Bounding normalization time through intersection types

(Article begins on next page)

02 May 2024

Bounding normalization time through intersection types

Erika De Benedetti   Simona Ronchi Della Rocca

Dipartimento di Informatica
Università di Torino, Italy
debenede@di.unito.it   ronchi@di.unito.it

1   Introduction

Intersection types were originally introduced as idempotent, i.e., modulo the equivalence $\sigma \wedge \sigma = \sigma$. In fact, they have been used essentially for semantic purposes, for building filter models for λ-calculus, where the interpretation of types as properties of terms induces naturally the idempotence property.
Recently it has been observed that, when dropping idempotency, intersection types can be used for reasoning about the complexity of β-reduction. Some results have been already obtained along this line. Terui [15] designed a system assigning non-idempotent intersection types to λ-calculus, which can type all and only the strongly normalizing terms, and such that the size of any derivation with subject M is bigger than the size of every term in the β-reduction sequence from M to its normal form. This property can be used for computing a bound of every normalizing β-reduction sequence starting from M. A more precise result in this direction has been obtained by Lengrand [2], who gave a precise measure of the number of β-reduction steps. Namely he designed a type assignment system, where intersection is considered without idempotency, and defined the notions of measure of derivation and of principal derivation for a given term. Then he proved that the measure of a principal derivation of a type for a normalizing term M corresponds to the maximal length of a normalizing β-reduction sequence for M.
In this line, we go one step forward, and use intersection types without neither idempotence nor associativity to express the functional dependence of the length of a normalizing β-reduction sequence from a term M on the size of M itself . In order to obtain such a result, we take inspiration from the system STA of Gaboardi and Ronchi Della Rocca [6], in its turn inspired by the Soft Linear Logic of Lafont [9], which characterizes the polynomial time computations. The resulting system allow us to give a bound on the number of steps necessary to reduce a normalizing term M to its normal form, in the form $|M|^{d+1}$, where $|M|$ is the size of the term, and d is a measure depending on the type derivation for it (the *depth*). Since for every normalizing term there is a type derivation with minimal depth, this bound does not depend on a particular derivation. A preliminary type assignment of this kind has been described in [1].

Some type assignment systems without idempotency have been already studied in the literature, for various purposes. Kfoury and Wells, in [8] used non-idempotent intersection in order to formalize a type inference semi-algorithm, whose complexity has been studied in [10]. Kfoury, in [7], connected non idempotent intersection types with linear β-reduction. Recently non idempotent intersection types have been used by Pagani and Ronchi Della Rocca for characterizing the solvability in the resource λ-calculus [12, 13]. In [5] the game semantics of a typed λ-calculus has been described in logical form using an intersection type assignment system where the intersection is not idempotent neither commutative nor associative. Some complexity results have been obtained by De Carvalho in [4], using a λ-algebra induced by non idempotent types. Recently a logical description of relational model of λ-calculus [3] has been designed, through a non-idempotent type assigment system [14].

2   System `STI`

We start by introducing `STI` (Soft Type assignment with Intersection), a type assignment system for λ-calculus assigning to λ-terms non-idempotent and not associative intersection types. The system assigns types to all and only strongly normalizing terms.

Definition 1.   *i) Terms of λ-calculus are defined by the following grammar:*

$$M ::= x \mid MM \mid \lambda x.M$$

where x *ranges over a countable set ot variables. The symbol* $\equiv$ *denotes the syntactical equality modulo renaming of bound variables.*

ii) The reduction relation $\underset{\beta}{\longrightarrow}$ *is the contextual closure of the rule* $(\lambda x.M)N \to M[N/x]$, *where* $M[N/x]$ *is the capture-free substitution of* N *to all the free occurrences of* x *in* M. $\underset{\beta}{\overset{*}{\longrightarrow}}$ *is the reflexive and transitive closure of* $\underset{\beta}{\longrightarrow}$.

iii) A term M *is an instance of* N *if it is obtained from* N *by renaming a subset of its free variables with a unique fresh name.*

iv) The set of `STI` *types is defined as follows:*

$$
\begin{aligned}
A &::= a \mid \sigma \to A & \text{\textit{(linear types)}} \\
\sigma &::= A \mid \underbrace{\sigma \wedge ... \wedge \sigma}_{n} \quad (n > 1) & \text{\textit{(intersection types)}}
\end{aligned}
$$

where a *ranges over a countable set of type variables. Linear types are ranged over by* A, B, C, *intersection types by* σ, τ, ρ. *The connective* $\wedge$ *is commutative, but it is not idempotent nor associative. The number of elements of a type is defined inductively as* $l(A) = 1$, $l(\sigma_1 \wedge ... \wedge \sigma_n) = l(\sigma_1) + ... + l(\sigma_n)$.

v) A context is a finite set of assumptions of the shape x : σ, *where* x *is a variable and* σ *is a type. Variables in a context are all distinct, and contexts are ranged over by* Γ, Δ. *dom*(Γ) *is the set* $\{x \mid x : \sigma \in \Gamma\}$. *The intersection of contexts is given by*

$$\Gamma \wedge \Delta = \{x : \sigma \mid x : \sigma \in \Gamma, x \notin dom(\Delta)\} \cup \{x : \tau \mid x : \tau \in \Delta, x \notin dom(\Gamma)\} \cup \{x : \sigma \wedge \tau \mid x : \sigma \in \Gamma, x : \tau \in \Delta\}$$

while Γ, Δ *represents the union of sets* Γ *and* Δ, *provided that* $\Gamma \# \Delta$, *i.e.* $dom(\Gamma) \cap dom(\Delta) = \emptyset$.

vi) The system `STI` *proves sequents of the shape* $\Gamma \vdash M : \sigma$, *where* Γ *is a context,* M *is a term of λ-calculus, and* σ *is a type. The rules are given in Table* 1.

vii) Derivations are denoted by Π, Σ. $\Pi \triangleright \Gamma \vdash M : \sigma$ *denotes a derivation* Π *with conclusion* $\Gamma \vdash M : \sigma$.

