
3. *If $t\not\to_{\mathsf{nmx}}$ then $u\not\to_{\mathsf{nmx}}$.*

$\mathscr{E}((\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}) \lhd ((\lambda x_3x_4.x_3\mathtt{II}x_4)(\lambda x_1x_2.x_1x_2)(\lambda x.xx), \emptyset)$
  $\mathscr{E}(\lambda x.x\mathtt{I}x : \mathcal{A}\to\bullet_{\mathcal{M}}) \lhd (\lambda x_3x_4.x_3\mathtt{II}x_4, \emptyset)$
    $\mathscr{E}(x\mathtt{I}x : \bullet_{\mathcal{M}}) \lhd (x_3\mathtt{II}x_4, \{x : [x_3 : \mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}, x_4 : \bullet_{\mathcal{M}}]\})$
      $\mathscr{E}(x\mathtt{I} : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (x_3\mathtt{II}, \{x : [x_3 : \mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\})$
        $\mathscr{E}(x : \mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (x_3, \{x : [x_3 : \mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\})$
        $\mathscr{E}(\mathtt{I} : [[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (\lambda x_5.x_5, \emptyset)$
          $\mathscr{E}(x : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (x_5, \{x : [x_5 : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\})$
        $\mathscr{E}(\mathtt{I} : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (\lambda x_6.x_6, \emptyset)$
          $\mathscr{E}(x : \bullet_{\mathcal{M}}) \lhd (x_6, \{x : [x_6 : \bullet_{\mathcal{M}}]\})$
      $\mathscr{E}(x : \bullet_{\mathcal{M}}) \lhd (x_4, \{x : [x_4 : \bullet_{\mathcal{M}}]\})$
  $\mathscr{E}(\lambda x.xx : \mathcal{B}\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (\lambda x_1x_2.x_1x_2, \emptyset)$
    $\mathscr{E}(xx : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (x_1x_2, \{x : [x_1 : [[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}, x_2 : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\})$
      $\mathscr{E}(x : [[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (x_1, \{x : [x_1 : [[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\to[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\})$
      $\mathscr{E}(x : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}) \lhd (x_2, \{x : [x_2 : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}]\})$
  $\mathscr{E}(\lambda x.xx : \bullet_{\mathcal{M}}) \lhd (\lambda x.xx, \emptyset)$
    $\mathscr{E}(xx : \bullet_{\mathcal{N}}) \lhd (xx, \{x : [x : \bullet_{\mathcal{N}}, x : \bullet_{\mathcal{N}}]\})$
      $\mathscr{E}(x : \bullet_{\mathcal{N}}) \lhd (x, \{x : [x : \bullet_{\mathcal{N}}]\})$
      $\mathscr{E}(x : \bullet_{\mathcal{N}}) \lhd (x, \{x : [x : \bullet_{\mathcal{N}}]\})$

Fig. 4. An expansion for $(\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}$

Proof. 1. Follows by contraposition from the fact that the result of $\mathscr{E}$ is an abstraction only when the initial term was itself an abstraction.
2. By induction on $t \in \mathcal{N}_{\mathsf{hd}}$.
3. It suffices to prove that if $\mathscr{E}(t : \sigma) \lhd (u, A)$, then $t \in \mathcal{M}$ implies $u \in \mathcal{M}$. A slightly stronger result is then proved by induction on t: (1) If $t \in \mathcal{N}$ then $u \in \mathcal{N}$; (2) If $t \in \mathcal{M}$ then $u \in \mathcal{M}$.

Let A be an expansion context. The **typing context associated to** A, denoted Γ_A, is defined by:

$$\Gamma_{\{x:[x:\mathtt{t}_1,\ldots,x:\mathtt{t}_n]\}} = \{x : [\mathtt{t}_1, \ldots, \mathtt{t}_n]\} \qquad \Gamma_{A+B} = \Gamma_A + \Gamma_B$$
$$\Gamma_{\{x:[x_1:\tau_1,\ldots,x_n:\tau_n]\}} = \{x : [\tau_1, \ldots, \tau_n]\}$$

We say that A is tight, denoted by $\mathtt{tight}(A)$, if $\mathtt{tight}(\Gamma_A)$.

Lemma 4. *If $\mathscr{E}(t : \sigma) \lhd (t_1, A)$, then $\Phi \rhd \Gamma_A \vdash_{\mathsf{mx}} t : \sigma$. If Φ is (garbage-)tight then we say that the expansion is (garbage-)tight.*

Corollary 2. *If $\mathscr{E}(t : \sigma) \lhd (t_1, A)$, then $x \in \mathtt{dom}(A)$ iff $x \in \mathtt{fv}(t)$.*

Lemma 5. *If $\Phi \rhd \Gamma \vdash_{\mathsf{mx}} t : \sigma$ then there are u and A, s.t. $\mathscr{E}(t : \sigma) \lhd (u, A)$, where $\Gamma_A = \Gamma$.*

Proof. By induction on Φ. We present the case for rule ($\mathtt{fun}_b$): $\Gamma \setminus\!\setminus x \vdash_{\mathsf{mx}} \lambda x.t : \Gamma(x)\to\tau$ follows from $\Gamma \vdash_{\mathsf{mx}} t : \tau$. We have two cases:

- If $\Gamma(x) = [\tau_i]_{i=1\ldots n}$, thus $\Gamma = \Gamma'; x : [\tau_i]_{i=1\ldots n}$ then by the *i.h.*, there exists u and A, such that $\mathscr{E}(t : \tau) \triangleleft (u, A)$ with $\Gamma_A = \Gamma$. Then $A = A'; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\}$, with some $x_1, \ldots, x_n$ fresh variables and $\Gamma_{A'} = \Gamma'$. From which we get $\mathscr{E}(\lambda x.t : [\tau_i]_{i=1\ldots n} \to \tau) \triangleleft (\lambda x_1 \ldots x_n.u, A')$, and $\Gamma_{A'} = \Gamma' = \Gamma \setminus\!\setminus x$ as required.
- If $\Gamma(x) = [\,]$, then by the *i.h.*, there exists u and A, such that $\mathscr{E}(t : \tau) \triangleleft (u, A)$ with $\Gamma_A = \Gamma$. Note that, by relevance of $\mathcal{MX}$, $x \notin \mathtt{fv}(t)$, therefore, $\mathscr{E}(\lambda x.t : [\,] \to \tau) \triangleleft (\lambda x.u, A)$, and $\Gamma_A = \Gamma$ as expected.

Lemma 6. *If $\mathscr{E}(t : \sigma) \triangleleft (t_0, A_0; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\})$, then x occurs n times in t.*

Lemma 7. *If $\mathscr{E}(t : \sigma) \triangleleft (t_0, A_0; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\})$ and $\mathscr{E}(u : \tau_i) \triangleleft (u_i, A_i)$ for $i = 1 \ldots n$ with $n > 0$, then $\mathscr{E}(t\{u/x\} : \sigma) \triangleleft (t_0\{u_1/x_1, \ldots, u_n/x_n\}, A_0 + A_1 + \cdots + A_n)$. Moreover, if premise expansions are garbage-tight then so is the resulting expansion.*

Proof. By induction on t. We present the case for $t = \lambda y.t'$ with its three subcases:

- $\mathscr{E}(\lambda y.t' : [\sigma_i]_{i=1\ldots m} \to \sigma) \triangleleft (\lambda y_1 \ldots y_m.t^*, A; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\})$, follows from $\mathscr{E}(t' : \sigma) \triangleleft (t^*, A; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\}; \{y : [y_1 : \sigma_1, \ldots, y_m : \sigma_m]\})$, with $y \in \mathtt{fv}(t')$ (which is guaranteed to be garbage-tight whenever the main expansion is, by Lem. 4). By the *i.h.* $\mathscr{E}(t'\{u/x\} : \sigma) \triangleleft (t^*\{u_1/x_1, \ldots, u_n/x_n\}, A + A_1 + \cdots + A_n; \{y : [y_1 : \sigma_1, \ldots, y_m : \sigma_m]\})$[5], from which follows $\mathscr{E}(\lambda y.t'\{u/x\} : [\sigma_i]_{i=1\ldots m} \to \sigma) \triangleleft (\lambda y_1 \ldots y_m.t^*\{u_1/x_1, \ldots, u_n/x_n\}, A + A_1 + \cdots + A_n)$.
- $\mathscr{E}(\lambda y.t' : [\,] \to \sigma) \triangleleft (\lambda y.t^*, A; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\})$, follows from $\mathscr{E}(t' : \sigma) \triangleleft (t^*, A; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\})$, with $y \notin \mathtt{fv}(t')$. By the *i.h.* $\mathscr{E}(t'\{u/x\} : \sigma) \triangleleft (t^*\{u_1/x_1, \ldots, u_n/x_n\}, A + A_1 + \cdots + A_n)$, from which follows $\mathscr{E}(\lambda y.t'\{u/x\} : [\,] \to \sigma) \triangleleft (\lambda y.t^*\{u_1/x_1, \ldots, u_n/x_n\}, A + A_1 + \cdots + A_n)$[6].
- $\mathscr{E}(\lambda y.t' : \bullet_{\mathcal{M}}) \triangleleft (\lambda y.t^*, A; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\})$, follows from $\mathscr{E}(t' : \mathtt{t}) \triangleleft (t^*, A; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\}; \{y : [y : \mathtt{t}_1, \ldots, y : \mathtt{t}_m]\})$. Then $\mathscr{E}(t'\{u/x\} : \mathtt{t}) \triangleleft (t^*\{u_1/x_1, \ldots, u_n/x_n\}, A + A_1 + \cdots + A_n; \{y : [y : \mathtt{t}_1, \ldots, y : \mathtt{t}_m]\})$ by the *i.h.*, from which follows $\mathscr{E}(\lambda y.t'\{u/x\} : \bullet_{\mathcal{M}}) \triangleleft (\lambda y.t^*\{u_1/x_1, \ldots, u_n/x_n\}, A + A_1 + \cdots + A_n)$.

Theorem 3. *Let $\mathscr{E}(t_1 : \mathtt{t}) \triangleleft (u_1, A_1)$ be a tight expansion and $t_1 \to_{\mathsf{nmx}} t_2$:*

1. *There is a term u_2 such that $\mathscr{E}(t_2 : \mathtt{t}) \triangleleft (u_2, A_2)$ is tight, $u_1 \to^*_{\mathsf{nmx}} u_2$ and $A_2 \subseteq A_1$.*
2. *If $\neg\mathtt{abs}(u_1)$ then for any $u' \neq u_2$ s.t. $u_1 \to^*_{\mathsf{nmx}} u_2 = u_1 \to^*_{\mathsf{nmx}} u' \to^*_{\mathsf{nmx}} u_2$, $\neg\mathtt{abs}(u')$.*

[5] Note that $(A; \{y : [y_1 : \sigma_1, \ldots, y_m : \sigma_m]\}) + A_1 + \cdots + A_n = A + A_1 + \cdots + A_n; \{y : [y_1 : \sigma_1, \ldots, y_m : \sigma_m]\}$ since by BVC $y \notin \mathtt{fv}(u)$ thus $y \notin \mathtt{dom}(A_i)$ for each $1 \leq i \leq n$.

