Bounding Skeletons, Locally Scoped Terms and Exact Bounds for Linear Head Reduction

Pierre Clairambault

Computer Laboratory, University of Cambridge
pierre.clairambault@cl.cam.ac.uk

Abstract. Bounding skeletons were recently introduced as a tool to study the length of interactions in Hyland/Ong game semantics. In this paper, we investigate the precise connection between them and execution of typed λ-terms. Our analysis sheds light on a new condition on λ-terms, called *local scope*. We show that the reduction of locally scoped terms matches closely that of bounding skeletons. Exploiting this connection, we give upper bound to the length of linear head reduction for simply-typed locally scoped terms. General terms lose this connection to bounding skeletons. To compensate for that, we show that λ-lifting allows us to transform any λ-term into a locally scoped one. We deduce from that an upper bound to the length of linear head reduction for arbitrary simply-typed λ-terms. In both cases, we prove the asymptotical optimality of the upper bounds by providing matching lower bounds.

1 Introduction

In the last two decades there has been a growing interest in the study of *quantitative* or *intensional* aspects of higher-order programs; in particular, the study of their *complexity* has generated a lot of effort. In the context of the λ-calculus, the first result that comes to mind is the work by Schwichtenberg [14], later improved by Beckmann [2], establishing upper bound to the length of β-reduction sequences for simply-typed λ-calculus. In the somewhat related line of work of *implicit complexity*, type systems have been developed to characterize extensionally certain classes of functions, such as polynomial [10] or elementary [8] time. Such systems rely on a soundness theorem establishing that well-typed terms normalize in a certain restricted time, which is itself established using syntactic methods that are specific to the system being studied. This calls for the development of syntax-independent tools to study precisely the execution time of higher-order programs. The present paper is a step towards that goal.

In [4], in the context of Hyland-Ong game semantics, we showed that provided some size information on strategies we could give a bound to the length of their interactions. This was done by annotating each step of an interaction sequence with a finite tree of natural numbers – hereby called a *bounding skeleton*[1] – and showing that progress in the interaction amounts to a simple reduction on the

[1] They were called *agents* in [4].

M. Hasegawa (Ed.): TLCA 2013, LNCS 7941, pp. 109–124, 2013.

bounding skeleton. We gave bounds to the length of this reduction, hence bounding with it the length of the interaction sequence. The strength of this approach is that the games model is syntax-independent: in the variant considered in [4], it accommodates the simply-typed λ-calculus possibly with computational effects such as non-determinism, control, or ground type references. Its key weakness however, is that the direct connection between game-theoretic interaction and execution has only been made explicit [6] for pure simply-typed λ-terms of the form $M\ N_1\ \ldots\ N_p$, where M and the N_is are η-long Böhm trees – we will call such terms *game situations*. Although terms can be transformed into game situations (as briefly described in [4]), the transformation is very inefficient and yields bounds that are sub-optimal and not very informative. For bounding skeletons to be a useful tool in complexity analysis, it is crucial to relate them directly to the execution of programs, without the detour by game semantics. Such a connection is non-trivial, as the dynamics of reduction in all generality is much more complicated than for game situations.

In this paper, we develop such a connection. This is done by introducing a new structural condition on terms, *local scope*, that ensures that information only flows locally through redexes, and not remotely through variables shared by distant subterms. We show that the reduction of η-long, locally scoped terms can be directly simulated within bounding skeletons. Using this property and (a small optimization of) our results in [4] on bounding skeletons, we deduce exact bounds to the execution time of locally scoped terms. We also show that the operation of λ-*lifting* [12] transforms arbitrary terms into locally scoped ones, and exploit this transformation to give exact bounds for the execution time of arbitrary simply-typed λ-terms.

Related works. There are multiple approaches to the complexity analysis of higher-order programs, but they seem to separate into two major families. On the one hand, Beckmann [2], extending earlier work by Schwichtenberg [14], gave exact bounds to the maximal length of β-reduction on simply-typed λ-terms. His analysis uses very basic information on the terms (their length, or height, and order), but gives bounds that are in general very rough. On the other hand other groups, including Dal Lago and Laurent [13], De Carvalho [9], or Bernardet and Lengrand [3], use semantic structures (respectively, game semantics, relational semantics, or non-idempotent intersection types) to capture abstractly the precise complexity of particular terms. Their bounds are much more precise on particular terms, but require information on the terms whose extraction is in general as long to obtain as actual execution. The present work belongs to the first family. However, unlike Beckmann and Schwichtenberg the reduction we consider is *linear head reduction*, which is the notion of execution implemented by abstract machines [7] and is therefore much closer to the actual execution of functional programming languages.

Outline. In Section 2, we start by introducing linear head reduction along with bounding skeletons, and recall the main result of [4]. In Section 3, we introduce local scope, show our simulation result and deduce exact bounds on the length of linear head reduction on locally scoped terms. Finally in Section 4, we show

how to use λ-lifting to transform arbitrary terms into locally scoped ones, and deduce exact bounds for linear head reduction on general terms.

2 Preliminaries

In this section, we start by recalling some of the background of this research. The natural starting point is *linear head reduction* [7], which can be seen as a direct implementation on λ-terms of the notion of execution performed by abstract machines. We will then turn to the presentation of *bounding skeletons*: we will recall the results of [4] on the length of their reductions, along with a small improvement.

2.1 Linear Head Reduction

We work here with the simply-typed λ-calculus *à la Church*, *i.e.* the variables are explicitly annotated with types (although we often omit the annotations for the sake of readability). Types are built from a unique atom o and the arrow constructor $\to$. We suppose that for every type A, there is a constant $*_A : A$ of type A. We will often omit the index and write $*$. As usual, we write $\mathrm{fv}(M)$ for the set of *free variables* of a term M. The typing relation $\Gamma \vdash M : A$ is defined by the usual deduction rules for simply-typed λ-calculus. All the terms considered in this paper are supposed well-typed. Note that our choices – only one atom, each type is inhabited – merely make the presentation simpler and are not strictly required for our results to hold.