Some comments are in order. Since the condition on contexts in rule $(\to E)$, terms are built in a linear form, and an explicit multiplexor rule is present (rule (m)). This allows to control the number of (multiple) contractions, which is responsible for the growth of the reduction time. The counterpart of the contraction on the right side of a derivation is the rule $(\wedge_n)$, which is parametric in *n*. In doing this, we were inspired by the Soft Linear Logic of Lafont.

Let us define *constructive* the rules, which contribute in building the subject, i.e., either (*Ax*), or $(\to I)$ or $(\to E)$).

$$\frac{}{\mathtt{x} : A \vdash \mathtt{x} : A} \ (Ax) \qquad \frac{\Gamma \vdash \mathtt{M} : \sigma \quad \mathtt{x} \notin \mathrm{dom}(\Gamma)}{\Gamma, \mathtt{x} : A \vdash \mathtt{M} : \sigma} \ (w)$$

$$\frac{\Gamma, \mathtt{x} : \sigma \vdash \mathtt{M} : A}{\Gamma \vdash \lambda \mathtt{x}.\mathtt{M} : \sigma \to A} \ (\to I) \qquad \frac{\Gamma \vdash \mathtt{M} : \sigma \to A \quad \Delta \vdash \mathtt{N} : \sigma \quad \Gamma \# \Delta}{\Gamma, \Delta \vdash \mathtt{MN} : A} \ (\to E)$$

$$\frac{\Gamma_1 \vdash \mathtt{M} : \sigma_1 \quad ... \quad \Gamma_n \vdash \mathtt{M} : \sigma_n \quad n > 1}{\bigwedge_{i=1}^{n} \Gamma_i \vdash \mathtt{M} : \sigma_1 \wedge ... \wedge \sigma_n} \ (\wedge_n)$$

$$\frac{\Gamma, \mathtt{x}_1 : \sigma_1, ..., \mathtt{x}_n : \sigma_n \vdash \mathtt{M} : \tau}{\Gamma, \mathtt{x} : \sigma_1 \wedge ... \wedge \sigma_n \vdash \mathtt{M}[\mathtt{x}/\mathtt{x}_1, ... , \mathtt{x}/\mathtt{x}_n] : \tau} \ (m)$$

Table 1: The type assignment system

Definition 2 (Intersection trees). *Let* (δ) *be a (possibly empty) sequence of applications of rules* (w) *and* (m). *An* intersection tree *is a maximal (sub)proof of the shape defined inductively in the following way:*

- *Let the last rule of Σ be a constructive rule . Then*

$$\frac{\Sigma}{\Gamma \vdash M : \sigma} \ (\delta)$$

  is an empty intersection tree, with conclusion $\Gamma \vdash M : \sigma$ and one leaf Σ.

- *If Σ_i is a (possibly empty) intersection tree $(1 \le i \le n)$, then*

$$\frac{\dfrac{\Sigma_i \rhd \Gamma_i \vdash \mathtt{M} : \sigma_i \quad (1 \le i \le n)}{\bigwedge_{i=1}^{n} \Gamma_i \vdash \mathtt{M} : \sigma_1 \wedge ... \wedge \sigma_n} \ (\wedge_n)}{\Gamma \vdash \mathtt{M}' : \sigma} \ (\delta)$$

  is an intersection tree, with conclusion $\Gamma \vdash \mathtt{M}' : \sigma$, where $\mathtt{M}'$ is an instance of $\mathtt{M}$, Γ is a contraction of $\bigwedge_{i=1}^{n} \Gamma_i$, and its leafs are the leafs of all the Σ_i.

Since the $(\wedge_n)$ rule is the only rule building an intersection type on the right of the turnstile symbol, it is possible to state the following, which is a key property for proving the normalization bound.

Property 1 (Subject with intersection type). *Let $\Pi \rhd \Gamma \vdash \mathtt{M} : \sigma_1 \wedge ... \wedge \sigma_m$ with $m > 1$. Then Π ends with a non empty intersection tree.*

Proof. By induction on the shape of Π. If the last applied rule is $(\wedge_n)$, then the statement is trivially true and δ is the empty sequence. Otherwise, the derivation needs to contain at least one application of rule $(\wedge_n)$, with subject $\mathtt{M}'$, such that $\mathtt{M}$ is an instance of $\mathtt{M}'$. Then this application can be followed only by δ of rules, which can contain only applications of rule (w) or rule (m). $\square$

The substitution property holds for terms having disjoint free variables sets.

Lemma 1 (Substitution). *Let $\Pi \rhd \Gamma, \mathtt{x} : \sigma \vdash \mathtt{M} : \tau$, $\Sigma \rhd \Delta \vdash \mathtt{N} : \sigma$, $\Gamma \# \Delta$ and $\mathtt{x} \notin dom(\Delta)$.*
Then there exists $S(\Sigma, \Pi)$ such that $S(\Sigma, \Pi) \rhd \Gamma, \Delta \vdash \mathtt{M}[\mathtt{N}/\mathtt{x}] : \tau$.

Proof. By induction on the shape of Π. The proof is trivial except for the cases of (w), $(\wedge_n)$ or (m).

If (w) is the last applied rule introducing a variable $y \neq x$, then the proof follows by induction. Otherwise, let Π be the proof

$$\frac{\Pi' \triangleright \Gamma \vdash \mathtt{M} : \sigma \quad \mathtt{x} \notin \mathrm{dom}(\Gamma)}{\Pi \triangleright \Gamma, \mathtt{x} : \mathtt{A} \vdash \mathtt{M} : \sigma} \ (w)$$

and let $\Sigma \triangleright \Delta \vdash \mathtt{N} : \mathtt{A}$. If Δ contains only bindings of variables to linear types, then $S(\Sigma, \Pi)$ is the proof

$$\frac{\Pi' \triangleright \Gamma \vdash \mathtt{M} : \sigma}{S(\Sigma, \Pi) \triangleright \Gamma, \Delta \vdash \mathtt{M} : \sigma} \ (w)$$

Otherwise, let us assume, without loss of generality, $\Delta = \Delta', \mathtt{y} : \tau$ such that $\mathtt{A}_1, \dots, \mathtt{A}_n$ are the elements of τ, and let Δ' contain only bindings of variables to linear types. Then $S(\Sigma, \Pi)$ is the proof

$$\frac{\dfrac{\Pi' \triangleright \Gamma \vdash \mathtt{M} : \sigma}{\Gamma, \Delta', \mathtt{y}_1 : \mathtt{A}_1, \dots, \mathtt{y}_n : \mathtt{A}_n \vdash \mathtt{M} : \sigma} \ (w)}{S(\Sigma, \Pi) \triangleright \Gamma, \Delta', \mathtt{y} : \tau \vdash \mathtt{M} : \sigma} \ (m)$$

where the sequence of applications of rule (m) is constructiong τ.