[6] $y \notin \mathtt{fv}(t')$ and by BVC $y \notin \mathtt{fv}(u)$ thus $y \notin \mathtt{fv}(t'\{u/x\})$

Proof. By induction on $t_1 \to_{\mathsf{nmx}} t_2$ we prove a stronger first statement: if $\mathscr{E}(t_1 : \sigma) \triangleleft (u_1, A_1)$ is garbage-tight, with $\mathtt{tight}(A_1)$ and either $\mathtt{tight}(\sigma)$ or $\neg\mathtt{abs}(t_1)$, and $t_1 \to_{\mathsf{nmx}} t_2$ then there is u_2 such that $\mathscr{E}(t_2 : \sigma) \triangleleft (u_2, A_2)$ is garbage-tight with $\mathtt{tight}(A_2)$, $A_2 \subseteq A_1$ and $u_1 \to^*_{\mathsf{nmx}} u_2$. We present the case $tu \to_{\mathsf{nmx}} tu'$, if $t \in \mathcal{N}_{\mathsf{hd}}$ and $u \to_{\mathsf{nmx}} u'$, with its three subcases:

- $\mathscr{E}(tu : \sigma) \triangleleft (t_0 u_1 \dots u_m, +_{j=0\dots m} A_j)$, if for some $m > 0$ and $\tau_1, \dots, \tau_m$, there are garbage-tight expansions $\mathscr{E}(t : [\tau_j]_{j=1\dots m} \to \sigma) \triangleleft (t_0, A_0)$ and $(\mathscr{E}(u : \tau_j) \triangleleft (u_j, A_j))_{j=1\dots m}$ with $\mathtt{tight}(A_j), j = 0 \dots m$. By the *i.h.* for each $j = 1 \dots m$, $\mathscr{E}(u' : \tau_j) \triangleleft (u'_j, A'_j)$, a garbage-tight expansion with $\mathtt{tight}(A'_j)$, $A'_j \subseteq A_j$ and $u_j \to^*_{\mathsf{nmx}} u'_j$. Therefore, $\mathscr{E}(tu' : \sigma) \triangleleft (t_0 u'_1 \dots u'_m, A_0 +_{j=1\dots m} A'_j)$ is a garbage-tight expansion with $\mathtt{tight}(A_0 +_{j=1\dots m} A'_j)$ and $A_0 +_{j=1\dots m} A'_j \subseteq +_{j=0\dots m} A_j$. By Lemma 3.2 we have $t_0 \in \mathcal{N}_{\mathsf{hd}}$ thus $t_0 u'_1 \dots u'_i \in \mathcal{N}_{\mathsf{hd}}$ for any $1 \leq i \leq m$ and $t_0 u_1 \dots u_m \to^*_{\mathsf{nmx}} t_0 u'_1 \dots u_m \to_{\mathsf{nmx}} \dots \to^*_{\mathsf{nmx}} t_0 u'_1 \dots u'_m$.
- $\mathscr{E}(tu : \sigma) \triangleleft (t_0 u_1, A_0 + A_1)$, if for some tight τ_1, there are garbage-tight expansions $\mathscr{E}(t : [\,] \to \sigma) \triangleleft (t_0, A_0)$ and $\mathscr{E}(u : \tau_1) \triangleleft (u_1, A_1)$ with $\mathtt{tight}(A_i), i = 0, 1$. By the *i.h.*, $\mathscr{E}(u' : \tau_1) \triangleleft (u'_1, A'_1)$ is garbage-tight with $\mathtt{tight}(A'_1)$ and $A'_1 \subseteq A_1$ and $u_1 \to^*_{\mathsf{nmx}} u'_1$. Therefore, $\mathscr{E}(tu' : \sigma) \triangleleft (t_0 u'_1, A'_0 + A'_1)$ is a garbage-tight expansion with $\mathtt{tight}(A_0 + A'_1)$ and $A_0 + A'_1 \subseteq A_0 + A_1$. By Lemma 3.2 we have $t_0 \in \mathcal{N}_{\mathsf{hd}}$ thus $t_0 u_1 \to^*_{\mathsf{nmx}} t_0 u'_1$.
- $\mathscr{E}(tu : \bullet_{\mathcal{N}}) \triangleleft (t^* u^*, A_1 + A_2)$, if for some tight type $\mathtt{t}$ there are garbage-tight expansions $\mathscr{E}(t : \bullet_{\mathcal{N}}) \triangleleft (t^*, A_1)$ and $\mathscr{E}(u : \mathtt{t}) \triangleleft (u^*, A_2)$ with $\mathtt{tight}(A_1 + A_2)$. By the *i.h.*, $\mathscr{E}(u' : \mathtt{t}) \triangleleft (u'', A'_2)$ is a garbage-tight expansion with $\mathtt{tight}(A'_2)$, $A'_2 \subseteq A_2$ and $u^* \to^*_{\mathsf{nmx}} u''$. Therefore, $\mathscr{E}(tu' : \bullet_{\mathcal{N}}) \triangleleft (t^* u'', A_1 + A'_2)$ is a garbage-tight expansion with $\mathtt{tight}(A_1 + A'_2)$ and $A_1 + A'_2 \subseteq A_1 + A_2$. By Lemma 3.2 we have $t^* \in \mathcal{N}_{\mathsf{hd}}$ thus $t^* u^* \to^*_{\mathsf{nmx}} t^* u''$.