This work focuses strongly on *linear substitution*, for which only one variable occurrence is substituted at a time. In this situation, it is convenient to have a distinguished notation for particular *occurrences* of variables. We will use the notations $x_0, x_1, \ldots$ to denote particular occurrences of the same variable x in a term M. When in need of additional variable identifiers, we will use $x^1, x^2, \ldots$. Sometimes, we will still denote occurrences of x by just x when their index is not relevant. Although it is not the focus of this development, we will occasionally also refer to β-reduction: it is the standard rewriting rule on λ-terms, defined by $(\lambda x.M)\ N \to_\beta M[N/x]$, where $M[N/x]$ is the substitution of *all* occurrences of the variable x by N, applied in any position within M. We write $\equiv_\beta$ for the corresponding equivalence relation. If x_0 is a specific occurrence of x, we will use $M[N/x_0]$ for the substitution of x_0 by N, leaving all other occurrences of x unchanged. We assume Barendregt's convention and consider all terms up to α-equivalence (so, substitution involves renaming of bound variables).

Intuitively, linear head reduction proceeds as follows. We first locate the head variable occurrence, *i.e.* the leftmost variable occurrence in the term M. Then we locate the abstraction, if any, that binds this variable. Then we locate (again if it exists) the subterm N of M in argument position for that abstraction, and we substitute the head occurrence by N. We touch neither the other occurrences of x nor the redex. It is worth noting that locating the argument subterm can be delicate, as it is not necessarily part of a β-redex. For instance

in $(\lambda y^A.(\lambda x^B.x_0 M))N_1 N_2$, we want to replace x_0 by N_2, even though N_2 is not directly applied to $\lambda x^B.x_0 M$. Therefore, the notion of redex will be generalized.

Note that a term is necessarily of the form $* \; M_1 \; \ldots \; M_n, x_0 \; M_1 \; \ldots \; M_n, \lambda x.M$ or $(\lambda x.M) \; M_1 \; \ldots \; M_n$. That will be used quite extensively to define and reason on linear head reduction. The **length** of a term M is the number of characters in M, *i.e.* $\mathrm{l}(*) = 1, \mathrm{l}(x_0) = 1, \mathrm{l}(\lambda x.M) = \mathrm{l}(M) + 1, \mathrm{l}(M_1 \; M_2) = \mathrm{l}(M_1) + \mathrm{l}(M_2)$. Its **height** is $\mathrm{h}(*) = 0, \mathrm{h}(x_0) = 1, \mathrm{h}(\lambda x.M) = \mathrm{h}(M), \mathrm{h}(M_1 \; M_2) = \max(\mathrm{h}(M_1), \mathrm{h}(M_2) + 1)$.

Definition 1. *Given a term M, we define its set of **prime redexes**. They are written as pairs $(\lambda x, N)$ where N is a subterm of M, and λx is used to denote the (if it exists, necessarily unique by Barendregt's convention) subterm of M of the form $\lambda x.N'$. We define the prime redexes of M by induction on its length, distinguishing several cases depending on the form of M.*

- *If $M = * \; M_1 \; \ldots \; M_n$, then M has no prime redex.*
- *If $M = x_0 \; M_1 \; \ldots \; M_n$, then M has no prime redex.*
- *If $M = \lambda x.M'$, then M has the prime redexes of M'.*
- *If $M = (\lambda x.M') \; M_1 \; \ldots \; M_n$, then the prime redexes of M are $(\lambda x, M_1)$ plus those of $M' \; M_2 \; \ldots \; M_n$.*

The **head occurrence** of a term M is the leftmost occurrence of a variable or constant in M. If $(\lambda x, N)$ is a prime redex of M where the head occurrence of M is an occurrence x_0 of the variable x, then the **linear head reduct** of M is $M' = M[N/x_0]$. We write $M \to_{\mathrm{lhr}} M'$.

Example 1. As an example, we give the linear head reduction sequence of the term $(\lambda f.\lambda x.f \; (f \; x)) \; (\lambda y.y)$.

$$\begin{aligned}(\lambda f.\lambda x.f \; (f \; x)) \; (\lambda y.y) &\to_{\mathrm{lhr}} (\lambda f.\lambda x.(\lambda z.z) \; (f \; x)) \; (\lambda y.y) \\ &\to_{\mathrm{lhr}} (\lambda f.\lambda x.(\lambda z.f \; x) \; (f \; x)) \; (\lambda y.y) \\ &\to_{\mathrm{lhr}} (\lambda f.\lambda x.(\lambda z.(\lambda u.u) \; x) \; (f \; x)) \; (\lambda y.y) \\ &\to_{\mathrm{lhr}} (\lambda f.\lambda x.(\lambda z.(\lambda u.x) \; x) \; (f \; x)) \; (\lambda y.y)\end{aligned}$$

At this point the reduction stops since the head occurrence is an occurrence of x, and the corresponding abstraction subterm is not part of a prime redex.

We will abbreviate linear head reduction by lhr. It is straightforward to see that lhr is compatible with β-reduction, in the sense that if $M \to_{\mathrm{lhr}} M'$ we have $M \equiv_\beta M'$. Just as for β-reduction, lhr always terminates on well-typed terms, let us denote by $\mathcal{N}(M)$ the length of the reduction sequence of M. Since redexes for lhr are not necessarily β-redexes, it will be necessary to consider the following generalization of redexes:

Definition 2 (Generalized redex). *The **generalized redexes** of a term M are the prime redexes of all subterms of M. In particular, all prime redexes are generalized redexes.*

Example 2. Consider the following λ-term:

$$M = (\lambda x.x)\ ((\lambda y.(\lambda z.u))\ v\ w)$$

The only prime redex of M is $(\lambda x, (\lambda y.(\lambda z.u))\ v\ w)$, and it is therefore also a generalized redex. The two other generalized redexes are $(\lambda y, v)$, which is also a β-redex, and $(\lambda z, w)$, which is not.