If the last applied rule is $(\wedge_n)$, with $n > 1$, then Π is of the shape

$$\frac{\Pi_1 \triangleright \Gamma_1, \mathtt{x} : \sigma_1 \vdash \mathtt{M} : \tau_1 \quad \dots \quad \Pi_n \triangleright \Gamma_n, \mathtt{x} : \sigma_n \vdash \mathtt{M} : \tau_n}{\Gamma, \mathtt{x} : \sigma_1 \wedge \dots \wedge \sigma_n \vdash \mathtt{M} : \tau_1 \wedge \dots \wedge \tau_n} \ (\wedge_n)$$

By Property 1, Σ is of the shape

$$\frac{\dfrac{\Sigma_1 \triangleright \Delta_1 \vdash \mathtt{N}' : \sigma_1 \quad \dots \quad \Sigma_n \triangleright \Delta_n \vdash \mathtt{N}' : \sigma_n}{\Delta' \vdash \mathtt{N}' : \sigma_1 \wedge \dots \wedge \sigma_n} \ (\wedge_n)}{\Sigma \triangleright \Delta \vdash \mathtt{N} : \sigma_1 \wedge \dots \wedge \sigma_n} \ (\delta)$$

where δ is a sequence of applications of (w) and (m) rules, and $\mathtt{N}$ is an instance of $\mathtt{N}'$.

By inductive hypothesis $S(\Sigma_i, \Pi_i) \triangleright \Gamma_i, \Delta_i \vdash \mathtt{M}[\mathtt{N}'/\mathtt{x}] : \tau_i$, since $\Gamma \# \Delta$ implies $\Gamma_i \# \Delta_i$ for all i, so $S(\Sigma, \Pi)$ is given by

$$\frac{\dfrac{S(\Sigma_1, \Pi_1) \triangleright \Gamma_1, \Delta_1 \vdash \mathtt{M}[\mathtt{N}'/\mathtt{x}] : \tau_1 \quad \dots \quad S(\Sigma_n, \Pi_n) \triangleright \Gamma_n, \Delta_n \vdash \mathtt{M}[\mathtt{N}'/\mathtt{x}] : \tau_n}{\Gamma, \Delta' \vdash \mathtt{M}[\mathtt{N}'/\mathtt{x}] : \tau_1 \wedge \dots \wedge \tau_n} \ (\wedge_n)}{\Gamma, \Delta \vdash \mathtt{M}[\mathtt{N}/\mathtt{x}] : \tau_1 \wedge \dots \wedge \tau_n} \ (\delta)$$

If the last applied rule is (m), then Π is of the shape

$$\frac{\Pi' \triangleright \Gamma, \mathtt{x}_1 : \sigma_1, \dots, \mathtt{x}_n : \sigma_n \vdash \mathtt{M} : \tau}{\Gamma, \mathtt{x} : \sigma_1 \wedge \dots \wedge \sigma_n \vdash \mathtt{M}[\mathtt{x}/\mathtt{x}_1, \dots, \mathtt{x}/\mathtt{x}_n] : \tau} \ (m)$$

Exactly as in the previous case, we can apply Property 1 to Σ, thus obtaining $\Sigma_i \triangleright \Delta_i \vdash \mathtt{N}' : \sigma_i$, for $1 \leq i \leq n$. Also, we must rename the variables in Δ_i, so that we actually get proofs $\Sigma_i' \triangleright \Delta_i' \vdash \mathtt{N}_i' : \sigma_i$ where $\mathtt{N}_i'$ is an instance of $\mathtt{N}'$ and all $\mathrm{dom}(\Delta_i')$ are disjoint from each other; this is not a trouble as we will be able to recover Δ' and $\mathtt{N}'$ easily by a suitable sequence ρ of applications of (m) rules.

By induction we can now build

$$S(\Sigma_1', \Pi') \triangleright \Gamma, \Delta_1', \mathtt{x}_2 : \sigma_2, \dots, \mathtt{x}_n : \sigma_n \vdash \mathtt{M}[\mathtt{N}_1'/\mathtt{x}_1] : \tau$$

$$S(\Sigma_2', S(\Sigma_1', \Pi')) \triangleright \Gamma, \Delta_1', \Delta_2', \mathtt{x}_3 : \sigma_3, \dots, \mathtt{x}_n : \sigma_n \vdash \mathtt{M}[\mathtt{N}_1'/\mathtt{x}_1, \dots, \mathtt{N}_2'/\mathtt{x}_2] : \tau$$

$$\vdots$$

$$S(\Sigma_n', S(\Sigma_{n-1}', ...S(\Sigma_1', \Pi')...)) \rhd \Gamma, \Delta_1', ..., \Delta_n' \vdash \texttt{M}[\texttt{N}_1'/\texttt{x}_1, ... , \texttt{N}_n'/\texttt{x}_n] : \tau$$

and by applying sequences ρ and δ of rule (m), we get the desired proof

$$\cfrac{\cfrac{\Gamma, \Delta_1', ..., \Delta_n' \vdash \texttt{M}[\texttt{N}_1'/\texttt{x}_1, ... , \texttt{N}_n'/\texttt{x}_n] : \tau}{\Gamma, \Delta' \vdash \texttt{M}[\texttt{N}'/\texttt{x}_1, ... , \texttt{N}'/\texttt{x}_n] : \tau} \ (\rho)}{S(\Sigma, \Pi) \rhd \Gamma, \Delta \vdash \texttt{M}[\texttt{N}/\texttt{x}_1, ... , \texttt{N}/\texttt{x}_n] : \tau} \ (\delta)$$

$\square$

The substitution property is sufficient for proving the subject reduction property, but we need to take into account that one step of β-reduction on the subject can be matched by a *set* of $n \geq 1$ parallel simplification steps in the underlying derivation, corresponding to reducing virtual copies of the same redex having different types.