5 A Characterisation of Weak-Linear Terms

In this section we introduce System $\mathcal{WL}$, proving that a term is tight-expanded if and only if its expansion is tight-typable in $\mathcal{WL}$. Let $\mathcal{WL}$ be the typing system with linear types, i.e. with types as in $\mathcal{T}$where multi-types are restricted to either empty or unitary multi-sets, i.e. with $|K| \leq 1$. Recall that our notion of linear is what is often referred to as affine, but given the relation of this work with [27] and [17], we will continue to use their notion of linear function.

The typing assignment system $\mathcal{WL}$ is given by the rules in Figure 5. System $\mathcal{WL}$ is a restriction of System $\mathcal{MX}$, i.e. if $\Gamma \vdash^{(b,r)}_{\mathsf{wl}} t : \tau$ then $\Gamma \vdash^{(b,r)}_{\mathsf{mx}} t : \tau$ where τ is linear. (Garbage-) tightness of derivations in $\mathcal{WL}$ thus have the same properties, allowing a characterisation of **weak-linear terms** as defined in [6]: A term t is weak linear if in any reduction sequence of t, when there is a contraction of a β-redex $(\lambda x.u)v$, then x occurs free in u at most once.

Theorem 4 (Characterisation of Weak-Linear Terms). *A term t is weak-linear iff t is tight-typable in system $\mathcal{WL}$.*

$$\frac{}{x : [\tau] \vdash_{\mathsf{wl}}^{(0,0)} x : \tau}\ (\mathtt{ax}) \qquad \frac{\Delta \vdash_{\mathsf{wl}}^{(b,r)} t : \sigma}{\Delta \vdash_{\mathsf{wl}}^{(b,r)} t : [\sigma]}\ (\mathtt{one}) \qquad \frac{\Delta \vdash_{\mathsf{wl}}^{(b,r)} t : \sigma}{\Delta \vdash_{\mathsf{wl}}^{(b,r)} t : [\,]}\ (\mathtt{none})$$

$$\frac{\Gamma \vdash_{\mathsf{wl}}^{(b,r)} t : \tau \qquad |\Gamma(x)| \leq 1}{\Gamma \setminus\!\setminus x \vdash_{\mathsf{wl}}^{(b+1,r)} \lambda x.t : \Gamma(x) \to \tau}\ (\mathtt{fun}_a) \qquad \frac{\Gamma \vdash_{\mathsf{wl}}^{(b,r)} t : \mathtt{t} \qquad \mathtt{tight}(\Gamma(x))}{\Gamma \setminus\!\setminus x \vdash_{\mathsf{wl}}^{(b,r+1)} \lambda x.t : \bullet_{\mathcal{M}}}\ (\mathtt{fun}_r)$$

$$\frac{\Gamma \vdash_{\mathsf{wl}}^{(b,r)} t : \mathcal{A} \to \tau \qquad \Delta \vdash_{\mathsf{wl}}^{(b',r')} u : \mathcal{A}}{\Gamma + \Delta \vdash_{\mathsf{wl}}^{(b+b'+1,r+r')} t\,u : \tau}\ (\mathtt{app}_b) \qquad \frac{\Gamma \vdash_{\mathsf{wl}}^{(b,r)} t : \bullet_{\mathcal{N}} \qquad \Delta \vdash_{\mathsf{wl}}^{(b',r')} u : \mathtt{t}}{\Gamma + \Delta \vdash_{\mathsf{wl}}^{(b+b',r+r'+1)} t\,u : \bullet_{\mathcal{N}}}\ (\mathtt{app}_r^{lo})$$

Fig. 5. System $\mathcal{WL}$

Proof. Let $\Phi \rhd \Gamma \vdash_{\mathsf{wl}}^{(b,r)} t : \tau$. By induction on Φ we prove that any abstraction $\lambda x.u$ in t where u has multiple free occurrences of x must be typed with rule $(\mathtt{fun}_r)$. Since Φ is also a tight derivation in $\mathcal{MX}$, by tight-correctness (Thm. 1) such abstraction is either in the corresponding $\to_{\mathsf{nmx}}$-normal form or is in a normal form erased along normalisation, i.e. is not applied in any step along the $\to_{\mathsf{nmx}}$-normalisation. On the other hand, tight typability of normal forms (see [1]) is still true for system $\mathcal{WL}$. Therefore, by $\mathcal{WL}$-typability of terms in $\mathcal{M}$, tight-completeness (Thm. 2) and observing that for any reduction step of a weak-linear term the typing restriction in system $\mathcal{WL}$ is sufficient for a (quantitative) subject expansion in the system, any weak-linear t is typable in system $\mathcal{WL}$.

In order to establish a relation between typings in $\mathcal{MX}$ and in $\mathcal{WL}$, we introduce a type translation and a mapping from expansion contexts, similarly to [17].

