Least Upper Bounds on the Size of Church-Rosser Diagrams in Term Rewriting and λ-Calculus

Jeroen Ketema[1] and Jakob Grue Simonsen[2]

[1] Faculty EEMCS, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
j.ketema@ewi.utwente.nl
[2] Department of Computer Science, University of Copenhagen (DIKU)
Universitetsparken 1, 2100 Copenhagen Ø, Denmark
simonsen@diku.dk

Abstract. We study the Church-Rosser property—which is also known as *confluence*—in term rewriting and λ-calculus. Given a system R and a peak $t \;{}^*\!\!\leftarrow s \rightarrow^* t'$ in R, we are interested in the length of the reductions in the *smallest* corresponding valley $t \rightarrow^* s' \;{}^*\!\!\leftarrow t'$ as a function $\mathrm{vs}_R(m, n)$ of the size m of s and the maximum length n of the reductions in the peak. For confluent term rewriting systems (TRSs), we prove the (expected) result that $\mathrm{vs}_R(m, n)$ is a computable function. Conversely, for every total computable function $\varphi(n)$ there is a TRS with a single term s such that $\mathrm{vs}_R(|s|, n) \geq \varphi(n)$ for all n. In contrast, for *orthogonal* term rewriting systems R we prove that there is a constant k such that $\mathrm{vs}_R(m, n)$ is bounded from above by a function exponential in k and independent of the size of s. For λ-calculus, we show that $\mathrm{vs}_R(m, n)$ is bounded from above by a function contained in the fourth level of the Grzegorczyk hierarchy.

1 Introduction

The Church-Rosser property—also called confluence—is a property of rewriting systems which states that any peak $t \;{}^*\!\!\leftarrow s \rightarrow^* t'$ has a corresponding valley $t \rightarrow^* s' \;{}^*\!\!\leftarrow t'$. The valley and the term s' are said to *complete* the diagram.

In functional programming, the Church-Rosser property ensures that different ways of evaluating a program will always yield the same end result (modulo non-termination): The outcome will be independent of the evaluation order or reduction strategy. In logic, if a deductive system has the Church-Rosser property, the system will be consistent: No statement can both hold and not hold.

While the Church-Rosser property has been shown to hold for a wide variety of rewrite systems, there has, to our knowledge, never been an investigation into the number of reduction steps in a valley that completes the diagram of a peak of a given size (see Figure 1). Succinctly: The question "How large is the valley as a function of the peak?" has apparently never been asked.

$$\begin{tikzcd}
& s \arrow[dl, "\leq n"'] \arrow[dr, "\leq n"] & \\
t \arrow[dr, dotted, "\leq l"'] & & t' \arrow[dl, dotted, "\leq l"] \\
& s' &
\end{tikzcd}$$

Fig. 1. The Church-Rosser property for a rewriting system R with bounds on the lengths of reductions. This paper is concerned with finding least upper bounds of l as a function of n. Succinctly: The peak being the upper half of the diagram, the valley being the lower half, how large is the valley as a function of the peak?

We believe the above question to be *intrinsically* interesting from a theoretical point of view, as Church-Rosser-type results are ubiquitous. We also believe the practical implications in mainstream functional programming to be limited: Standard functional languages like ML and HASKELL employ a fixed evaluation strategy such as call-by-value or call-by-need, and there seems to be little interest in performing optimisations by switching strategies (modulo non-termination). However, for more specialised languages, like declarative DSLs where the evaluation order may not be fixed, there may be practical implications: If, for small peaks, the size of the smallest corresponding valley is so large that a term completing the Church-Rosser diagram cannot be computed using realistic resources, then it matters *very much* what kind of reduction strategy is used: Choosing the 'wrong' evaluation strategy (say, call-by-value) and performing just a few steps of computation could result in a very long reduction before a result is reached—better to backtrack to the original term and try another strategy. Apparently, there is no prior research concerning this problem in the foundational basis of declarative programming—λ-calculus and term rewriting. There does exist some literature on length of shortest and longest reductions to normal form for certain classes of systems [15,8,17], but the Church-Rosser theorem does not concern normal forms: It also applies to systems where some (or all) terms may fail to have normal forms.

In this paper, we perform the first fundamental study of the size of peaks and valleys for systems having the Church-Rosser property; specifically we study how the size of a peak affects the *valley size* of the smallest corresponding valley. We consider three very general settings: That of (arbitrary) first-order term rewriting systems, of orthogonal term rewriting systems (roughly corresponding to first-order functional programs that have no fixed evaluation order), and untyped λ-calculus. We believe that these three areas cover most of the non-specialised areas where the Church-Rosser property occurs; the most significant omission is the case of general higher-order rewrite systems (including general higher-order functional programs and logics with bound variables)—we expect general upper bounds in that case to be difficult to derive (and, likely, to be astronomical), as is foreshadowed by our treatment of λ-calculus in Section 5.

The remainder of this paper proceeds as follows: Section 2 reviews preliminary notions. Section 3 formally introduces valley sizes and shows, respectively, that for term rewriting systems the valley size will always be a computable function and that every computable function can be majorized by a valley size of a specific family of peaks in a term rewriting system. Section 4 gives an exponential upper bound for valley sizes in orthogonal term rewriting systems and Section 5 shows that valley sizes in λ-calculus are bounded from above by a function in the fourth level of the Grzegorczyk hierarchy. Section 6 concludes.

2 Preliminaries

We presuppose a working knowledge of Turing machines and a basic familiarity with term rewriting and λ-calculus. We give brief definitions below. The basic references for term rewriting are [1,16,9]; for λ-calculus we refer the reader to [2,16,5]. For Turing machines, almost any introductory textbook on computability will do, e.g. [12,7,14,5]. Section 5 of the paper uses the Grzegorczyk hierarchy; we refer the reader to [6,11] for definitions.

2.1 Abstract Rewriting and the Church-Rosser Property

We introduce some basic notions related to abstract rewriting and confluence.