Property 2 (Subject reduction). $\Pi \rhd \Gamma \vdash \texttt{M} : \sigma$ *and* $\texttt{M} \xrightarrow{\beta} \texttt{M}'$ *implies* $\Pi' \rhd \Gamma \vdash \texttt{M}' : \sigma$.

Proof. $\texttt{M} \xrightarrow{\beta} \texttt{M}'$ means $\texttt{M} = C[(\lambda \texttt{x}.\texttt{Q})\texttt{N}]$ and $\texttt{M}' = C[\texttt{Q}[\texttt{N}/\texttt{x}]]$, for some context $C[.]$. The proof is by induction on $C[.]$. Let us consider just the base case in which $C[.] = [.]$, i.e., $\texttt{M} = (\lambda \texttt{x}.\texttt{Q})\texttt{N}$. Then the most difficult case is when Π ends by a non empty intersection tree. Note that the shape of $\texttt{M}$ implies each leaf Π_i of the intersection tree be of the shape:

$$\cfrac{\cfrac{\cfrac{\Sigma_i' \rhd \Gamma_i', \texttt{x} : \sigma_i \vdash \texttt{Q}_i' : \texttt{A}_i}{\Gamma_i' \vdash \lambda \texttt{x}.\texttt{Q}_i' : \sigma_i \to \texttt{A}_i} \ (\to I)}{\Gamma_i \vdash \lambda \texttt{x}.\texttt{Q}_i : \sigma_i \to \texttt{A}} \ (\delta_i) \qquad \Sigma_i'' \rhd \Delta_i \vdash \texttt{N}_i : \sigma_i \quad \Gamma_i \# \Delta_i}{\Gamma_i, \Delta_i \vdash (\lambda \texttt{x}.\texttt{Q}_i)\texttt{N}_i : \texttt{A}_i} \ (\to E)$$

where $1 \leq i \leq n$, for some $n > 1$, $(\lambda \texttt{x}.\texttt{Q})\texttt{N}$ is an instance of $(\lambda \texttt{x}.\texttt{Q}_i)\texttt{N}_i$, and δ_i is a (possibly empty) sequence of applications of (w) and (m) rules. Since all (m) rules in δ_i deal with variables in $\text{dom}(\Gamma)$, sequence δ_i can be delayed to obtain the proof

$$\cfrac{\cfrac{\cfrac{\Sigma_i' \rhd \Gamma_i', \texttt{x} : \sigma_i \vdash \texttt{Q}_i' : \texttt{A}}{\Gamma_i' \vdash \lambda \texttt{x}.\texttt{Q}_i' : \sigma_i \to \texttt{A}_i} \ (\to I) \qquad \Sigma_i'' \rhd \Delta_i \vdash \texttt{N}_i : \sigma_i}{\Gamma_i', \Delta_i \vdash (\lambda \texttt{x}.\texttt{Q}_i')\texttt{N}_i : \texttt{A}_i} \ (\to E)}{\Gamma_i, \Delta_i \vdash (\lambda \texttt{x}.\texttt{Q}_i)\texttt{N}_i : \texttt{A}_i} \ (\delta_i)$$

By Lemma 1, there are proofs $S(\Sigma_i'', \Sigma_i') \rhd \Gamma_i, \Delta_i \vdash \texttt{Q}_i[\texttt{N}_i/\texttt{x}]$, and then the result is obtained by replacing the leafs Π_i of the intersection tree by $S(\Sigma_i'', \Sigma_i')$ $(1 \leq i \leq n)$.

$\square$

Moreover the system is strongly normalizing. Formally:

Property 3 (Strong normalization). $\Pi \rhd \Gamma \vdash \texttt{M} : \sigma$ *if and only if* $\texttt{M}$ *is strongly normalizing.*

For the right implication, the proof is obtained in the next section by observing that the measure of Π decreases with each reduction step, and this does not depend on any particular strategy. As for the left implication, the proof can be obtained by adapting Neergaard's proof [11] to system STI. In fact, Neergaard proved the strong normalization property for a system with rigid intersection types, i.e. intersection without commutativity, associativity nor idempotency.

Example 1. *Here we will show an example of a derivation in* STI, *aiming to clarify the behaviour on the subject reduction in the case of a non-empy intersection tree. Let*

$$\cfrac{\cfrac{\overline{\text{y}:\text{A}\vdash\text{y}:\text{A}}\ (Ax)}{\vdash\lambda\text{y.y}:\text{A}\rightarrow\text{A}}\ (\rightarrow I)\qquad\overline{\text{z}:\text{A}\vdash\text{z}:\text{A}}\ (Ax)}{\Sigma_1\triangleright\text{z}:\text{A}\vdash(\lambda\text{y.y})\text{z}:\text{A}}\ (\rightarrow E)\qquad and\qquad\cfrac{\cfrac{\overline{\text{y}:\text{a}\vdash\text{y}:\text{a}}\ (Ax)}{\vdash\lambda\text{y.y}:\text{a}\rightarrow\text{a}}\ (\rightarrow I)\qquad\overline{\text{z}:\text{a}\vdash\text{z}:\text{a}}\ (Ax)}{\Sigma_2\triangleright\text{z}:\text{a}\vdash(\lambda\text{y.y})\text{z}:\text{a}}\ (\rightarrow E)$$

where $\text{A}=\text{a}\rightarrow\text{a}$.
We want to reduce the term $(\lambda\text{x.xx})((\lambda\text{y.y})\text{z})$ *to normal form; the derivation is the following:*