Type translation $\mathtt{w}(-)$ is defined by:

$$\mathtt{w}(\mathtt{t}) = \mathtt{t} \qquad \mathtt{w}([\tau_j]_{j=1\ldots m}) = [\mathtt{w}(\tau_j)]_{j=1\ldots m}$$
$$\mathtt{w}([\,] \to \tau) = [\,] \to \mathtt{w}(\tau) \qquad \mathtt{w}([\sigma_i]_{i=1\ldots m} \to \tau) = [\mathtt{w}(\sigma_1)] \to \cdots \to [\mathtt{w}(\sigma_m)] \to \mathtt{w}(\tau)$$

Extension of $\mathtt{w}(-)$ to typing contexts is straightforward. **Expanded Context** Γ_A^e is defined for any expansion context A by:

$$\Gamma^e_{\{x:[x:\mathtt{t}_1,\ldots,x:\mathtt{t}_n]\}} = \{x : [\mathtt{t}_1, \ldots, \mathtt{t}_n]\} \qquad \Gamma^e_{A+B} = \Gamma^e_A + \Gamma^e_B$$
$$\Gamma^e_{\{x:[x_1:\tau_1,\ldots,x_n:\tau_n]\}} = \{x_1 : [\mathtt{w}(\tau_1)], \ldots, x_n : [\mathtt{w}(\tau_n)]\}$$

Example 3. Considering $t \equiv (\lambda x.x\mathtt{I}x)\Delta$ from Ex. 1, with (tight) typing $\Psi \rhd \vdash_{\mathsf{mx}}^{(8,2)} (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}$ and expanded term $t_0 \equiv (\lambda x_3 x_4.x_3\mathtt{I}\mathtt{I}x_4)(\lambda x_1 x_2.x_1 x_2)(\lambda x.xx)$ (see Fig. 4), we present the $\mathcal{WL}$ typing corresponding to its expansion steps:

Unexpanded copy of Δ given by $\mathscr{E}(\lambda x.xx : \bullet_{\mathcal{M}}) \lhd (\lambda x.xx, \emptyset)$ has a $\mathcal{WL}$ typing Φ no different from the corresponding $\mathcal{MX}$ typing:

$$\dfrac{\dfrac{\overline{x : [\bullet_{\mathcal{N}}] \vdash_{\mathsf{wl}}^{(0,0)} x : \bullet_{\mathcal{N}}} \quad \overline{x : [\bullet_{\mathcal{N}}] \vdash_{\mathsf{wl}}^{(0,0)} x : \bullet_{\mathcal{N}}}}{x : [\bullet_{\mathcal{N}}, \bullet_{\mathcal{N}}] \vdash_{\mathsf{wl}}^{(0,1)} xx : \bullet_{\mathcal{N}}}}{\vdash_{\mathsf{wl}}^{(0,2)} \lambda x.xx : \bullet_{\mathcal{M}}}$$

On the other side, the expanded copy of Δ given by $\mathscr{E}(\lambda x.xx : \mathcal{B} \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}) \triangleleft (\lambda x_1 x_2.x_1 x_2, \emptyset)$ for $\mathcal{B} = [[[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}] \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}, [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}]$ has the corresponding $\mathcal{WL}$ typing Φ', for $\varphi_1 = [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}$, of the form:

$$\dfrac{\dfrac{\dfrac{}{x_1 : [[\varphi_1] \to \varphi_1] \vdash^{(0,0)}_{\mathsf{wl}} x_1 : [\varphi_1] \to \varphi_1} \quad \dfrac{\dfrac{}{x_2 : [\varphi_1] \vdash^{(0,0)}_{\mathsf{wl}} x_2 : \varphi_1}}{x_2 : [\varphi_1] \vdash^{(0,0)}_{\mathsf{wl}} x_2 : [\varphi_1]}}{x_2 : [\varphi_1], x_1 : [[\varphi_1] \to \varphi_1] \vdash^{(1,0)}_{\mathsf{wl}} x_1 x_2 : \varphi_1}}{x_1 : [[\varphi_1] \to \varphi_1] \vdash^{(2,0)}_{\mathsf{wl}} \lambda x_2.x_1 x_2 : [\varphi_1] \to \varphi_1}}{\vdash^{(3,0)}_{\mathsf{wl}} \lambda x_1 x_2.x_1 x_2 : [[\varphi_1] \to \varphi_1] \to [\varphi_1] \to \varphi_1}$$

where $\mathtt{w}(\mathcal{B} \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}) = \mathtt{w}([[\varphi_1] \to \varphi_1, \varphi_1] \to \varphi_1) = [[\varphi_1] \to \varphi_1] \to [\varphi_1] \to \varphi_1$.

Finally, given expansion $\mathscr{E}(\lambda x.x\mathtt{I}x : \mathcal{A} \to \bullet_{\mathcal{M}}) \triangleleft (\lambda x_3 x_4.x_3 \mathtt{II} x_4, \emptyset)$ for $\mathcal{A} = [\bullet_{\mathcal{M}}, \mathcal{B} \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}]$, let $\varphi_2 = [\varphi_1] \to \varphi_1$ and $\varphi_3 = [\varphi_2] \to \varphi_2$. From $\mathcal{WL}$ typings $\Psi(\varphi_1)$ and $\Psi(\bullet_{\mathcal{M}})$:

$$\dfrac{\dfrac{\dfrac{}{x_5 : [\varphi_1] \vdash^{(0,0)}_{\mathsf{wl}} x_5 : \varphi_1}}{\vdash^{(1,0)}_{\mathsf{wl}} \lambda x_5.x_5 : [\varphi_1] \to \varphi_1}}{\vdash^{(1,0)}_{\mathsf{wl}} \lambda x_5.x_5 : [[\varphi_1] \to \varphi_1]} \qquad \dfrac{\dfrac{\dfrac{}{x_6 : [\bullet_{\mathcal{M}}] \vdash^{(0,0)}_{\mathsf{wl}} x_6 : \bullet_{\mathcal{M}}}}{\vdash^{(1,0)}_{\mathsf{wl}} \lambda x_6.x_6 : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}}{\vdash^{(1,0)}_{\mathsf{wl}} \lambda x_6.x_6 : [[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}]}$$