2.2 Bounding Skeletons

This section focuses on a pivotal notion of this paper, that of a bounding skeleton. Intuitively, it is what is left of a term when all precise dynamic information is forgotten, and only the structural size information necessary to study termination is retained. Formally, a **bounding skeleton** is a finite tree whose nodes and edges are labeled by natural numbers. We write:

$$n[\{d_1\}a_1, \ldots, \{d_p\}a_p] = \quad \begin{array}{ccccc} & & n & & \\ & {\scriptstyle d_1}\swarrow & & \searrow{\scriptstyle d_p} & \\ a_1 & & \ldots & & a_p \end{array}$$

This notion was introduced in [4] where it was extracted from *game semantics*, and more precisely from the notion of *pointing sequence* central to Hyland-Ong games [11], but also appearing crucially in the earlier work of Coquand [5]. Bounding skeletons arise as measures of positions in pointing sequences, progress in the sequence corresponding to reduction of the skeleton. By the operational content of game semantics [6], bounding skeletons can also be seen as measures of terms obtained by lhr from a term of a particular form called a *game situation*. A **game situation** is a term of the form $M\ N_1\ \ldots\ N_n$, where $M : A_1 \to \ldots \to A_n \to o$ and $N_i : A_i$ are closed η-long Böhm trees – the terminology is motivated by the strong geometric correspondence between *η-long Böhm trees* and the *innocent strategies* of [11]. We know by the result of Danos, Herbelin and Regnier [6] that the lhr sequence of $M\ N_1\ \ldots\ N_n$ is in step-by-step correspondence with the game-theoretic interaction between the corresponding strategies $[\![M]\!]$ and $[\![N_i]\!]$. To illustrate how bounding skeletons arise from lhr of game situations, let us suppose for simplicity that $M : (A \to o) \to o$ and that M has the form $\lambda x.x\ M'$ with M' η-long Böhm tree of type A, possibly including x as a free variable. Then we have the lhr step:

$$(\lambda x.x\ M')\ N \to_{\mathrm{lhr}} (\lambda x.N\ M')\ N$$

Therefore, a situation with a closed Böhm tree M applied to a closed Böhm tree N is reduced to a closed Böhm tree N applied to an *open* Böhm tree M', along with an environment associating x to the closed Böhm tree N. In other words:

$$M \star N \to N \star M'^{\{x \mapsto N\}}$$

where the notation used will remain informal, but should be clear nonetheless. This can be represented by the following operation on trees of terms:

$$\begin{array}{ccc} & & N \\ & & | \\ M & & M' \\ | & \to & | \\ N & & N \end{array}$$

Replacing the terms by some measure of size (which will be made precise later in the paper) and annotating the edges with a measure of their types, these trees give rise to bounding skeletons, and this reduction appears as an instance of the following non-deterministic rule, illustrated in Figure 1.

$$n[\{d_1\}a_1, \ldots, \{d_p\}a_p] \to_{\text{bs}} a_i \cdot_{d_i - 1} (n-1)[\{d_1\}a_1, \ldots, \{d_p\}a_p]$$

where $n, d_i \geq 1$ and $a \cdot_d b$ denotes the skeleton $n[\{d_1\}a_1, \ldots, \{d_p\}a_q, \{d\}b]$, for two skeletons $a = n[\{d_1\}a_1, \ldots, \{d_p\}a_q]$ and b.

This observation still applies with further reductions of $M\ N$, as we established in [4] through game semantics. Along with bounds to the length of reduction of bounding skeletons, this allowed us to bound the maximum length of lhr sequences starting from game situations. We also gave a bound for regular terms, relying on a very rough translation of arbitrary terms into game situations – as a result, this bound was far from optimal and not very informative. We aim in this paper to study the direct connection of bounding skeletons and syntax outside of game situations and independently on game semantics.

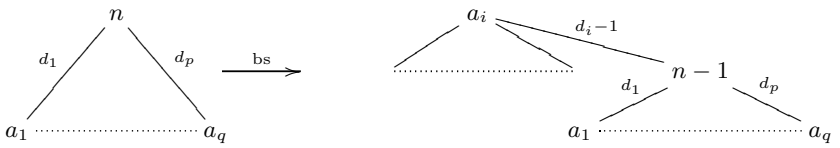

Fig. 1. Rewriting rule on skeletons

Remark 1. Note that reduction is set to only happen in root position, *i.e.* at the root of the tree. Generalizing it to apply deeper leads to pathological behavior. For instance deep reduction does not terminate on the variant without edge labels, whereas the standard (root) reduction does. It is not known whether deep reduction terminates in the presence of edge labels, or to which extent the relationship with syntactic reduction is preserved – the correspondence with game semantics is lost.

The main result of [4] is a bound on the length of reduction for bounding skeletons. The bound we state here is in fact a minor improvement of the result in [4], however the tools and methods to get it are the same. Therefore to save space, we omit the details of the optimization. As for terms, we write $\mathcal{N}(a)$ for the

norm of a bounding skeleton a, *i.e.* the length of its longest reduction sequence. We also write $\max(a)$ for the highest node label in a, $\mathrm{ord}(a)$ for the **order** of a, *i.e.* the highest edge label in a and $\mathrm{depth}(a)$ for the **depth** of a, *i.e.* the maximal depth of a node in a, the root being at depth 1. Here, log denotes the logarithm to base 2 and the tower of exponentials 2_n^p is defined by $2_0^p = p$ and $2_{n+1}^p = 2^{2_n^p}$.

Theorem 1 (Upper bound). *If* $\mathrm{ord}(a), \mathrm{depth}(a), \max(a) \geq 1$, *then*

$$\mathcal{N}(a) \leq 2_{\mathrm{ord}(a)-1}^{\mathrm{depth}(a)\log(\max(a)+1)}$$

Constructions. In defining the interpretation of terms as bounding skeletons, we will make use of the following constructions. If $(a_i)_{1\leq i\leq n}$ is a finite family of bounding skeletons, then writing $a_i = n_i[\{d_{i,1}\}b_{i,1}, \ldots, \{d_{i,p_i}\}b_{i,p_i}]$, we define:

$$\bigsqcup_{i=1}^{n} a_i = (\max_{1\leq i\leq n} n_i) \cdot [\{d_{i,j}\}b_{i,j} \mid 1 \leq i \leq n \ \&\ 1 \leq j \leq p_i]$$

$$\sum_{i=1}^{n} a_i = (\sum_{i=1}^{n} n_i) \cdot [\{d_{i,j}\}b_{i,j} \mid 1 \leq i \leq n \ \&\ 1 \leq j \leq p_i]$$

so, they either take the maximum or the sum of the roots, and simply append all the subtrees of the a^is. In the binary case, we write as usual $+$ for the sum. Finally, each natural number n can be seen as an atomic bounding skeleton $n[]$ without subtrees, still denoted by n. That should never cause any confusion.