Definition 2.1. *An* abstract rewriting system *(ARS) is a pair (A, R) with A a set of* objects *and R a* binary relation *over A where we write $(a, b) \in R$ as $a \to b$. We write $\to^*$ for the reflexive, transitive closure of R. A* reduction *or* rewrite sequence *is a finite sequence $\langle a_1, a_2, \dots, a_n\rangle$ with $a_i \to a_{i+1}$ for all $i < n$, which we usually write as $a_1 \to a_2 \to \cdots \to a_{n-1} \to a_n$, or even as $a_1 \to^* a_n$.*

Definition 2.2. *A* peak *is a pair of reductions*

$$(s \to s_1 \to \cdots \to s_{i-1} \to s_i,\ s \to s'_1 \to \cdots \to s'_{j-1} \to s'_j)$$

both starting from s. A valley *is a pair of reductions*

$$(s_i \to t_1 \to \cdots \to t_{k-1} \to t,\ s'_j \to t'_1 \to \cdots \to t'_{l-1} \to t)$$

both ending in t. We usually write a peak as $s_i \ {}^{i}\!\leftarrow s \to^j s'_j$ and a valley as $s_i \to^k t \ {}^{l}\!\leftarrow s'_j$, occasionally replacing reduction lengths with the Kleene star $*$ when the lengths are unimportant.

Definition 2.3. *An ARS (A, R) has the* Church-Rosser property *or is* confluent *iff for every peak $t \ {}^{*}\!\leftarrow s \to^* t'$ there exists a valley $t \to^* s' \ {}^{*}\!\leftarrow t'$.*

Definition 2.4. *Let (A, R) be an ARS and let $a \in A$. Define the* reduction graph *of a, denoted $\mathcal{G}(a)$, to be the graph (V_a, E_a) inductively defined by*

$$V_{a,n} = \begin{cases} \{a\} & \text{if } n = 0 \\ \{b : \exists a' \in V_{a,n-1}.a' \to b\} & \text{if } n > 0 \end{cases}$$

and

$$E_{a,n} = \begin{cases} \emptyset & \text{if } n = 0 \\ \{(a', b) : a' \in V_{a,n-1},\, b \in V_{a,n}.\, a' \to b\} & \text{if } n > 0 \end{cases}$$

And $\mathcal{G}(a) = (V_a, E_a) = (\bigcup_{n \geq 0} V_{a,n}, \bigcup_{n \geq 0} E_{a,n})$.

Thus, $V_{a,n}$ is the set of objects b such that $a \to^n b$.

2.2 Term Rewriting Systems

We define term rewriting systems. Throughout, we assume a fixed, finite *signature* Σ with each function symbol of non-negative integer arity and a denumerable, infinite set of *variables* V. The set of *terms* over Σ and V, denoted $\text{Ter}(\Sigma, V)$, is defined by induction, as usual. We assume the following.

Definition 2.5. *Let s be a term.*

- *The term s is* ground *if no variables occur in s.*
- *The set of* positions *of s, denoted $\mathbf{pos}(s)$ is the subset of $\mathbb{N}^*$ inductively defined by $\mathbf{pos}(x) = \{\epsilon\}$ and $\mathbf{pos}(f(s_1, \ldots, s_n)) = \{\epsilon\} \cup (\bigcup_{i=1}^{n} i \cdot \mathbf{pos}(s_i))$.*
- *The set of* variables *of s, denoted $vars(s)$, is the finite subset of V inductively defined by $vars(x) = \{x\}$ and $vars(f(s_1, \ldots, s_n)) = \bigcup_{i=1}^{n} vars(s_i)$.*
- *The* size *of s, denoted $|s|$, is defined inductively as:*
  - $|x| = 1$*;*
  - $|f(s_1, \ldots, s_n)| = 1 + |s_1| + \cdots + |s_n|$.

Positions are equipped with a partial (strict) order $\prec$ such that $p \prec q$ if p is a proper prefix of q. Moreover, we write $s|_p$ for the *subterm* of a term s that occurs at position $p \in \mathbf{pos}(s)$.

Substitutions, written $\theta : V \longrightarrow \text{Ter}(\Sigma, V)$, are defined as usual. Contexts are terms over $\Sigma \uplus \{\Box\}$, written as $C[]$, where we say that that a context $C[]$ is a *k-hole context* if there are exactly k occurrences of $\Box$ in $C[]$.

Definition 2.6. *A* rule *over Σ is a pair (l, r), invariably written $l \to r$, where l and r are terms over Σ such that $l \notin V$ and $vars(r) \subseteq vars(l)$. A term s* rewrites *to a term t by $l \to r$ if there is a one-hole context $C[]$ and a substitution θ such that $s = C[\theta(l)]$ and $t = C[\theta(r)]$.*

A term rewriting system (TRS) *is a pair (Σ, R) with Σ a signature and R a finite set of rules over Σ.*

We usually suppress explicit mention of the signature Σ and refer to the TRS (Σ, R) as R. Every TRS R gives rise to an ARS (A, R') in the obvious fashion: The elements of A are the terms and R' is the above rewrite relation.

Definition 2.7. *A rule is* left-linear *if every variable of occurs at most once in l. A TRS R is left-linear if all of its rules are.*

A rule $l_1 \to r_1$ is said to overlap *a rule $l_2 \to r_2$ at position $p \in \mathbf{pos}(l_2)$ if $l_2|_p \notin V$ and there are two substitutions σ, θ such that $\theta(l_1) = \sigma(l_2|_p)$. A TRS (Σ, R) is said to be* orthogonal *if R is left-linear, and the only overlaps of rules in R are those where a rule overlaps itself at position ϵ.*

Two TRSs (Σ_0, R_0) and (Σ_1, R_1) are said to be mutually orthogonal *if they are left-linear, and no rule of R_0 overlaps with a rule of R_1, and vice versa.*

2.3 λ-Calculus

The (untyped) λ-calculus is the ARS $(\Lambda, \to_\beta)$ with Λ the set of objects M defined inductively by

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

where $x \in V$ is a variable and with $\to_\beta$ the rewrite relation induced by the β-rule:

$$(\lambda x.M)\, N \to_\beta M\{N/x\}$$

where $M\{N/x\}$ equals M with N substituted for every free occurrence of x in M. Contexts for λ-calculus are defined as for TRSs. We assume the following.