we have derivation Ψ' of the form:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x_3 : [\varphi_3] \vdash^{(0,0)}_{\mathsf{wl}} x_3 : \varphi_3} \quad \Psi(\varphi_1)}{x_3 : [\varphi_3] \vdash^{(2,0)}_{\mathsf{wl}} x_3\mathtt{I} : \varphi_2} \quad \Psi(\bullet_{\mathcal{M}})}{x_3 : [\varphi_3] \vdash^{(4,0)}_{\mathsf{wl}} x_3\mathtt{II} : \varphi_1} \quad \dfrac{\dfrac{}{x_4 : [\bullet_{\mathcal{M}}] \vdash^{(0,0)}_{\mathsf{wl}} x_4 : \bullet_{\mathcal{M}}}}{x_4 : [\bullet_{\mathcal{M}}] \vdash^{(0,0)}_{\mathsf{wl}} x_4 : [\bullet_{\mathcal{M}}]}}{x_3 : [\varphi_3], x_4 : [\bullet_{\mathcal{M}}] \vdash^{(5,0)}_{\mathsf{wl}} x_3\mathtt{II}x_4 : \bullet_{\mathcal{M}}}}{x_3 : [\varphi_3] \vdash^{(6,0)}_{\mathsf{wl}} \lambda x_4.x_3\mathtt{II}x_4 : \varphi_1}}{\vdash^{(7,0)}_{\mathsf{wl}} \lambda x_3 x_4.x_3\mathtt{II}x_4 : [\varphi_3] \to \varphi_1}$$

where $\mathtt{w}(\mathcal{A} \to \bullet_{\mathcal{M}}) = [\varphi_3] \to \varphi_1$. Then we get the (tight) typing $\Psi \rhd \vdash^{(12,2)}_{\mathsf{wl}} t_0 : \bullet_{\mathcal{M}}$.

We prove that any expanded term is typable in system $\mathcal{WL}$.

Lemma 8. *If $\mathscr{E}(t : \sigma) \triangleleft (t_1, A)$, then there is a derivation $\Psi \rhd \Gamma^e_A \vdash^{(b,r)}_{\mathsf{wl}} t_1 : \mathtt{w}(\sigma)$. Moreover, if the expansion is (garbage-)tight, so is Ψ.*

Proof. By induction on t. We present the case when $t = t'u'$:

- $\mathscr{E}(t'u' : \sigma) \triangleleft (t_0 u'_1 \dots u'_m, B_0 + \dots + B_m)$, if for some $m > 0$ and $\sigma_1, \dots, \sigma_m$, $\mathscr{E}(t' : [\sigma_j]_{j=1\dots m} \to \sigma) \triangleleft (t_0, B_0)$ and $\left(\mathscr{E}(u' : \sigma_j) \triangleleft (u'_j, B_j)\right)_{j=1\dots m}$. By *i.h.*, $\Gamma^e_{B_0} \vdash^{(b_0,r_0)} t_0 : [\mathtt{w}(\sigma_1)] \to \dots \to [\mathtt{w}(\sigma_m)] \to \mathtt{w}(\sigma)$ and $\left(\Gamma^e_{B_j} \vdash^{(b_j,r_j)} u'_j : \mathtt{w}(\sigma_j)\right)_{j=1\dots m}$. Therefore $+_{j=0\dots m}\Gamma^e_{B_j} \vdash^{(m+_{j=0\dots m}b_j, +_{j=0\dots m}r_j)} t_0 u'_1 \dots u'_m : \mathtt{w}(\sigma)$ by m applications of rules (`one`) and (`app`), where $+_{j=0\dots m}\Gamma^e_{B_j} = \Gamma^e_{+_{j=0\dots m}B_j}$.

- $\mathscr{E}(t'u' : \sigma) \triangleleft (t_0 u'_1, B_0 + B_1)$, if $\mathscr{E}(t' : [\,] \to \sigma) \triangleleft (t_0, B_0)$ and $\mathscr{E}(u' : \tau) \triangleleft (u'_1, B_1)$ for some type τ. By the *i.h.*, $\Gamma^e_{B_0} \vdash^{(b,r)} t_0 : [\,] \to \mathtt{w}(\sigma)$ and $\Gamma^e_{B_1} \vdash^{(b',r')} u'_1 : \mathtt{w}(\tau)$. Therefore $\Gamma^e_{B_0} + \Gamma^e_{B_1} \vdash^{(b+b'+1, r+r')} t_0 u'_1 : \mathtt{w}(\sigma)$ by rules ($\mathtt{none}$) and ($\mathtt{app}_b$), where $\Gamma^e_{B_0} + \Gamma^e_{B_1} = \Gamma^e_{B_0+B_1}$.
- $\mathscr{E}(t'u' : \bullet_{\mathcal{N}}) \triangleleft (t^* u^*, B_1 + B_2)$, follows from $\mathscr{E}(t' : \bullet_{\mathcal{N}}) \triangleleft (t^*, B_1)$ and $\mathscr{E}(u' : \mathtt{t}) \triangleleft (u^*, B_2)$. By *i.h.*, $\Gamma^e_{B_1} \vdash^{(b,r)} t^* : \bullet_{\mathcal{N}}$ and $\Gamma^e_{B_2} \vdash^{(b',r')} u^* : \mathtt{t}$. By rule ($\mathtt{app}^{lo}_r$) we have $\Gamma^e_{B_1} + \Gamma^e_{B_2} \vdash^{(b+b', r+r'+1)} t^* u^* : \bullet_{\mathcal{N}}$, where $\Gamma^e_{B_1} + \Gamma^e_{B_2} = \Gamma^e_{B_1+B_2}$.