Embedding. The norm of a bounding skeleton is unchanged by permutation of subtrees, or merging of identical subtrees, and is only increased by an increase of labels. If $a = n[\{d_1\}a_1, \ldots, \{d_p\}a_p]$ and $a' = n'[\{d'_1\}a'_1, \ldots, \{d'_{p'}\}a'_{p'}]$, we say that a **embeds** in a', written $a \hookrightarrow a'$, if $n \leq n'$ and for any $i \in \{1, \ldots, p\}$ there exists $j \in \{1, \ldots, p'\}$ such that $d_i \leq d'_j$ and $a_i \hookrightarrow a'_j$. Then we have:

Lemma 1 (Embedding lemma). *If* $a \hookrightarrow b$, *then* $\mathcal{N}(a) \leq \mathcal{N}(b)$.

We illustrate the reduction in Figure 2, where at each step we emphasize the subtree selected non-deterministically for the next reduction step. For conciseness, we also do not represent the subtrees under a node labeled 0, as they can play no further part in the reduction.

3 Locally Scoped Terms and Bounding Skeletons

This section explains the direct connection between linear head reduction and the reduction of bounding skeletons. We will first introduce *locally scoped terms*, for which this connection holds, then prove their simulation within bounding skeletons, and finally deduce bounds for the length of their reduction.

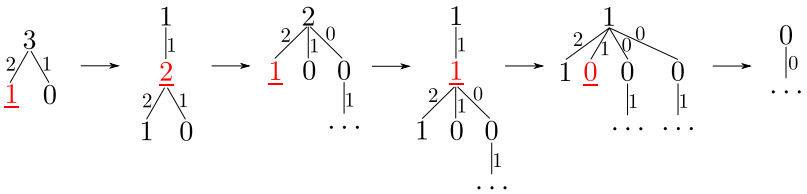

Fig. 2. Example reduction sequence on bounding skeletons

3.1 Locally Scoped Terms

Define inductively a **closure** as an open term M along with an **environment** σ, mapping free variables of M to closures of the same type. We say that a closure M^σ is **hereditarily normal** when M is β-normal and η-long, and when for any $x \in \mathrm{fv}(M)$, the closure $\sigma(x)$ is hereditarily normal. Hereditarily normal closures are very close to bounding skeletons: from a hereditarily normal closure M^σ one can obtain a bounding skeleton having the height of M as root, and the bounding skeletons corresponding to $\sigma(x)$ for $x \in \mathrm{fv}(x)$ as subtrees.

Although we will not make this formal in this paper, our simulation of lhr in bounding skeletons exploits that some terms can be represented as hereditarily normal closures. For instance, take the term:

$$K_1 = (\lambda x^{o\to o}.(\lambda y^{o\to o}.y)\ x)\ (\lambda z^o.z)$$

The term K_1 is faithfully represented by the hereditarily normal closure:

$$y^{y\mapsto x^{x\mapsto \lambda z.z}}$$

From that, we see that K_1 corresponds (ignoring edge labels) to the bounding skeleton $1[1[1]]$. Note in passing that K_1 reduces to $(\lambda x.(\lambda y.x)\ x)\ (\lambda z.z)$, which by the same idea as above corresponds to the hereditarily normal form $x^{x\mapsto\lambda z.z}$, and to the bounding skeleton $1[1]$ – which embeds in the bs-reduct $1[1,0[1[1]]]$ of $1[1[1]]$, so the lhr reduction of K_1 is accounted for in bounding skeletons.

Unfortunately, this connection does not always work. For instance, take:

$$K_2 = (\lambda x.x\ *)\ (\lambda z.(\lambda y.y)\ z)$$

When trying to represent K_2 as a hereditarily normal closure, we run into the issue that since z is not a closed subterm, there is not way to replace the redex $(\lambda y.y)\ z$ by an environment. Of course, in K_1, we also had a generalized redex $(\lambda y, x)$ where x is not closed. But in K_1, x was *active*, in the sense that we had a redex $(\lambda x, \lambda z.z)$, so we knew how to define the environment on x. On the other hand, z is *passive* in K_2: there is no generalized redex $(\lambda z, N)$. In summary, the issue with K_2 is that there is a generalized redex $(\lambda y, z)$ where z (obviously) contains a passive free variable z, and because of that K_2 cannot be directly represented as a hereditarily normal closure.

Definition 3. *A variable x in M is **active** iff it is a free variable or if there is a generalized redex $(\lambda x, N)$ in M. It is **passive** otherwise. A term M is **locally scoped** (abbreviated l.s.) if for any generalized redex $(\lambda x, N)$ in M all the free variables in N are active in M. Likewise, M is **strongly locally scoped** (abbreviated s.l.s.) if for any generalized redex $(\lambda x, N)$ in M, N is closed.*

So, the term K_1 above is locally scoped, but K_2 is not since there is a generalized redex $(\lambda y, z)$ with z passive. Neither of those are strongly locally scoped. Any β-normal term is strongly locally scoped, and so is any term obtained by applications of β-normal forms (such as λ-terms corresponding to terms of combinatory logic). Local scope will be sufficient to ensure that the interpretation to bounding skeletons is a simulation, but the correspondence between terms and bounding skeletons will be tighter for strongly locally scoped terms: for those, the tree structure of the bounding skeleton will match the tree structure of imbricated generalized redexes. Strongly locally scoped terms are not stable under lhr, so we need to develop the full connection on locally scoped terms instead.