Definition 2.8. *Let M be a λ-term.*

- *The set of* positions *of M, denoted $\mathbf{pos}(M)$, is the subset of $\mathbb{N}^*$ inductively defined by $\mathbf{pos}(x) = \{\epsilon\}$, $\mathbf{pos}(\lambda x.M) = \{\epsilon\} \cup 0 \cdot \mathbf{pos}(M)$, and $\mathbf{pos}(M_1\, M_2) = \{\epsilon\} \cup 0 \cdot \mathbf{pos}(M_1) \cup 1 \cdot \mathbf{pos}(M_2)$.*
- *The* size *of M, denoted $|M|$, is defined by inductively as:*
  - $|x| = 1$*;*
  - $|\lambda x.M| = 1 + |M|$*;*
  - $|M\, N| = |M| + |N|$*.*

Positions are again equipped with a partial (strict) order $\prec$ such that $p \prec q$ if p is a proper prefix of q.

The notion of a *residual* of a β-redex across reduction, i.e. the formalisation of “what happens” to a redex across a reduction, is defined as usual [2]. Recall that a *development* of a set of redexes U of a λ-term M is a reduction starting from M contracting a residual of a redex in U in each step. Moreover, a development is *complete* if the set of residuals of redex in U across the development is empty. We have the following.

Theorem 2.9 (Finite Developments Theorem). *Let M be a λ-term and U a set of redexes of M. All developments of U are finite and there is a unique λ-term N that is the final term of all complete developments of U.*

3 Valley Sizes in ARSs and TRSs

We now define the main object of study in this paper: The function vs_R.

Definition 3.1. *Let (A, R) be an ARS that has the Church-Rosser property and let $|\cdot| : A \longrightarrow \mathbb{N}$ be a function (‘size’) such that for each $m \in \mathbb{N}$, the set $\{a \in A : |a| \leq m\}$ is finite. The* valley size $\mathrm{vs}_R : \mathbb{N}^2 \longrightarrow \mathbb{N}$ *is defined as $\mathrm{vs}_R(m, n) = l$ where l is the least number such that for every object a with $|a| \leq m$ and every peak starting from a with reductions of length at most n there is a corresponding valley with reductions of length at most l.*

Observe that vs_R is well-defined as $\{a \in A : |a| \leq m\}$ is finite and (A, R) has the Church-Rosser property. The 'size' function $|\cdot|$ will depend on the class of ARSs considered. In this paper, we are concerned solely with term rewriting systems and λ-calculus where we consider terms modulo the renaming of (free) variables to ensure $\{a \in A : |a| \leq m\}$ is finite.

We employ $|a| \leq m$, and not $|a| = m$, in the above the definition to ensure that vs_R is monotone. Replacing $|a| \leq m$ by $|a| = m$ gives us a less well-behaved function; The example we give below demonstrates this: $\mathrm{vs}_R(2, 1)$ would be equal to 1 instead of being equal to $\mathrm{vs}_R(1, 1) = 2$.

In an ARS with the Church-Rosser property, there will usually be several (or even infinitely many) different valleys that complete the diagram of a specific peak. If the ARS is both Church-Rosser and terminating, a valley can always be found by reducing to normal form (but this may yield a valley with longer reductions than necessary); if the ARS has cycles, there may be an infinite number of possible valleys.

The function $\mathrm{vs}_R(m, n)$ picks the *smallest* valley for each specific peak, but has to take into account all peaks with a starting term of size (at most) m and reductions of size (at most) n; thus, $\mathrm{vs}_R(m, n)$ may be larger than what is needed for 'most' peaks—it gives the least valley size that will *surely* work for all terms and peaks limited by m and n.

We illustrate the workings of vs_R by computing $\mathrm{vs}_R(2, 1)$ for a small TRS in the following example.

Example 3.2. Let R be the TRS with rules

$$\left\{ \begin{array}{lll} a \to b & b \to d & d \to e \\ a \to c & c \to a & g(x) \to h(a) \\ a \to e & d \to a & h(x) \to e \end{array} \right\}$$

This TRS is confluent[3] (and normalising, but not terminating[4]).

Consider the peak $g(b) \leftarrow g(a) \to g(c)$. Some valleys completing the diagram are: (i) $g(b) \to h(a) \leftarrow g(c)$, (ii) $g(b) \to g(d) \to g(a) \to g(c)$, (iii) $g(b) \to h(a) \to e \leftarrow h(a) \leftarrow g(c)$, (iv) $g(b) \to g(d) \to h(a) \leftarrow g(c)$, and so on. Observe there are an infinite number of valleys of the form $g(b) \to g(d) \to^* g(a) \to h(a) \leftarrow g(a) {}^*\!\leftarrow g(c)$ and that there is no *largest* valley completing the diagram.

The *smallest* possible valley is the first of the above: Both reductions have length 1. Note that this valley does *not* involve normal forms, and that any valley with reductions to normal form involves strictly longer reductions.

By definition of the size of terms (Def. 2.5), the term $g(a)$ has size 2, and by inspection, we find that for any peak with reductions of length at most 1 starting from a term of size 2, there is a corresponding valley where each reduction has length at most 1. However, for terms of size 1, there is the peak $b \leftarrow a \to c$ whose

[3] The system has the unique normal form property and is weakly confluent (see [16] for details and definitions).

[4] Normalisation and termination are also called, respectively, weak normalisation and strong normalisation.

smallest valleys involve reductions of length 2, e.g. $b \to d \to a \leftarrow c$. Thus, for peaks involving terms of size *at most* 2 and reductions of length *at most* 1, the smallest corresponding valleys involve reductions of length *at most* 2, and there is a peak that needs a valley with reductions of length 2. Hence, $\mathrm{vs}_R(2,1) = 2$.

In the above example, R was a *non-orthogonal* TRS. We shall see in Sect. 4 that for orthogonal TRSs *the term size does not matter*; thus, the first argument of vs_R can be dropped in that case.