$$\cfrac{\cfrac{\cfrac{\cfrac{\overline{\text{x}_1:\text{A}\vdash\text{x}_1:\text{A}}\ (Ax)\quad\overline{\text{x}_2:\text{a}\vdash\text{x}_2:\text{a}}\ (Ax)}{\text{x}_1:\text{A},\text{x}_2:\text{a}\vdash\text{x}_1\text{x}_2:\text{a}}\ (\rightarrow E)}{\text{x}:\text{A}\wedge\text{a}\vdash\text{xx}:\text{a}}\ (m)}{\vdash\lambda\text{x.xx}:(\text{A}\wedge\text{a})\rightarrow\text{a}}\ (\rightarrow I)\qquad\cfrac{\Sigma_1\triangleright\text{z}:\text{A}\vdash\boxed{(\lambda\text{y.y})\text{z}}:\text{A}\quad\Sigma_2\triangleright\text{z}:\text{a}\vdash\boxed{(\lambda\text{y.y})\text{z}}:\text{a}}{\Sigma\triangleright\text{z}:\text{A}\wedge\text{a}\vdash(\lambda\text{y.y})\text{z}:\text{A}\wedge\text{a}}\ (\wedge_2)}{\text{z}:\text{A}\wedge\text{a}\vdash(\lambda\text{x.xx})((\lambda\text{y.y})\text{z}):\text{a}}\ (\rightarrow E)$$

Notice that, since Σ *ends by a non empty intersection tree, there are two "virtual" copies of the same redex; therefore, if we reduce the redex* $(\lambda\text{y.y})\text{z}$, *we get the following derivation:*

$$\cfrac{\cfrac{\cfrac{\cfrac{\overline{\text{x}_1:\text{A}\vdash\text{x}_1:\text{A}}\ (Ax)\quad\overline{\text{x}_2:\text{a}\vdash\text{x}_2:\text{a}}\ (Ax)}{\text{x}_1:\text{A},\text{x}_2:\text{a}\vdash\text{x}_1\text{x}_2:\text{a}}\ (\rightarrow E)}{\text{x}:\text{A}\wedge\text{a}\vdash\text{xx}:\text{a}}\ (m)}{\vdash\lambda\text{x.xx}:(\text{A}\wedge\text{a})\rightarrow\text{a}}\ (\rightarrow I)\qquad\cfrac{\cfrac{\overline{\text{z}:\text{A}\vdash\text{z}:\text{A}}\ (Ax)\quad\overline{\text{z}:\text{a}\vdash\text{z}:\text{a}}\ (Ax)}{\text{z}:\text{A}\wedge\text{a}\vdash\text{z}:\text{A}\wedge\text{a}}\ (\wedge_2)}{}}{\Pi\triangleright\text{z}:\text{A}\wedge\text{a}\vdash(\lambda\text{x.xx})\text{z}:\text{a}}\ (\rightarrow E)$$

where both the redexes of Σ_1 *and* Σ_2 *have been reduced.*
Finally, we reduce $(\lambda\text{x.xx})\text{z}$ *(easy, as* Π *ends with an empty intersection tree), obtaining the proof*

$$\cfrac{\cfrac{\overline{\text{z}_1:\text{A}\vdash\text{z}_1:\text{A}}\ (Ax)\quad\overline{\text{z}_2:\text{a}\vdash\text{z}_2:\text{a}}\ (Ax)}{\text{z}_1:\text{A},\text{z}_2:\text{a}\vdash\text{z}_1\text{z}_2:\text{a}}\ (\rightarrow E)}{\text{z}:\text{A}\wedge\text{a}\vdash\text{zz}:\text{a}}\ (m)$$

Notice that, as explained in the proof for Lemma 1, the premises of rule $(\wedge_2)$ *need to be rewritten in the substitution so that their contexts are disjoint; the original context is then recovered by a suitable sequence of* (m) *rules.*

3   Normalization bound

In computing the normalization bound, we take inspiration from SLL [9] and [6], but taking into account the mismatch between proof simplification and β-reduction. So here we do not use the derivation as reduction machine, but rather as a tool for computing the number of reduction steps.

To do so, we first introduce a few necessary definitions of measures.

Definition 3 (Measures)**.**

i) The **size** $|\Pi|$ of a proof Π is defined inductively as follows:
   - if the last rule of Π is the axiom rule, then $|\Pi|=1$;
   - if the last rule of Π is a rule with n premises Π_i, then $|\Pi|=(\sum_{i=1}^{n}|\Pi_i|)+1$.

ii) The **size** $|M|$ of a term M is defined inductively as follows:

$$|x| = 1; \qquad |\lambda x.M| = |M| + 1; \qquad |MN| = |M| + |N| + 1.$$

iii) The **rank** of a multiplexor

$$\frac{\Gamma, x_1 : \sigma_1, ..., x_n : \sigma_n \vdash M : \tau}{\Gamma, x : \sigma_1 \wedge ... \wedge \sigma_n \vdash M[x_i \mapsto x]_{i=1}^n : \tau} \ (m)$$

is the number $k \le n$ of variables x_i such that $x_i \in FV(M)$. Let r be the maximum rank of a rule (m) in Π. The rank $\mathrm{rk}(\Pi)$ of Π is the maximum between 1 and r.

iv) The **degree** of a proof Π, denoted by $d(\Pi)$, is the maximal nesting of applications of the $(\wedge_n)$ rule in Π, i.e. the maximal number of applications of the $(\wedge_n)$ rule in a path connecting the conclusion and one axiom of Π.

v) The **weight** $W(\Pi, r)$ of Π with respect to r is defined inductively as follows:

- if (Ax) is the last applied rule, then $W(\Pi, r) = 1$;
- if $(\rightarrow I)$ is the last applied rule and Σ is the premise of the rule, then $W(\Pi, r) = W(\Sigma, r) + 1$;
- if $(\rightarrow E)$ is the last applied rule and Σ_1, Σ_2 are the premises of the rule, then $W(\Pi, r) = W(\Sigma_1, r) + W(\Sigma_2, r) + 1$;
- if $(\wedge_n)$ is the last applied rule and $\Sigma_1, ..., \Sigma_n$ are the premises of the rule, then $W(\Pi, r) = r \cdot \max_{i=1}^n W(\Sigma_i, r)$;
- if either (w) or (m) is the last applied rule and Σ is the unique premise derivation, then $W(\Pi, r) = W(\Sigma, r)$.