Lemma 2. *If M is a locally scoped term* of ground type *and $M \to_{\mathrm{lhr}} M'$, then M' is locally scoped.*

3.2 Interpretation in Bounding Skeletons

Interpretation. The **level** of a type is defined by $\mathrm{lv}(o) = 0$ and $\mathrm{lv}(A \to B) = \max(\mathrm{lv}(A) + 1, \mathrm{lv}(B))$. Likewise, the **level** $\mathrm{lv}(M)$ of a term M is the level of its type. Finally, the **order** $\mathrm{ord}(M)$ of a term M is the maximal $\mathrm{lv}(N)$, for all subterms N of M. Within a term $\Gamma \vdash M : A$ such that $(x : B) \in \Gamma$, we write $\mathrm{lv}_M(x) = \mathrm{lv}(B)$. The term M will generally be obvious from the context, so we will just write $\mathrm{lv}(x)$.

Definition 4. *Let $\Gamma \vdash M : A$ be a term, with a **bs-environment** ρ, being defined as a partial function associating to each variable x of Γ on which it is defined a bounding skeleton $\rho(x)$. Then the bounding skeleton $[\![M]\!]_\rho$ is defined by induction on the length of M, as follows:*

$$
\begin{aligned}
[\![* \ M_1 \ \ldots \ M_n]\!]_\rho &= 0 \\
[\![x_0 \ M_1 \ \ldots \ M_n]\!]_\rho &= 1 + \bigsqcup_{i=1}^{n} [\![M_i]\!]_\rho && \text{if } \rho(x) \text{ undefined} \\
[\![x_0 \ M_1 \ \ldots \ M_n]\!]_\rho &= (1 + \bigsqcup_{i=1}^{n} [\![M_i]\!]_\rho) \cdot_{\mathrm{lv}(x)+1} \rho(x) && \text{if } \rho(x) \text{ defined} \\
[\![\lambda x.M]\!]_\rho &= [\![M]\!]_\rho \\
[\![(\lambda x.M) \ M_1 \ \ldots \ M_n]\!]_\rho &= [\![M \ M_2 \ \ldots \ M_n]\!]_{\rho \cup \{x \mapsto [\![M_1]\!]_\rho\}}
\end{aligned}
$$

We write $[\![M]\!]$ for $[\![M]\!]_\emptyset$.

Measures on terms and their preservation. To estimate lhr on s.l.s. terms, we need to define measures on terms that reflect the geometry of the corresponding bounding skeletons. So instead of the *height*, we have two alternative quantities.

The **depth** $\mathrm{depth}(M)$ of a term M is defined by induction on the length of M:

$$\begin{aligned}
\mathrm{depth}(*\ M_1\ \ldots\ M_n) &= 1\\
\mathrm{depth}(x_0\ M_1\ \ldots\ M_n) &= \max_{1\leq i\leq n} \mathrm{depth}(M_i)\\
\mathrm{depth}(\lambda x.M) &= \mathrm{depth}(M)\\
\mathrm{depth}((\lambda x.M)\ M_1\ \ldots\ M_n) &= \max(\mathrm{depth}(M\ M_2\ \ldots\ M_n), \mathrm{depth}(M_1)+1)
\end{aligned}$$

Likewise, the **local height** $\mathrm{lh}(M)$ of a term M is defined by:

$$\begin{aligned}
\mathrm{lh}(*\ M_1\ \ldots\ M_n) &= 0\\
\mathrm{lh}(x_0\ M_1\ \ldots\ M_n) &= 1+\max_{1\leq i\leq n} \mathrm{lh}(M_i)\\
\mathrm{lh}(\lambda x.M) &= \mathrm{lh}(M)\\
\mathrm{lh}((\lambda x.M)\ M_1\ \ldots\ M_n) &= \max(\mathrm{lh}(M\ M_2\ \ldots\ M_n), \mathrm{lh}(M_1))
\end{aligned}$$

Then, we have the following lemma:

Lemma 3. *If M is a strongly locally scoped term, then we have:*

$$\mathrm{depth}([\![M]\!]) \leq \mathrm{depth}(M) \qquad \max([\![M]\!]) \leq \mathrm{lh}(M) \qquad \mathrm{ord}([\![M]\!]) \leq \mathrm{ord}(M)$$

Simulation. In order to have our simulation result of linear head reduction into bounding skeletons, we need the additional requirement that the terms being interpreted are **η-long** – it is natural since our tools originate from game semantics, in which strategies are representations of η-long normal forms. As usual, η-expansion is the rule $M \to_\eta \lambda x^A.M\ x$, that applies when M has type $A \to B$ and $x \notin \mathrm{fv}(M)$. Non β-normal η-long terms are often defined as the terms on which any further η-expansion creates a new β-redex. Since we have generalized the notion of redex, we instead define them as the terms for which any η-expansion creates a new generalized redex. Then, η-long terms are stable under lhr. Moreover, we have:

Proposition 1 (Simulation). *Let $\Gamma \vdash M, M' : o$ be η-long locally scoped terms, and suppose $M \to_{\mathrm{lhr}} M'$. Then, there is a such that $[\![M]\!] \to_{\mathrm{bs}} a \hookleftarrow [\![M']\!]$.*

3.3 Bounds for Strongly Locally Scoped Terms

As a first application, we give exact bounds for the maximal length of lhr on strongly locally scoped terms. Formally, we will estimate the following quantity.

$$\mathrm{Lls}_n(h,d) = \max\{\mathcal{N}(M) \mid \mathrm{ord}(M) \leq n\ \&\ \mathrm{lh}(M) \leq h\ \&\ \mathrm{depth}(M) \leq d\ \&\ M \text{ s.l.s.}\}$$

To express our results, we will use some standard notations for comparing growth rates of functions. For functions $f, g : \mathbb{N} \to \mathbb{N}$, we write $f(n) = \Theta(g(n))$ when there exists reals $c_1, c_2 > 0$ and $N \in \mathbb{N}$ such that for all $n \geq N$, $c_1 g(n) \leq f(n) \leq c_2 g(n)$. This is generalized to functions of multiple variables $f, g : \mathbb{N}^p \to \mathbb{N}$

by setting that $f(n_1,\dots,n_p) = \Theta(g(n_1,\dots,n_p))$ iff there are $c_1, c_2 > 0$ and $N_i \in \mathbb{N}$ for all $i \in \{1,\dots,p\}$ such that for all $n_i \geq N_i$ we have $c_1 g(n_1,\dots,n_p) \leq f(n_1,\dots,n_p) \leq c_2 g(n_1,\dots,n_p)$. If $h : \mathbb{N} \to \mathbb{N}$ is another function, we write $f(n_1,\dots,n_p) = h(\Theta(g(n_1,\dots,n_p)))$ iff there is a function $\phi : \mathbb{N}^p \to \mathbb{N}$ such that $f(n_1,\dots,n_p) = h(\phi(n_1,\dots,n_p))$ and $\phi(n_1,\dots,n_p) = \Theta(g(n_1,\dots,n_p))$.