Remark 3.3. The function vs_R need not be computable for an ARS: Let $h : \mathbb{N} \longrightarrow \mathbb{N}$ be any non-computable total function, let $A = \mathbb{N} \cup \mathbb{N}^2$, and let $|i| = i$ and $|(i,j)| = i + j$ for all $i \in \mathbb{N}$ and $(i,j) \in \mathbb{N}^2$. Define, for every $m \geq 1$ and $n > 1$: $m \to (m,1)$, $m \to (m, h(m)+1)$, and $(m,n) \to (m, n-1)$. Then, (A, R) has the Church-Rosser property by the last rule, but $\mathrm{vs}_R(|m|, 1) = h(m)$, whence $\mathrm{vs}_R(m,n)$ is not computable.

3.1 The Valley Size is a Computable Function for TRSs

We now show that vs_R is computable for *arbitrary* term rewriting systems R; in fact it is *uniformly* so: There is a program that, given an encoding of a confluent TRS, returns another program that computes vs_R. We give a formal account in the following.

Recall that we consider only TRSs with a *finite* signature and a *finite* number of rules. As terms are inductively defined, it is clear that every such TRS (Σ, R) can be recursively encoded and decoded as an integer $j_{(\Sigma,R)}$. In the remainder of the paper we assume a fixed such encoding and decoding.

Theorem 3.4. *There is a (partial) computable function $g : \mathbb{N}^3 \longrightarrow \mathbb{N}$ such that if $j_{(\Sigma,R)}$ encodes a TRS (Σ, R) with the Church-Rosser property, then $\mathrm{vs}_{(\Sigma,R)}(m,n) = g(j_{(\Sigma,R)}, m, n)$ for all m, n.*

Proof. Let P be a program that does the following: On input $(j_{(\Sigma,R)}, m, n)$, P decodes $j_{(\Sigma,R)}$, builds all terms $t_1, \ldots, t_l$ (modulo the renaming of variables), of size at most m over Σ, and stores them in memory. Using R and the fact each term has a finite number of one-step reducts, for each $t_i \in \{t_1, \ldots, t_l\}$, P brute-force applies all rules of R to obtain, after a finite number of steps, every term t'_i such that $t_i \to^{\leq n} t'_i$. Next, for every pair (s_i, s'_i) of such terms, P uses R to simultaneously build increasingly larger parts $(\bigcup_{0 \leq k \leq j} V_{s_i,k}, \bigcup_{0 \leq k \leq j} E_{s_i,k})$ and $(\bigcup_{0 \leq k \leq j} V_{s'_i,k}, \bigcup_{0 \leq k \leq j} E_{s'_i,k})$ of the reduction graphs of s_i and s'_i. If (Σ, R) has the Church-Rosser property, eventually a j is reached such that a term r_i exists that is both in $\bigcup_{0 \leq k \leq j} V_{s_i,k}$ and $\bigcup_{0 \leq k \leq j} V_{s'_i,k}$. The program P stores the least such j for (s_i, s'_i). Clearly, the least such j is equal to the number of steps in the longest reduction of the smallest valley of s_i and s'_i. After iterating over every pair (s_i, s'_i), P takes the maximum of the stored lengths and returns it. This value is clearly $\mathrm{vs}_{(\Sigma,R)}(m,n)$. Thus, P computes a function $g(j_{(\Sigma,R)}, m, n)$ as desired. □

Theorem 3.5. *If (Σ, R) is a TRS having the Church-Rosser property, then $\mathrm{vs}_{(\Sigma,R)}$ is a total computable function.*

Proof. By Theorem 3.4, we have $\mathrm{vs}_{(\Sigma,R)}(m,n) = g(j_{(\Sigma,R)}, m, n)$ for all m, n. That $\mathrm{vs}_{(\Sigma,R)}$ is a partial computable function follows immediately by the s-m-n Theorem [13]. That the function $\mathrm{vs}_{(\Sigma,R)}$ is total follows by the fact that $\mathrm{vs}_{(\Sigma,R)}$ is well-defined by the comments below Definition 3.1. □

3.2 All Computable Functions can be Majorized by Valleys in TRSs

Above we showed that for every TRS (Σ, R) the size vs_R is computable; colloquially, we have a very tight computable upper bound on valley sizes. Naïvely, one might conjecture that an even tighter bound is obtainable—e.g. that $\mathrm{vs}_{(\Sigma,R)}$ is always primitive recursive. We now proceed to show this is not possible in a very strong sense: For every computable function $\varphi : \mathbb{N} \longrightarrow \mathbb{N}$, there is a TRS and a single term of some size m such that $\mathrm{vs}_{(\Sigma,R)}(m,n) \geq \varphi(n)$ for all $n \geq 2$.

Encoding Turing Machines. We shall use the following (inconsequential) constraints on the Turing machines we encode:

Definition 3.6. *All Turing machines are one-head, one-tape machines with no auxiliary input or output tapes. There are no transitions to the initial state q_s, nor are there any transitions from the halting state q_h. The input and tape alphabets of all Turing machines are $\{0, 1, \Box\}$ where $\Box$ is 'blank' as usual. All inputs are assumed to be given in unary; hence, $n \in \mathbb{N}$ is encoded as 0^n. The initial configuration of a Turing machine will always be in the initial state with the input starting in the tape cell immediately to the right of the read/write head. The machine is assumed never to be stuck on a legal configuration; for every state $q \in Q \setminus \{q_h\}$ and every element $b \in \{0, 1, \Box\}$, the transition $\delta(q, b)$ is defined.*

We give the standard encoding of [16]. The tape alphabet is modelled by unary function symbols 0, 1 and $\Box$, respectively. Both tape ends are modelled by the nullary symbol $\rhd$. The representation of the string $01\Box1$ enclosed on the right by a tape end will thus be $0(1(\Box(1(\rhd))))$; the left tape end and position of the read/write head of the machine will be encoded in the TRS rules representing the Turing machine transitions. For each state $q \in Q$ of the machine, we assume a binary function symbol q. The TRS induced by the transitions of a Turing machine M is given in Figure 2.

For our purposes, we augment $\Delta(M)$ with a constant symbol T and a binary function symbol r. In addition, we augment the rewrite rules of $\Delta(M)$ with the rule set from Figure 3, which extends the rule set from [4, Sect. 5] with the rules $r(x, 0y) \to r(x, 00y)$ and $r(x, 00y) \to r(x, 0y)$.