The previously introduced measures are related to each other as shown explicitly by the following lemma:

Lemma 2. *Let* $\Pi \triangleright \Gamma \vdash M : \sigma$. *Then:*

i) $\mathrm{rk}(\Pi) \le |M| \le |\Pi|$.

ii) $W(\Pi, r) \le r^{d(\Pi)} \cdot W(\Pi, 1)$.

iii) $W(\Pi, 1) = |M|$.

Proof. The proofs are given by induction on the shape of Π.

i) The most interesting case is for Π of the shape

$$\frac{\Sigma \triangleright \Gamma, x_1 : \tau_1, ..., x_n : \tau_n \vdash M : \sigma}{\Pi \triangleright \Gamma, x : \tau_1 \wedge ... \wedge \tau_n \vdash M[x/x_1, ... , x/x_n] : \sigma} \ (m)$$

By inductive hypothesis, $\mathrm{rk}(\Sigma) \le |M| \le |\Sigma|$.

Let $k \le n$ be the number of variables in $\{x_1, ..., x_n\} \cap FV(M)$. By Definition 3, $\mathrm{rk}(\Pi) = \max\{\mathrm{rk}(\Sigma), k\}$, $k \le |M[x/x_1, ... , x/x_n]| = |M|$ and $|\Pi| = |\Sigma| + 1$, therefore

- if $\max\{\mathrm{rk}(\Sigma), k\} = \mathrm{rk}(\Sigma)$, then $\mathrm{rk}(\Pi) = \mathrm{rk}(\Sigma) \le |M[x/x_1, ... , x/x_n]| \le |\Sigma| + 1$
- if $\max\{\mathrm{rk}(\Sigma), k\} = k$, then $\mathrm{rk}(\Pi) = k \le |M[x/x_1, ... , x/x_n]| \le |\Sigma| + 1$

and $\mathrm{rk}(\Pi) \le |M[x/x_1, ... , x/x_n]| \le |\Pi|$.

ii) The most interesting case is for Π of the shape

$$\frac{\Sigma_1 \triangleright \Gamma_1 \vdash \mathtt{M} : \sigma_1 \quad ... \quad \Sigma_n \triangleright \Gamma_n \vdash \mathtt{M} : \sigma_n}{\Pi \triangleright \bigwedge_{i=1}^n \Gamma_i \vdash \mathtt{M} : \sigma_1 \wedge ... \wedge \sigma_n} \ (\wedge_n)$$

By inductive hypothesis, $\mathtt{W}(\Sigma_i, r) \leq r^{\mathtt{d}(\Sigma_i)} \cdot \mathtt{W}(\Sigma_i, 1)$ for $1 \leq i \leq n$, and in particular $\max_{i=1}^n \mathtt{W}(\Sigma_i, r) \leq r^{\max_{i=1}^n \mathtt{d}(\Sigma_i)} \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, 1)$. Moreover, by Definition 3, $\mathtt{W}(\Pi, r) = r \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, r)$, $\mathtt{d}(\Pi) = \max_{i=1}^n \mathtt{d}(\Sigma_i) + 1$ and $\mathtt{W}(\Pi, 1) = 1 \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, 1)$, therefore

$$r \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, r) \leq r \cdot r^{\max_{i=1}^n \mathtt{d}(\Sigma_i)} \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, 1) = r^{\max_{i=1}^n \mathtt{d}(\Sigma_i)+1} \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, 1)$$

and $\mathtt{W}(\Pi, r) \leq r^{\mathtt{d}(\Pi)} \cdot \mathtt{W}(\Pi, 1)$.

iii) We only show the case where Π is of the shape

$$\frac{\Sigma_1 \triangleright \Gamma_1 \vdash \mathtt{M} : \sigma_1 \quad ... \quad \Sigma_n \triangleright \Gamma_n \vdash \mathtt{M} : \sigma_n}{\Pi \triangleright \bigwedge_{i=1}^n \Gamma_i \vdash \mathtt{M} : \sigma_1 \wedge ... \wedge \sigma_n} \ (\wedge_n)$$

By inductive hypothesis $\mathtt{W}(\Sigma_i, 1) = |\mathtt{M}|$ for $1 \leq i \leq n$. Moreover, by Definition 3, $\mathtt{W}(\Pi, 1) = 1 \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, 1)$, therefore $\max_{i=1}^n \mathtt{W}(\Sigma_i, 1) = |\mathtt{M}|$, and $\mathtt{W}(\Pi, 1) = |\mathtt{M}|$.

$\square$

So we can give the following weighted version of Lemma 1:

Lemma 3 (Weighted substitution). *Let $\Pi \triangleright \Gamma, \mathtt{x} : \sigma \vdash \mathtt{M} : \tau$ and $\Sigma \triangleright \Delta \vdash \mathtt{N} : \sigma$, with $\Gamma \# \Delta$ and $\mathtt{x} \notin dom(\Delta)$. Then $S(\Sigma, \Pi) \triangleright \Gamma, \Delta \vdash \mathtt{M}[\mathtt{N}/\mathtt{x}] : \tau$ and $\mathtt{W}(S(\Sigma, \Pi), r) \leq \mathtt{W}(\Pi, r) + \mathtt{W}(\Sigma, r)$, for every $r \geq \max\{\mathtt{rk}(\Pi), \mathtt{rk}(\Sigma)\}$.*

Proof. By induction on the shape of Π: we will refer to the proof for Lemma 1 and show that the condition on the measure holds. Again, most cases are trivial so we will only show the most meaningful ones, namely (w), $(\wedge)$ and (m).

If the last applied rule is (w), since $\mathtt{W}(S(\Sigma, \Pi), r) = \mathtt{W}(\Pi', r) = \mathtt{W}(\Pi, r)$, the inequality $\mathtt{W}(S(\Sigma, \Pi), r) \leq \mathtt{W}(\Sigma, r) + \mathtt{W}(\Pi, r)$ is satisfied: in fact, the sequence of rules needed to recover Δ is a sequence of (w) and (m) rules, which do not contribute to the weigth.