η-long form. Our simulation result only applies to η-long terms. Therefore, in order to obtain the upper bound we first need the following result:

Proposition 2. *If M is a term, then there is an η-long term M' such that:*

$$\mathrm{lh}(M') \leq \mathrm{lh}(M) + \mathrm{ord}(M) \qquad \mathrm{depth}(M') = \mathrm{depth}(M)$$
$$\mathrm{ord}(M') = \mathrm{ord}(M) \qquad \mathcal{N}(M') \geq \mathcal{N}(M)$$

Moreover if M was s.l.s., M' is still s.l.s..

The proof is mostly direct, but rather long and technical. We show first that η-expansion can only increase the norm and that it preserves strong local scope and the order of terms. Moreover, if η-expansion is restricted so that it does not create new generalized redexes, then it terminates on an η-long form. Besides, restricted η-expansion preserves depth and a variant $\mathrm{lh}'(M)$ of $\mathrm{lh}(M)$ such that $\mathrm{lh}(M) \leq \mathrm{lh}'(M) \leq \mathrm{lh}(M) + \mathrm{ord}(M)$, taking into account the potential size of the variables that are not yet expanded. Details are omitted.

Upper bound. If $\Gamma \vdash M : A_1 \to \ldots \to A_n \to o$ is a s.l.s. term, we first make it of ground type by forming $\Gamma \vdash M \ *_{A_1} \ \ldots \ *_{A_n} : o$ – its norm can only increase, the other quantities stay unchanged and the term is still s.l.s.. By Proposition 2, there is M' η-long, of ground type, and s.l.s. such that $\mathrm{lh}(M') \leq \mathrm{lh}(M)+\mathrm{ord}(M)$, $\mathrm{depth}(M') = \mathrm{depth}(M)$, $\mathrm{ord}(M') = \mathrm{ord}(M)$ and $\mathcal{N}(M') \geq \mathcal{N}(M)$. Along with Lemma 3, Proposition 1 and Theorem 1 this gives the following proposition.

Proposition 3. *Suppose M is a strongly locally scoped term of order at least one. Then, $\mathcal{N}(M) \leq 2^{\mathrm{depth}(M)\log(\mathrm{lh}(M)+\mathrm{ord}(M)+1)}_{\mathrm{ord}(M)-1}$.*

Lower bound. We now set to prove the optimality of this upper bound by exhibiting a family of terms whose reduction length asymptotically reaches it. The family we describe is a variant of one used by Beckmann in [2], constructed by iterated exponentiation of Church numerals. We define higher types for Church integers by setting $A_{-2} = o$ and $A_{n+1} = A_n \to A_n$. Then, writing $\underline{n}_p$ for the Church integer for n of type A_p, we define, for $n, k, p \geq 0$ and $M : A_p$:

$$[n]_p^0(M) = M \qquad [n]_p^{k+1}(M) = \underline{n}_{p+1}\ [n]_p^k(M)$$

One can immediately check that $[n]_p^k(M) : A_p$ and that for all $q \in \mathbb{N}$, $[n]_p^k(\underline{q}_p) \to_\beta^* \underline{q^{n^k}}_p$. Exploiting this construction we set, for $n, k, p \geq 0$:

$$S_{n,k,p} = [n]_p^k(\underline{2}_p)\ \underline{2}_{p-1}\ \ldots\ \underline{2}_0$$

For which it is immediate to check that for all $n, k, p \geq 0$ we have $S_{n,k,p} \to_\beta^* \underline{2_p^{2^{n^k}}}_0$. Moreover, by construction of $S_{n,k,p}$, for $n \geq 2$ and $p, k \geq 1$ we have $\mathrm{lh}(S_{n,k,p}) = n + 1$, $\mathrm{depth}(S_{n,k,p}) = k + 1$ and $\mathrm{ord}(S_{n,k,p}) = p + 3$, and $S_{n,k,p}$ is s.l.s.. To deduce a lower bound from this, we need to relate it to lhr using:

Lemma 4. *If* $M \to_\beta^* \underline{n}_0$*, then* $\mathcal{N}(M\ \mathrm{id}_o) \geq n$*, where* $\mathrm{id}_o = \lambda x^o.x$*.*

Proof. By induction on n, exploiting that lhr preserves β-equivalence.

Theorem 2. *For fixed* $n \geq 2$ *we have* $\mathrm{Lls}_n(h, d) = 2_{n-1}^{\Theta(d \log(h))}$*.*

Proof. Let us first consider the case where $n \geq 3$, as $n = 2$ requires a separate construction for the lower bound. Let us fix $h \geq 3$ and $d \geq 2$. By Proposition 3, we already know that $\mathrm{Lls}_n(d, h) \leq 2_{n-1}^{d \log(h+n+1)}$. Moreover, we have $\mathrm{lh}(S_{h-1,d-1,n-3}\ \mathrm{id}_o) = h$ and $\mathrm{depth}(T_{h-1,d-1,n-3}\ \mathrm{id}_o) = d$, and by Lemma 4 we have $\mathcal{N}(T_{h-1,d-1,n-3}\ \mathrm{id}_o) \geq 2_{n-1}^{(d-1)\log(h-1)}$. To summarize:

$$2_{n-1}^{(d-1)\log(h-1)} \leq \mathrm{Lls}_n(d, h) \leq 2_{n-1}^{d \log(h+n+1)}$$

Therefore, with $n \geq 3$ fixed and d, h parameters we have $\mathrm{Lls}_n(h, d) = 2_{n-1}^{\Theta(d \log(h))}$.