To prove confluence of $\Delta_C(M)$ in the case where M halts on all inputs, we first give a general lemma concerning mutually orthogonal systems. For $i \in \{0, 1\}$ we define $\bar{i} = (i + 1) \bmod 2$.

Rewrite rules induced by transition rules of the Turing machine M ($\varDelta_N(M)$):

(L/R)-move	rewrite rules (for each $q \in Q$, $a \in \{0, 1, \Box\}$)
$\delta(q, b) = (q', b', R)$	$q(x, by) \to q'(b'x, y)$
$\delta(q, b) = (q', b', L)$	$q(ax, by) \to q'(x, ab'y)$

Extra rules ($\varDelta_E(M)$):

(L/R)-move	extra rewrite rules (for each $q \in Q$, $a \in \{0, 1, \Box\}$)
$\delta(q, \Box) = (q', b', R)$	$q(x, \triangleright) \to q'(b'x, \triangleright)$
$\delta(q, b) = (q', b', L)$	$q(\triangleright, by) \to q'(\triangleright, \Box b'y)$
$\delta(q, \Box) = (q', b', L)$	$q(ax, \triangleright) \to q'(x, ab'\triangleright)$ $q(\triangleright, \triangleright) \to q'(\triangleright, \Box b'\triangleright)$

$$\varDelta(M) = \varDelta_N(M) \cup \varDelta_E(M)$$

Fig. 2. Basic encoding $\varDelta(M)$ of a Turing machine M

Rule for transitioning to T when the halting state has been reached (†):

$$\boxed{q_h(x, y) \to T}$$

Rules for non-deterministic choice of $n \in \mathbb{N}$ ($\varDelta_{\text{ndt}}(M)$):

$r(x, \triangleright) \to T$	$r(\triangleright, y) \to q_s(\triangleright, y)$
$r(x, 0y) \to r(0x, y)$	$r(0x, y) \to r(x, 0y)$
$r(x, 0y) \to r(x, 00y)$	$r(x, 00y) \to r(x, 0y)$

$$\varDelta_C(M) = \varDelta(M) \cup \{\dagger\} \cup \varDelta_{\text{ndt}}(M)$$

Fig. 3. Extra rules for non-deterministic choice and confluence

Lemma 3.7. *Let R_0 and R_1 be mutually orthogonal systems such that for each $i \in \{0, 1\}$ and for each peak $t \;{}_i^*\!\!\leftarrow s \to_i^* t'$, there* either *exists a corresponding valley $t \to_i^* s' \;{}_i^*\!\!\leftarrow t'$,* or *a corresponding valley $t \to_{\bar{i}}^* s' \;{}_{\bar{i}}^*\!\!\leftarrow t'$. Then, $R_0 \cup R_1$ has the Church-Rosser property.*

Proof (Sketch). Straightforward tiling of peaks. $\Box$

Proposition 3.8. *If M halts on all inputs, the two systems $R_0 = \varDelta(M) \cup \{(\dagger)\}$ and $R_1 = \varDelta_{ndt}(M)$ satisfy the conditions of Lemma 3.7.*

Proof. Both systems are left-linear and clearly no left-hand side of a rule of R_0 overlaps with a left-hand side of a rule of R_1 and vice versa, whence the two systems are mutually orthogonal. Also, R_0 is orthogonal, hence has the Church-Rosser property. Furthermore, observe that two rules from of $\varDelta_{\text{ndt}}(M)$ can only overlap at the root. As there are no collapsing rules in $\varDelta_{\text{ndt}}(M)$, we thus obtain confluence if every peak $t \;{}^*\!\!\leftarrow r(s, s') \to^* t'$ has a corresponding valley. By inspection of the rules of $\Delta_{\mathrm{ndt}}(M)$, it is seen that if $r(s,s') \to_1^* r(t,t')$, then $r(t,t') \to_1^* r(s,s')$. Thus, the only peaks of R_1 that do not have corresponding valleys in R_1 are the ones on the form

$$T \ {}_1^{*}\!\!\leftarrow r(s,s') \to_1^* q_s(\rhd, t)$$

By inspection of the rules of $\Delta_{\mathrm{ndt}}(M)$, we see that such a peak is only possible if $t = 0^n \rhd$. As M halts on all configurations, we obtain $q_s(\rhd, t) \to_0^* q_h(t', t) \to_0 T$, concluding the proof. □

Corollary 3.9. *If M halts on all inputs, then $\Delta_C(M)$ has the Church-Rosser property.*

We have the following lemma:

Lemma 3.10. *Let $\varphi_M : \mathbb{N} \longrightarrow \mathbb{N}$ be a total computable function. Then there is a Turing machine M' that (i) halts on all inputs, and (ii) on input 0^n halts in at least $\varphi_M(n)$ steps.*

Proof. Let M' be the Turing machine containing an inlined copy of M and, on input 0^n, computes $k = \varphi_M(n)$, then performs k "idle steps" before halting. As M halts on all inputs, so does M', and by construction M' runs for at least $\varphi_M(n)$ steps before halting. □

Majorizing a Computable Function with Valleys in a TRS. We now show that for every computable function $\varphi_M : \mathbb{N} \longrightarrow \mathbb{N}$, there exists a TRS R having the Church-Rosser property and a term s such that there is a peak of size n with smallest corresponding valley of size $\varphi_M(n)$. Thus, $\mathrm{vs}_{(\Sigma,R)}(m,n) \geq \varphi_M(n)$ for all $m \geq |s|$.