If the last applied rule is $(\wedge_n)$, with $n > 1$, then, by the proof for Lemma 1, $\mathtt{W}(\Pi, r) = r \cdot \max_{i=1}^n \mathtt{W}(\Pi_i, r)$ and $\mathtt{W}(\Sigma, r) = r \cdot \max_{i=1}^n \mathtt{W}(\sigma_i, r)$. By inductive hypothesis $S(\Sigma_i, \Pi_i) \triangleright \Gamma_i, \Delta_i \vdash \mathtt{M}[\mathtt{N}'/\mathtt{x}] : \tau_i$ and $\mathtt{W}(S(\Sigma_i, \Pi_i), r) \leq \mathtt{W}(\Sigma_i, r) + \mathtt{W}(\Pi_i, r)$ for $1 \leq i \leq n$. Since $\mathtt{W}(S(\Sigma, \Pi), r) = r \cdot \max_{i=1}^n \mathtt{W}(S(\Sigma_i, \Pi_i), r) \leq r \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, r) + r \cdot \max_{i=1}^n \mathtt{W}(\Pi_i, r) = \mathtt{W}(\Sigma, r) + \mathtt{W}(\Pi, r)$, the inequality is satisfied.

Let the last applied rule be (m), and let k be its rank. From the proof for Lemma 1, we can assume $\{\mathtt{x}_1, ... , \mathtt{x}_k\} = \mathtt{FV}(\mathtt{M}) \cap \{\mathtt{x}_1, ... , \mathtt{x}_n\}$, and moreover $\mathtt{W}(\Sigma, r) = r \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i, r) = r \cdot \max_{i=1}^n \mathtt{W}(\Sigma_i', r)$.

Let $r \geq \max\{\mathtt{rk}(\Pi), \mathtt{rk}(\Sigma)\} \geq \max\{\mathtt{rk}(\Sigma_1), ... , \mathtt{rk}(\Sigma_n), \mathtt{rk}(\Pi'), k\}$. By induction $\mathtt{W}(S(\Sigma_1', \Pi'), r) \leq \mathtt{W}(\Sigma_1', r) + \mathtt{W}(\Pi', r)$; then $\mathtt{W}(S(\Sigma_2', S(\Sigma_1', \Pi'))_2, r) \leq \mathtt{W}(\Sigma_2', r) + \mathtt{W}(S(\Sigma_1', \Pi'), r)$, and so on. By applying substitutions from 1 through k we get $\mathtt{W}(S(\Sigma_k', ...S(\Sigma_1', \Pi')), r) \leq \mathtt{W}(\Sigma_k', r) + \mathtt{W}(S(\Sigma_{k-1}', ...S(\Sigma_1', \Pi')), r)$. By applying sequences of rules ρ and δ to $S(\Sigma_k', ...S(\Sigma_1', \Pi'))$, and then a suitable sequence of (w) and (m) rules to recover the context Δ, we get the desired proof. Notice that both ρ and δ do not contribute to the weight. By induction, $\mathtt{W}(S(\Sigma_k', ...S(\Sigma_1', \Pi')), r) \leq \mathtt{W}(\Sigma_1', r) + ... + \mathtt{W}(\Sigma_k', r) + \mathtt{W}(\Pi', r) \leq k \cdot \max_{i=1}^k \mathtt{W}(\Sigma_i', r) + \mathtt{W}(\Pi', r)$. Since $\mathtt{W}(S(\Sigma, \Pi), r) = \mathtt{W}(S(\Sigma_k', ...S(\Sigma_1', \Pi')), r)$ and $\mathtt{W}(\Pi, r) = \mathtt{W}(\Pi', r)$,

$$\mathtt{W}(S(\Sigma, \Pi), r) = \mathtt{W}(S(\Sigma_k', ...S(\Sigma_1', \Pi')), r) \leq k \cdot \max_{i=1}^k \mathtt{W}(\Sigma_i', r) + \mathtt{W}(\Pi', r) \leq \mathtt{W}(\Sigma, r) + \mathtt{W}(\Pi, r)$$

and the inequality is satisfied.

$\square$

Using the previous property, we can prove that the weight of a proof decreases while reducing the subject.

Lemma 4. $\Pi \triangleright \Gamma \vdash M : \sigma$ *and* $M \xrightarrow{\beta} M'$ *imply there is a derivation* $\Pi' \triangleright \Gamma \vdash M' : \sigma$, *such that for every* $r \geq \mathtt{rk}(\Pi)$, $\mathtt{W}(\Pi', r) < \mathtt{W}(\Pi, r)$.

Proof. As in the proof of Property 2, we consider just the base case, when $M = (\lambda \mathtt{x}.\mathtt{Q})\mathtt{N}$. Then the most difficult case is when Π ends by a non empty intersection tree. We will use the same terminology as in Property 2. Remember that Π' is obtained from Π by replacing every subproof Π_i:

$$\frac{\dfrac{\dfrac{\Sigma_i' \triangleright \Gamma_i', \mathtt{x} : \sigma_i \vdash \mathtt{Q}_i' : \mathtt{A}_i}{\Gamma_i' \vdash \lambda \mathtt{x}.\mathtt{Q}_i' : \sigma_i \to \mathtt{A}_i} (\to I)}{\Gamma_i \vdash \lambda \mathtt{x}.\mathtt{Q}_i : \sigma_i \to \mathtt{A}} (\delta_i) \qquad \Sigma_i'' \triangleright \Delta_i \vdash \mathtt{N}_i : \sigma_i}{\Gamma_i, \Delta_i \vdash (\lambda \mathtt{x}.\mathtt{Q}_i)\mathtt{N}_i : \mathtt{A}_i} (\to E)$$

by $S(\Sigma_i'', \Sigma_i')$, and leaving the intersection tree connecting all these subproofs unchanged.

By Lemma 3, for every $r \geq \mathtt{rk}(\Pi)$, $\mathtt{W}(S(\Sigma_i'', \Sigma_i'), r, \leq)\mathtt{W}((,\Sigma)_i'', r) + \mathtt{W}(\Sigma_i', r,)$.