For $n = 2$, the upper bound still holds. For $d, p \geq 2$, define:

$$U_{n,d} = \underline{n}_1\ (\underline{n}_1\ \ldots (\underline{n}_1\ \mathrm{id}_o) \ldots)$$

where there are d copies of $\underline{n}_1$ in total. Then, the term $U_{n,d}$ is s.l.s. and we have $\mathrm{lh}(U_{n,d}) = n + 1$, $\mathrm{depth}(U_{n,d}\ \mathrm{id}_o) = d + 1$, $\mathrm{ord}(U_{n,d}\ \mathrm{id}_o) = 2$ and $\mathcal{N}(U_{n,d}) \geq n^d = 2^{d \log(n)}$. It follows that $\mathrm{Lls}_2(d, h) = 2^{\Theta(d \log(h))}$.

In particular, reduction length for s.l.s. second-order terms of fixed depth is bounded by a polynomial of degree less than the depth.

4 Exact Bounds for General Terms

4.1 Lambda-Lifting to Strongly Locally Scoped Terms

In order to deduce bounds for general terms, we now describe a transformation taking any λ-term M to a corresponding s.l.s. term M'; this transformation is a variant of the familiar notion of λ-lifting [12], adapted to lift variables through generalized redexes as well as β-redexes.

Take a term $M = \lambda x^A.(\lambda y^A.y)\ x$. Obviously, M is not s.l.s.: indeed there is a prime redex $(\lambda y, x)$ and the subterm x has x free. In order to make the variable x "local", we modify the abstraction subterm $\lambda y.y$ to forward explicitly the variable x. We get the term $M' = \lambda x^A.(\lambda y^{A \to A}.y\ x)(\lambda {x'}^A.x')$. The type of y has changed, but not the type of the overall term. Note that the terms M and M' are still β-equivalent, although we are not going to use that explicitly. More importantly, the norm has increased, the order has increased by one, and

$$\frac{y \in \mathrm{fv}(M_1)}{(\lambda x.M)\ M_1\ \ldots\ M_n \to_{\lambda \mathrm{l}} (\lambda x.M[x\ y/x])\ (\lambda y'.M_1[y'/y])\ \ldots\ M_n}$$

$$\frac{M_i \to_{\lambda \mathrm{l}} M_i'}{x_0\ M_1\ \ldots\ M_n \to_{\lambda \mathrm{l}} x_0\ M_1\ \ldots\ M_i'\ \ldots\ M_n} \qquad \frac{M \to_{\lambda \mathrm{l}} M'}{\lambda x.M \to_{\lambda \mathrm{l}} \lambda x.M'}$$

$$\frac{M_1 \to_{\lambda \mathrm{l}} M_1'}{(\lambda x.M)\ M_1\ \ldots\ M_n \to_{\lambda \mathrm{l}} (\lambda x.M)\ M_1'\ \ldots\ M_n}$$

$$\frac{M\ M_2\ \ldots\ M_n \to_{\lambda \mathrm{l}} M'\ M_2'\ \ldots\ M_n'}{(\lambda x.M)\ M_1\ \ldots\ M_n \to_{\lambda \mathrm{l}} (\lambda x.M')\ M_1\ M_2'\ \ldots\ M_n'}$$

Fig. 3. Definition of the λ-lifting expansion $\to_{\lambda \mathrm{l}}$

the other quantities are essentially unchanged. We formalize this construction by the λ-lifting expansion $\to_{\lambda \mathrm{l}}$, defined in Figure 3.

In general $\to_{\lambda \mathrm{l}}$ leaves the type unchanged, although it can change the type of bound variables. Moreover $\to_{\lambda \mathrm{l}}$ terminates, and its normal form is necessarily s.l.s.. Altogether, we have the following result:

Lemma 5. *For any term M, there is a strongly locally scoped M' such that:*

$$\mathrm{lh}(M') \leq \mathrm{lh}(M) + 1 \qquad \mathrm{depth}(M') = \mathrm{depth}(M)$$
$$\mathrm{ord}(M') \leq \mathrm{ord}(M) + 1 \qquad \mathcal{N}(M') \geq \mathcal{N}(M)$$

This is established by a rather lengthy technical proof, studying commutations between $\to_{\lambda \mathrm{l}}$ and $\to_{\mathrm{lhr}}$. Preservation of depth is easy since we do not add generalized redexes, and (relative) preservation of order and local height is established as for $\to_\eta$, by building variants lh' and ord' which take into account the potential expansion of variables, that satisfy $\mathrm{lh}(M) \leq \mathrm{lh}'(M) \leq \mathrm{lh}(M) + 1$ and $\mathrm{ord}(M) \leq \mathrm{ord}'(M) \leq \mathrm{ord}(M) + 1$ and are preserved by $\to_{\lambda \mathrm{l}}$.

4.2 Expanding Variables

For non locally scoped terms, the local height and the depth are rather unnatural quantities, and the bounds are not naturally expressed in terms of them. We convert one to the other using another norm-increasing term transformation.

Lemma 6. *For any term M, there exists a term M' such that $M \to_{\eta^*} M'$,*

$$\mathrm{lh}(M') \leq 2 \qquad \mathrm{depth}(M') \leq \mathrm{h}(M)$$
$$\mathrm{ord}(M') = \mathrm{ord}(M) \qquad \mathcal{N}(M') \geq \mathcal{N}(M)$$

The term M' is obtained by replacing each occurrence x_0 in M of a variable $x : A_1 \to \ldots \to A_n \to o$ by its η-expanded form $\lambda y^1{}^{A_1}.\ldots.\lambda y^n{}^{A_n}.x_0\ y^1\ \ldots\ y^n$. Since we have $M \to_{\eta^*} M'$, we already know that $\mathcal{N}(M') \geq \mathcal{N}(M)$, the other inequalities are easily established by induction.

4.3 Exact Bounds for General Terms

In this section, we are interested in estimating the quantity:

$$\mathrm{Lgen}_n(h) = \max\{\mathcal{N}(M) \mid \mathrm{ord}(M) \leq n \ \&\ \mathrm{h}(M) \leq h\}$$

We do that by applying the tools developed earlier to get an upper bound on the length of reduction, and then prove a matching lower bound by providing terms whose length of reduction asymptotically reaches the upper bound.