Theorem 3.11. *For every total computable function $\varphi_M : \mathbb{N} \longrightarrow \mathbb{N}$, there exists a TRS R having the Church-Rosser property, a ground term s, and a ground normal form s_0 of R such that, for every natural number n, there is a term s_n with (i) $s_0 \ {}^2\!\!\leftarrow s \to^n s_n$, (ii) $s_n \to^* s_0$, and (iii) every reduction $s_n \to^* s_0$ has length at least $\varphi_M(n)$.*

$$\begin{tikzcd} s \arrow[r, "n"] \arrow[d, "2"'] & s_n \arrow[d, dotted, "\geq\varphi_M(n)"] \\ s_0 \arrow[r, equal] & s_0 \end{tikzcd}$$

Proof. Let M' be the Turing machine obtained by applying Lemma 3.10 to φ_M. Then, M' halts on all inputs and halts in at least $\varphi_M(n)$ steps on input 0^n for all $n \in \mathbb{N}$. We set $R = \Delta_C(M')$, $s = r(\rhd, 0\rhd)$, $s_0 = T$, and $s_n = q_s(\rhd, 0^n\rhd)$. For all $n \in \mathbb{N}$, we then have $s \to r(0\rhd, \rhd) \to T$ and $s \to^n s_n$. Observe that R has the Church-Rosser property by Corollary 3.9, and that s is ground. By the fact that each step of $\Delta(M')$ simulates exactly one step of M', we obtain that $q_s(\rhd, 0^n\rhd) \to^m q_h(t,t')$ (for terms t, t') where $m \geq \varphi_M(n)$. As M' is deterministic, this is the only possible reduction from $q_s(\triangleright, 0^n \triangleright)$ to $q_h(t, t')$. Finally, we use rule (†) to obtain $q_h(t, t') \rightarrow T = s_0$. Hence, $s_n \rightarrow^* s_0$ and all such reductions are of length at least $\varphi_M(n)$. □

We hence have:

Theorem 3.12. *For every total computable function* $\varphi_M : \mathbb{N} \longrightarrow \mathbb{N}$*, there is an explicitly constructible TRS* (Σ, R) *that has the Church-Rosser property and an explicitly constructible ground term* s *of* R *such that for all* $m \geq |s|$ $\mathrm{vs}_{(\Sigma,R)}(m, n) \geq \varphi_M(n)$*.*

4 Bounds on Valley Sizes in Orthogonal TRSs

For *orthogonal* TRSs, much better bounds can be obtained than those presented in the previous section. We shall prove existence, for every TRS R, of a constant μ_R such that $\mathrm{vs}_{(\Sigma,R)}(m, n) \leq n \cdot (\mu_R)^n$.

Definition 4.1. *Let* R *be a TRS. The* parallel rewrite relation $\Rightarrow$ *is defined as follows:* $s \Rightarrow^k t$ *if there is a* k*-hole context such that (i)* $s = C[s_1, \ldots, s_k]$*, (ii)* $t = C[t_1, \ldots, t_k]$*, and (iii) for all* $1 \leq i \leq k$*, we have* $s_i \rightarrow t_i$*.*

Definition 4.2. *The* multiplicity *of a finite TRS* R*, denoted* μ_R*, is defined as:*

$$\max_{l \rightarrow r \in R} \max_{x \in vars(l)} (1, \textit{number of occurrences of } x \textit{ in } r)$$

Thus, the multiplicity of a system is simply the maximum number of times that a variable can occur in a right-hand side of a rule of R.

Example 4.3. Let $R = \{f(x, y) \rightarrow g(x, x, y),\ g(x, y, z) \rightarrow f(x, z)\}$ Then $\mu_R = 2$ as x occurs twice in the right-hand side of the rule $f(x, y) \rightarrow g(x, x, y)$, and no variable occurs more often in a right-hand side.

Lemma 4.4 (Parallel Moves Lemma with reduction lengths). *Let* R *be an orthogonal TRS and let* s *be a term. If* $t \; {}^{m}\!\Leftarrow s \Rightarrow^n t'$ *is a peak, then there is a valley* $t \Rightarrow^{\leq n \cdot \mu_R} s' \; {}^{\leq m \cdot \mu_R}\!\Leftarrow t'$*.*

Proof. Existence of a valley follows by the standard Parallel Moves Lemma [1]. The reduction in $t \Rightarrow s'$ consists of a parallel contraction of the residuals of the redexes contracted in $s \Rightarrow^n t'$ across contraction of the m redexes in $s \Rightarrow^m t$, and vice versa. The step $s \Rightarrow^m t$ consists of m separate $\rightarrow$-steps, each contracting a single redex parallel to the other $m - 1$ redexes. By the definition of the rewrite relation $\rightarrow$, every single step using a rule $l \rightarrow r$ may copy each of its subterms by as many times a variable occurs in r. Each of the n parallel redexes contracted in $s \Rightarrow^n t'$ may, or may not, occur inside one of the subterms copied by a redex in $s \Rightarrow^m t$. The total number of copies that occur in t is hence bounded from above by n times the maximum number of times that a single variable can occur in the right-hand side of a rule, hence $n \cdot \mu_R$. The situation with $m \cdot \mu_R$ is symmetrical. □

Theorem 4.5. *Let the TRS R be orthogonal and let s be a term in R with a peak $t\ {}^{j}\!\!\leftarrow s \rightarrow^{i} t'$. Then there is a valley $t \rightarrow^{\leq j\cdot(\mu_R)^i} s'\ {}^{\leq i\cdot(\mu_R)^j}\!\!\leftarrow t'$. Hence, $\mathrm{vs}_{(\Sigma,R)}(m,n) \leq n\cdot(\mu_R)^n$.*

Proof. As every $\rightarrow$-reduction is also a $\Rightarrow$-reduction and as $\Rightarrow^* = \rightarrow^*$, repeated application of Lemma 4.4 allows us to erect the tiling diagram in Figure 4. The result now follows by tallying the number of steps on the right-most and bottom-most sides of the diagram. □