Since $\mathtt{W}(\Pi_i, r, =)\mathtt{W}(\Sigma_i'', r, +)\mathtt{W}(\Sigma_i', r, +)1$, the proof is given. $\qquad\square$

We can now prove that both the number of normalization steps and the size of the normal form are bounded by a function of the size of the term.

Theorem 1 (Measure of reduction)**.** *Let* $\Pi \triangleright \Gamma \vdash M : \sigma$, *and let* M β-*reduce to* M' *in n steps. Then:*

i) $n < |M|^{\mathtt{d}(\Pi)+1}$;

ii) $|M'| < |M|^{\mathtt{d}(\Pi)+1}$.

Proof. Let $M \xrightarrow{\beta} M_1 \xrightarrow{\beta} ... \xrightarrow{\beta} M_n = M'$. Then, by repeatedly applying Lemma 4, there is $\Pi_i \triangleright M_i : \sigma$ such that, for all $r \geq \mathtt{rk}(\Pi)$, $\mathtt{W}(\Pi_{i+1}, r) < \mathtt{W}(\Pi_i, r)$, for all $1 \leq i \leq n-1$. Since the rank of a proof never increases when reducing the subject, if $r = \mathtt{rk}(\Pi)$, then $\mathtt{W}(\Pi_{i+1}, r) < \mathtt{W}(\Pi_i, r)$. Then the proof of the first point follows.

By Lemma 2, for all i, $|M_i| = \mathtt{W}(\Pi_i, 1)$. Then the proof follows again from Lemma 4. $\qquad\square$

So the exponent of the function is, in general, dependent on the term; for this reason, the bound on the normalization procedure can easily become exponential. Nevertheless, the proof given above is independend on a given reduction strategy.

Remark 1. *One of the referees of this paper asked why we chosed the intersection as n-ary instead than binary connective, since by the lack of associativity the typability power of the system is the same in both cases, and binary intersection is more "standard". The answer is simple. We are interested not only in typability, but in using derivations for measuring the complexity of the reduction. Consider the term* $M = (\lambda xy.y\underbrace{xx...x}_{n})(II)$, *where* $I = \lambda x.x$. $|M| = 2n + 6$. *The minimal depth of a derivation in* STI *typing* M *has depth* 1, *and rank n, so the resulting bound for the number of* β-*reduction steps is* $(2n+6)$, *while the effective number of reductions is* $2n + 1$. *In case of binary intersection, so modifying* STI *in order to have only a multiplexor of rank* 2, *the minimal derivation has depth* $n-1$, *and the resulting bound is* $(2n+6)^n$, *so becomes exponential.*

References

[1] Erika De Benedetti (2011): *Polynomial Lambda-Calculus via Intersection Types*. Master's thesis, Università degli Studi di Torino.

[2] Alexis Bernadet & Stéphane Lengrand (2011): *Complexity of strongly normalising λ-terms via non-idempotent intersection types*. In Martin Hofmann, editor: *Proceedings of (FOSSACS'11)*, LNCS 6604, Springer, doi:`10.1007/978-3-642-19805-2_7`.

[3] Antonio Bucciarelli, Thomas Ehrhard & Giulio Manzonetto (2007): *Not Enough Points Is Enough*. In Jacques Duparc & Thomas A. Henzinger, editors: *Proceedings of CSL 2007*, LNCS 4646, Springer, pp. 298–312, doi:`10.1007/978-3-540-74915-8_24`.

[4] Daniel de Carvalho (2009): *Execution Time of lambda-Terms via Denotational Semantics and Intersection Types*. *MSCS* to appear. Available at `http://arxiv.org/abs/0905.4251`.

[5] Pietro Di Gianantonio, Furio Honsell & Marina Lenisa (2008): *A type assignment system for game semantics*. *Theoretical Computer Science* 398, pp. 150–169, doi:`10.1016/j.tcs.2008.01.023`.

[6] Marco Gaboardi & Simona Ronchi Della Rocca (2007): *A Soft Type Assignment System for λ-Calculus*. In Jacques Duparc & Thomas A. Henzinger, editors: *Proceedings of CSL'07*, LNCS 4646, Springer, pp. 253–267, doi:`10.1007/978-3-540-74915-8_21`.

[7] Assaf J. Kfoury (2000): *A Linearization of the Lambda-Calculus and Consequences*. *J. Logic Comp.* 10(3), pp. 411–436, doi:`10.1093/logcom/10.3.411`.

[8] Assaf J. Kfoury & J. B. Wells (2004): *Principality and type inference for intersection types using expansion variables*. *Theoretical Computer Science* 311(1-3), pp. 1–70, doi:`10.1016/j.tcs.2003.10.032`.

[9] Y. Lafont (2004): *Soft linear logic and polynomial time*. *Theoretical Computer Science* 318(1-2), pp. 163–180, doi:`10.1016/j.tcs.2003.10.018`.

[10] Harry Mairson & Peter M. Neergaard (2004): *Types, potency, and idempotency: why nonlinearity and amnesia make a type system work*. In: *proceedings of ICFP 2004*, pp. 138–149, doi:`10.1145/1016850.1016871`.

[11] Peter Møller Neergaard (2005): *Theoretical Pearls: A bargain for intersection types: a simple strong normalization proof*. *J. Funct. Program.* 15(5), pp. 669–677. Available at `http://dx.doi.org/10.1017/S0956796805005587`.

[12] Michele Pagani & Simona Ronchi Della Rocca (2010): *Solvability in Resource Lambda-Calculus*. In C.-H. Luke Ong, editor: *Proceedings of FOSSACS 2010*, LNCS 6014, Springer, pp. 358–373, doi:`10.1007/978-3-642-12032-9_25`.

[13] Michele Pagani & Simona Ronchi Della Rocca (2010): *Linearity, Non-determinism and Solvability*. *Fundamenta Informaticae* 103, pp. 358–373.

[14] Luca Paolini, Mauro Piccolo & Simona Ronchi Della Rocca (2012): *Logical Relational Lambda-Models*. Available at `http://www.di.unito.it/%7Elambda/biblio/entry-paolini12draft.html`. DRAFT.

[15] K. Terui (2006): *Intersection Types for Computational Complexity*. Available at `citeseer.ist.psu.edu/294754.html`. Slides.