Upper bound. Starting from a term M, we first expand variables using Lemma 6, then make it s.l.s. using Lemma 5. This gives M' such that:

$$\begin{array}{rr} \mathrm{lh}(M') \leq 3 & \mathrm{depth}(M') = \mathrm{h}(M) \\ \mathrm{ord}(M') \leq \mathrm{ord}(M) + 1 & \mathcal{N}(M') \geq \mathcal{N}(M) \end{array}$$

By applying Proposition 3, we get:

Proposition 4. *Suppose M is a term. Then, $\mathcal{N}(M) \leq 2^{\mathrm{h}(M)\log(\mathrm{ord}(M)+5)}_{\mathrm{ord}(M)}$.*

Lower bound. We provide a lower bound matching asymptotically the upper bound offered by Proposition 4. The construction is essentially the same as the one used in [2] for the lower bound in terms of height.

For $p \geq 1$ and $k \geq 0$, we define $b^p_0 = \underline{2}_p$ and $b^p_{k+1} = \lambda x^{A_{p-1}}.b^p_k\ (b^p_k\ x)$. Then, we set:

$$B^p_k = b^p_k\ \underline{2}_{p-1}\ \ldots\ \underline{2}_0$$

Note that this term is *not* s.l.s.. By standard arithmetic of Church numerals, we have that for any $p \geq 1, k \geq 0$, $B^p_k \to^*_\beta \underline{2^k_{p+2}}_0$. By Lemma 4 it follows that $\mathcal{N}(B^p_k\ \mathrm{id}_o) \geq 2^k_{p+2}$. It is direct to check that $\mathrm{ord}(B^p_k) = p+2$ and $\mathrm{h}(B^p_k) = k+3$ (for $k \geq 1$). Therefore, we have:

Theorem 3. *For fixed $n \geq 3$ we have $\mathrm{Lgen}_n(h) = 2^{\Theta(h)}_n$.*

For a term M of height h and order n, Beckmann's results [2] predict that any β-reduction chain of M terminates in less than $2^{\Theta(h)}_{n+1}$ steps. It might seem counter-intuitive that our bound (with linear head reduction) is smaller than Beckmann's (with β-reduction) since we substitute only one occurrence at a time, which is obviously longer. However, Beckmann considers arbitrary β-reduction, not head β-reduction. The possibility of reducing in arbitrary locations of the term unlocks much longer reductions, since higher-order free variables or constants can isolate

sections of the term that will never arrive in head position but can still be affected by arbitrary β-reduction. The fact that the length of linear head reduction has the same order of magnitude as head β-reduction is not surprising in the light of Accattoli and Dal Lago's recent result [1] that a similar notion of linear head reduction is quadratically related to head reduction.

5 Conclusion

We have worked out the precise connection between bounding skeletons and syntactic reduction, deducing bounds for linear head reduction in the simply-typed λ-calculus. The analysis uncovers *locally scoped terms*, whose reduction relates closely to game-theoretic interaction. Through this work, we obtain syntax-independent tools to reason on the complexity of programs, hopefully useful in implicit complexity. Although we have only described this connection here for the pure λ-calculus, the connection with games suggest that similar constructions should yield the same results for languages with effects such as control, non-determinism or ground state. In future work we plan plan to generalize these tools to more expressive languages, in particular in the presence of recursion.

Acknowledgment. We gratefully acknowledge the support of the ERC Advanced Grant ECSYM.

References

1. Accattoli, B., Lago, U.D.: On the invariance of the unitary cost model for head reduction. In: Tiwari, A. (ed.) RTA. LIPIcs, vol. 15, pp. 22–37. Leibniz-Zentrum fuer Informatik, Schloss Dagstuhl (2012)
2. Beckmann, A.: Exact bounds for lengths of reductions in typed lambda-calculus. J. Symb. Log. 66(3), 1277–1285 (2001)
3. Bernadet, A., Lengrand, S.: Complexity of strongly normalising λ-terms via non-idempotent intersection types. In: Hofmann, M. (ed.) FOSSACS 2011. LNCS, vol. 6604, pp. 88–107. Springer, Heidelberg (2011)
4. Clairambault, P.: Estimation of the length of interactions in arena game semantics. In: Hofmann, M. (ed.) FOSSACS 2011. LNCS, vol. 6604, pp. 335–349. Springer, Heidelberg (2011)
5. Coquand, T.: A semantics of evidence for classical arithmetic. J. Symb. Log. 60(1), 325–337 (1995)
6. Danos, V., Herbelin, H., Regnier, L.: Game semantics and abstract machines. In: Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science, LICS 1996, pp. 394–405. IEEE (1996)
7. Danos, V., Regnier, L.: How abstract machines implement head linear reduction (2003) (unpublished)
8. Danos, V., Joinet, J.-B.: Linear logic and elementary time. Inf. Comput. 183(1), 123–137 (2003)
9. de Carvalho, D., Pagani, M., de Falco, L.T.: A semantic measure of the execution time in linear logic. TCS 412(20), 1884–1902 (2011)

10. Girard, J.-Y.: Light linear logic. In: Leivant, D. (ed.) LCC 1994. LNCS, vol. 960, pp. 145–176. Springer, Heidelberg (1995)
11. Hyland, J.M.E., Ong, C.-H.L.: On full abstraction for PCF: I, II, and III. Inf. Comput. 163(2), 285–408 (2000)
12. Johnsson, T.: Lambda lifting: Transforming programs to recursive equations. In: Jouannaud, J.-P. (ed.) FPCA 1985. LNCS, vol. 201, pp. 190–203. Springer, Heidelberg (1985)
13. Dal Lago, U., Laurent, O.: Quantitative game semantics for linear logic. In: Kaminski, M., Martini, S. (eds.) CSL 2008. LNCS, vol. 5213, pp. 230–245. Springer, Heidelberg (2008)
14. Schwichtenberg, H.: An upper bound for reduction sequences in the typed λ-calculus. Archive for Mathematical Logic 30(5), 405–408 (1991)