$$\begin{tikzcd}
s_{0,0} = s \arrow[r, Rightarrow, "1"] \arrow[d, Rightarrow, "1"'] & s_{0,1} \arrow[r, Rightarrow, "1"] \arrow[d, Rightarrow, "\leq 1\cdot\mu_R"] & s_{0,2} \arrow[r, Rightarrow, "1"] \arrow[d, Rightarrow, "\leq \mu_R\cdot\mu_R"] & \cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & s_{0,i-1} \arrow[r, Rightarrow, "1"] \arrow[d, Rightarrow, "\leq(\mu_R)^{i-1}"] & s_{0,i} = t \arrow[d, Rightarrow, "\leq(\mu_R)^{i-1}\cdot\mu_R"] \\
s_{1,0} \arrow[r, Rightarrow, "\leq 1\cdot\mu_R"'] \arrow[d, Rightarrow, "1"'] & s_{1,1} \arrow[r, Rightarrow, "\leq 1\cdot\mu_R"'] \arrow[d, Rightarrow, "\leq 1\cdot\mu_R"] & s_{1,2} \arrow[r, Rightarrow, "\leq 1\cdot\mu_R"'] \arrow[d, Rightarrow, "\leq \mu_R\cdot\mu_R"] & \cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & s_{1,i-1} \arrow[r, Rightarrow, "\leq 1\cdot\mu_R"'] \arrow[d, Rightarrow, "\leq(\mu_R)^{i-1}"] & s_{1,i} \arrow[d, Rightarrow, "\leq(\mu_R)^{i-1}\cdot\mu_R"] \\
s_{2,0} \arrow[r, Rightarrow, "\leq \mu_R\cdot\mu_R"'] \arrow[d, Rightarrow, "1"'] & s_{2,1} \arrow[r, Rightarrow, "\leq \mu_R\cdot\mu_R"'] \arrow[d, Rightarrow, "\leq 1\cdot\mu_R"] & s_{2,2} \arrow[r, Rightarrow, "\leq \mu_R\cdot\mu_R"'] \arrow[d, Rightarrow, "\leq \mu_R\cdot\mu_R"] & \cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & s_{2,i-1} \arrow[r, Rightarrow, "\leq \mu_R\cdot\mu_R"'] \arrow[d, Rightarrow, "\leq(\mu_R)^{i-1}"] & s_{2,i} \arrow[d, Rightarrow, "\leq(\mu_R)^{i-1}\cdot\mu_R"] \\
\cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & \cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & \cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & \cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & \cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & \cdot \arrow[d, dotted, dash] \\
s_{j-1,0} \arrow[r, Rightarrow, "\leq(\mu_R)^{j-1}"'] \arrow[d, Rightarrow, "1"'] & s_{j-1,1} \arrow[r, Rightarrow, "\leq(\mu_R)^{j-1}"'] \arrow[d, Rightarrow, "\leq 1\cdot\mu_R"] & s_{j-1,2} \arrow[r, Rightarrow, "\leq(\mu_R)^{j-1}"'] \arrow[d, Rightarrow, "\leq \mu_R\cdot\mu_R"] & \cdot \arrow[r, dotted, dash] \arrow[d, dotted, dash] & s_{j-1,i-1} \arrow[r, Rightarrow, "\leq(\mu_R)^{j-1}"'] \arrow[d, Rightarrow, "\leq(\mu_R)^{i-1}"] & s_{j-1,i} \arrow[d, Rightarrow, "\leq(\mu_R)^{i-1}\cdot\mu_R"] \\
s_{j,0} = t' \arrow[r, Rightarrow, "\leq(\mu_R)^{j-1}\cdot\mu_R"'] & s_{j,1} \arrow[r, Rightarrow, "\leq(\mu_R)^{j-1}\cdot\mu_R"'] & s_{j,2} \arrow[r, Rightarrow, "\leq(\mu_R)^{j-1}\cdot\mu_R"'] & \cdot \arrow[r, dotted, dash] & s_{j,i-1} \arrow[r, Rightarrow, "\leq(\mu_R)^{j-1}\cdot\mu_R"'] & s_{j,i} = t
\end{tikzcd}$$

Fig. 4. Tiling diagram annotated with reduction lengths for the proof of Theorem 4.5

Remark 4.6. The bounds of the above theorem are tight for non-erasing TRSs (Σ, R) in the following sense: There is an infinite number of terms s such that $\mathrm{vs}_{(\Sigma,R)}(|s|, n) = n\cdot(\mu_R)^n$. Let $l \rightarrow r$ be a rule such that there is a variable x in l that occurs μ_R times in r. For $j \geq 0$ let s_j be the term defined inductively by $s_0 = l$ and $s_{j+1} = l[s_j]_{p_x}$ where p_x is the (unique, by left-linearity) position of the variable x in l. For every $n \geq 1$, consider the term s_{2n} and the peak obtained by performing (a) a complete development of the n outermost redexes, and (b) the n innermost redexes; observe that both of these reductions are of length precisely n. The (a)-reduction copies the 'inner' term s_n a total of $(\mu_R)^n$ times ending in some term t. The (b)-reduction leaves exactly one copy of each of the top n redexes, ending in some term t'. To complete the Church-Rosser diagram, one needs to reach the term obtained by a complete development of all redexes in s_{2n}. From term t', a total of n steps is required to reach this step. From term

t, reaching the final term requires the contraction of n redexes in $(\mu_R)^n$ parallel subterms, for a total of $n \cdot (\mu_R)^n$ steps.

5 A Bound on Valley Sizes in λ-Calculus

In λ-calculus we cannot expect the valley size $\mathrm{vs}_\Lambda(m,n)$ to be independent of m as in Theorem 4.5: In λ-calculus, the growth rate of terms across β-steps depends on the number of bound variables in the original term. Hence, as the size of the valleys is determined by the number of copies of redexes, $\mathrm{vs}_\Lambda(m,n)$ must thus depend on m.

Of the many available proofs of the Church-Rosser property for λ-calculus, the one most amenable to analysis of reduction lengths consists of "tiling a peak" with commuting squares of so-called *complete developments* of sets of redexes in a single term; the construction is essentially the same as the one depicted by figure in the proof of Theorem 4.5 (indeed, the figure is often called a *tiling diagram* [16]), except that for λ-calculus, the "parallel reduction" relation used in each square is a complete development of a set of redexes in a single term. An analysis of this proof reveals $\mathrm{vs}_\Lambda(m,n)$ to be bounded from above by a function in the fourth level, $\mathcal{E}_4$, of the Grzegorczyk hierarchy, roughly corresponding to limited recursion on iterated exponentiation, also called *tetration*—a typical function is $n \mapsto 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$ (2 taken to the power of itself n times). Indeed, considering the special case of the so-called "Strip Lemma" where one reduction in the peak has length 1 and the other length k (see Lemma 5.3), naïve analysis yields a bound $|M_{i,0}|^{2^{2\cdot|M_{i,0}|^{2^k}+k}}$ for the length of the reduction $M_{i+1,0} \to^* M_{i+1,k}$. We give a somewhat better bound in the present section; this bound is still in $\mathcal{E}_4$, but much less than the bound obtained by naïve analysis: $|M_{i,0}|^{2^{2^k}+k}$ for the Strip Lemma.