Theorem 5. *If $\mathscr{E}(t : \sigma) \triangleleft (t', A)$ is tight, then t' is weak-linear.*

Proof. Let $\mathscr{E}(t : \sigma) \triangleleft (t', A)$ be a tight expansion. By Lemma 8 there is a tight $\Phi \rhd \Gamma^e_A \vdash^{(b,r)}_{\mathsf{wl}} t' : \mathtt{w}(\sigma)$, then by Thm. 4 t' is weak-linear.

6 Related Works and Discussions

As we mentioned before, the notion of linearisation of the λ-calculus, was first introduced in [27]. Directly related to this notion is the use of linearisation in hardware compilation [19, 20] for synthesising digital circuits, in which a notion of serialization is defined to deal with contraction in concurrent contexts, based on type information that indicates bounds for the usage of functions. Kfoury's linearisation is also closely related to the notion of affine approximations by Mazza et al. [31], which identifies distinct occurrences of variables that apply linearly to sequences of arguments. A λ-term admits a simply-typed affine approximation if and only if it has an intersection type (in a non-idempotent and non-commutative intersection type system).

Non-idempotent types are crucial in the notion of linearisation defined by Kfoury, similarly to what happens in our work. However the linearisation process defined in [27] was not presented as a direct encoding of λ-terms into terms of the new calculus. Furthermore, intersections in Kfoury's work are also non-commutative, which differs from our approach and has the effect of excluding valid expanded terms in the sense that they still provide an effective simulation of the initial term: for example $(\lambda x_1 x_2 . x_1 x_2) II$ and $(\lambda x_1 x_2 . x_2 x_1) II$ can both be considered valid expansions of the non-linear term $(\lambda x . xx) I$. This leads to one interesting open question regarding the relation between the two approaches.

Our notion of expansion is directly related to the notion of expansion defined by Florido and Damas [17], where non-idempotent intersections were also considered and the target language was the linear λ-calculus. One of the implications of the expansion relation in [17] is that, any term typable using intersection types, can be expanded to a term typable using simple types, similarly to what happens with affine approximations. Also, Bucciarelli et al. [13], have given a translation of intersection typing derivations into Curry typeable terms, that preserves β-reduction, since the target language is the simply typed λ-calculus. The expansion relation in [17] is not preserved by β-reduction, but it is preserved by weak-head reduction. Note that taking into account different algebraic properties of $\cap$, can lead to different expansion relations and different properties on preservation of reduction.

Our goal here is to have preservation of normal forms and our target is the weak linear λ-calculus, introduced in [6], in which the notion of linearity is an operational one: functions are required to be linear, if they are ever applied. Redexes that are eventually contracted, which correspond to the notion of virtual redexes, are as paths in the initial term, deriving from Lévy's definition of labelled reduction [29]. In this work, unlike the approach by Kfoury, the λ-calculus is simulated by a proper *linear* subset of the calculus, rather than by a non-standard calculus. Weak linear terms have good properties: non-duplicating reduction, therefore strong normalisation; it is decidable to know if a λ-term is weak linear; and type inference for weak linear terms is both decidable and polynomial. Linearisation from standard λ-terms into weak linear terms was defined by computing legal paths [8] on the initial term, identifying virtual redexes. The linearisation procedure preserves β-normal forms, provided that paths are chosen in the right way. Furthermore, if the procedure terminates then it yields a weak linear λ-term. However, there is a conjecture regarding termination: the linearisation procedure is a total function for the strongly normalising λ-calculus. Note that a similar conjecture exists for Kfoury's linearisation.

Our expansion relies on a quantitative type system that was used in [1] to extract exact measures for a maximal reduction strategy, which is closely related to the reduction strategy used in this paper. Both typing system and strategy in [1] are annotated with measures, which are not relevant to our expansion process, but which are essential to prove some properties of our expansion. We also believe they can provide an appropriate framework to reason about quantitative aspects of the expansion relation.

In [6], weak linear λ-terms proved to be typable in polynomial time. This was achieved by the definition of a typing system, which we will refer to as $\mathcal{WA}$, and a corresponding type inference algorithm that proved to type any weak linear λ-term in polynomial time. System $\mathcal{WA}$ can be seen as a generalization of Hindley's system of simple types [22, 23]. A tight version of system $\mathcal{WA}$, where tight types are used to type functions that are not applied, can be seen as an intermediate system between $\mathcal{WL}$ and $\mathcal{WA}$. Translations can be defined between the systems where tight-types, which can be seen as type constants with no particular meaning, can be replaced by an appropriate type, therefore typability in polynomial time is also a property of the class of terms obtained by our expansion relation.

7 Conclusions and Future Work

In this paper we have defined an expansion relation between strongly normalising λ-terms and weak linear λ-terms with the same normal form. This expansion relation relies on a quantitative typing system and on the notion of tight typings, which were also used to give an exact typing characterisation of the class of weak linear λ-terms. This work provided an answer to a conjecture that stated that all strongly normalising λ-terms can be encoded as weak linear λ-terms.

To guide our expansion we used a quantitative typing system that provides measures for reduction sequences as well as the size of corresponding normal forms. One interesting question that is left for future work is to investigate how the expansion process affects these measures. Due to its operational linearity, weak linear λ-terms reduce in linear time, but the size of the expanded terms can be exponential with respect to the size of the term. Our intuition is that this could potentially show a nice interplay between time and space in computation.

Another relevant question that is left for future work is the relation between the approach by Kfoury and our approach. As mentioned before, Kfoury relies on non-commutative intersection types, so there is possibly a class of weak linear terms corresponding to a unique expanded term in $\Lambda^\wedge$. Establishing a relation between the two approaches could also provide an answer to the conjecture in [27].

References

1. B. Accattoli, S. Graham-Lengrand, and D. Kesner. Tight typings and split bounds, fully developed. *J. Funct. Program.*, 30:e14, 2020.
2. B. Accattoli and G. Guerrieri. Types of fireballs. In *APLAS*, volume 11275 of *Lecture Notes in Computer Science*, pages 45–66. Springer, 2018.
3. B. Accattoli, G. Guerrieri, and M. Leberle. Types by need. In *ESOP*, volume 11423 of *Lecture Notes in Computer Science*, pages 410–439. Springer, 2019.
4. S. Alves. *Linearisation of the Lambda Calculus*. PhD thesis, University of Porto, April 2007.
5. S. Alves, M. Fernández, M. Florido, and I. Mackie. Linearity and recursion in a typed lambda-calculus. In *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and Practices of declarative programming*, pages 173–182. ACM, 2011.
6. S. Alves and M. Florido. Weak linearization of the lambda calculus. *Theor. Comput. Sci.*, 342(1):79–103, 2005.
7. S. Alves, D. Kesner, and D. Ventura. A quantitative understanding of pattern matching. In *25th International Conference on Types for Proofs and Programs*, volume 175 of *LIPIcs*, pages 3:1–3:36. Schloss Dagstuhl, 2019.
8. A. Asperti and C. Laneve. Paths, computations and labels in the λ-calculus. *Theoretical Computer Science*, 142(2):277–297, 1995.
9. T. Balabonski, P. Barenbaum, E. Bonelli, and D. Kesner. Foundations of strong call by need. *Proc. ACM Program. Lang.*, 1(ICFP):20:1–20:29, 2017.
10. H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in logic and the foundation of mathematics*. North-Holland, 1984.
11. H. P. Barendregt, J. Bergstra, J. W. Klop, and H. Volken. Degrees, reductions and representability in the lambda calculus. Technical Report Preprint no.22, University of Utrecht, Department of Mathematics, 1976.
12. A. Bucciarelli, D. Kesner, and D. Ventura. Non-idempotent intersection types for the lambda-calculus. *Log. J. IGPL*, 25(4):431–464, 2017.
13. A. Bucciarelli, S. D. Lorenzis, A. Piperno, and I. Salvo. Some computational properties of intersection types. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 109–118, 1999.
14. M. Coppo and M. Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame J. Form. Log.*, 21(4):685–693, 1980.

15. H. B. Curry. Functionality in combinatory logic. In *Proceedings of the National Academy of Sciences U.S.A.*, volume 20, pages 584–590, 1934.
16. D. de Carvalho. *Sémantiques de la logique linéaire et temps de calcul*. Phd thesis, Université Aix-Marseille II, 2007.
17. M. Florido and L. Damas. Linearization of the lambda-calculus and its relation with intersection type systems. *J. Funct. Program.*, 14(5):519–546, 2004.
18. P. Gardner. Discovering needed reductions using type theory. In *TACS*, volume 789 of *LNCS*, pages 555–574. Springer, 1994.
19. D. R. Ghica. Geometry of synthesis: a structured approach to VLSI design. In *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 363–375. ACM, 2007.
20. D. R. Ghica and A. Smith. Geometry of synthesis iii: Resource management through type inference. In *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 345–356. ACM, 2011.
21. J. Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.
22. J. R. Hindley. The principal type-scheme of an object in combinatory logic. *Trans. American Math. Soc.*, 146:29–60, 1969.
23. J. R. Hindley. *Basic Simple Type Theory*. Cambridge University Press, 1997.
24. D. Kesner and D. Ventura. A resource aware semantics for a focused intuitionistic calculus. *Math. Struct. Comput. Sci.*, 29(1):93–126, 2019.
25. D. Kesner and P. Vial. Consuming and persistent types for classical logic. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, page 619–632. ACM, 2020.
26. D. Kesner and P. Vial. Non-idempotent types for classical calculi in natural deduction style. *Log. Methods Comput. Sci.*, 16(1), 2020.
27. A. Kfoury. A linearization of the lambda-calculus and consequences. *Journal of Logic and Computation*, 10(3):411–436, 2000.
28. Y. Lafont. The linear abstract machine. *Theoretical Computer Science*, 59:157–180, 1988.
29. J.-J. Lévy. *Réductions correctes et optimales dans le lambda-calcul*. PhD thesis, Éditeur inconnu, 1978.
30. I. Mackie. The geometry of interaction machine. In *Proceedings of the 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 1995*, pages 198–208. ACM Press, 1995.
31. D. Mazza, L. Pellissier, and P. Vial. Polyadic approximations, fibrations and intersection types. *Proceedings of the ACM on Programming Languages*, 2(6), 2018.
32. F. van Raamsdonk, P. Severi, M. H. Sørensen, and H. Xi. Perpetual reductions in lambda-calculus. *Inf. Comput.*, 149(2):173–225, 1999.
33. D. Walker. Substructural type systems. In *Advanced Topics in Types and Programming Languages*, chapter 1, pages 3–43. MIT Press, Cambridge, 2005.