Upper bounds on the length of developments [3] and standard reductions [17] have been investigated in the literature, as have lower bounds for normalising reductions in typed systems [15]; the present paper is the first study of the size of Church-Rosser diagrams in λ-calculus.

Proposition 5.1. *Let $M_0 \to M_1 \to \cdots M_{n-1} \to M_n$ be a reduction of length $n \geq 0$, and let u be a redex in M_0. For each position $p \in \mathbf{pos}(M_n)$, at most 2^n residuals of u occur in M_n at prefix positions of p.*

Proof (Sketch). By induction on n. □

Lemma 5.2. *Let M be a term and U a set of redexes in M. Suppose for each $p \in \mathbf{pos}(M)$ that at most $i \geq 0$ other redexes from U occur at prefix positions of p. Then contracting all redexes in U yields a term of at most size $|M|^{2^{2\cdot i}}$.*

Proof (Sketch). By induction on i. □

5.1 Bounds for the Strip Lemma

Lemma 5.3 (Strip Lemma with term sizes and reduction lengths). *Let $k \geq 1$ and consider the peak*

$$M_{i+1,0} \,{}_{\beta}\!\leftarrow M_{i,0} \to_{\beta} M_{i,1} \to_{\beta} M_{i,2} \to_{\beta} \cdots M_{i,k-1} \to_{\beta} M_{i,k}\,.$$

Then we may obtain a valley by tiling the peak using the Finite Developments Theorem in the following way:

$$\begin{tikzcd}
M_{i,0} \arrow[r, "1"] \arrow[d, "1"'] & M_{i,1} \arrow[r, "1"] \arrow[d, dotted, "*"'] & M_{i,2} \arrow[r, dotted, dash, "1"] \arrow[d, dotted, "*"'] & M_{i,k-1} \arrow[r, "1"] \arrow[d, dotted, "*"'] & M_{i,k} \arrow[d, dotted, "*"'] \\
M_{i+1,0} \arrow[r, dotted, "*"] & M_{i+1,1} \arrow[r, dotted, "*"] & M_{i+1,2} \arrow[r, dotted, dash] & M_{i+1,k-1} \arrow[r, dotted, "*"] & M_{i+1,k}
\end{tikzcd}$$

where the following holds for $1 \leq j \leq k$:

1. $|M_{i,j}| \leq |M_{i,0}|^{2^j}$ *and* $|M_{i+1,j}| \leq |M_{i,0}|^{2^{2^{j+1}+j}}$*, and*
2. *the reduction* $M_{i+1,j-1} \to_{\beta}^{*} M_{i+1,j}$ *has length at most* $|M_{i,0}|^{2^{2^j+j-1}}$.

Moreover, the reduction $M_{i+1,0} \to_{\beta}^{*} M_{i+1,k}$ *has length at most* $|M_{i,0}|^{2^{2^k+k}}$.

Proof. If $P \to_{\beta} Q$, then $|Q| \leq |P|^2$. Hence, straightforward induction shows that $|M_{i,k}| \leq |M_{i,0}|^{2^k}$. Let u be the redex contracted in $M_{i,0} \to_{\beta} M_{i+1,0}$. By Proposition 5.1, the number of residuals of u along any path from the root to a leaf of $M_{i,k}$ is at most 2^k.

Observe that the reduction $M_{i,k} \to_{\beta}^{*} M_{i+1,k}$ is a complete development of $U = u/(M_{i,0 \to_{\beta}^{*} M_{i,k}})$. Then, Lemma 5.2 and the first part of the lemma yield

$$|M_{i+1,k}| \leq (|M_{i,0}|^{2^k})^{2^{2 \cdot 2^k}} = |M_{i,0}|^{2^k \cdot 2^{2^{k+1}}} = |M_{i,0}|^{2^{2^{k+1}+k}}\,.$$

The reduction $M_{i+1,j-1} \to_{\beta}^{*} M_{i+1,j}$ is a complete development of a set of residuals of the single redex contracted in $M_{i,j-1} \to_{\beta} M_{i,j}$, and an innermost development has length bounded from above by the size of $M_{i+1,j-1}$; by the previous item of the lemma, that size is at most $|M_{i,0}|^{2^{2^j+j-1}}$. By the previous parts of the lemma, the length of the entire bottom reduction $M_{i+1,0} \to_{\beta}^{*} M_{i+1,k}$ is then bounded from above by

$$\sum_{j=1}^{k} |M_{i,0}|^{2^{2^j+j-1}} \leq 2 \cdot |M_{i,0}|^{2^{2^k+k-1}} \leq |M_{i,0}|^{2^{2^k+k}}\,,$$

completing the proof. □

5.2 Valley Sizes in λ-Calculus are in $\mathcal{E}_4$

Lemma 5.4. *Consider the following family of peaks (for $l, k \geq 0$):*

$$M_{l,0} \ {}_\beta\!\leftarrow \cdots {}_\beta\!\leftarrow M_{1,0} \ {}_\beta\!\leftarrow M_{0,0} \rightarrow_\beta M_{0,1} \rightarrow_\beta \cdots \rightarrow_\beta M_{0,k}$$

and write $m = |M_{0,0}|$. Then, in the tiling of the peak with complete developments, the length, $\mathtt{bl}\,(l,k,m)$ of the bottom side of the tiling diagram satisfies the following recursion inequality

$$\mathtt{bl}\,(l,k,m) \leq \begin{cases} k & \textit{if } l = 0 \\ m^{2^{2^{\mathtt{bl}(l-1,k,m)}+\mathtt{bl}(l-1,k,m)+l}} & \textit{if } l > 0 \end{cases}$$

Proof. The tiling diagram may be viewed as m versions of the Strip Lemma (horizontal tiling) stacked on top of each other. The result now follows by a simple induction using Lemma 5.3 (observing for $1 \leq i < l$ that the upper left term in the ith copy of the Strip Lemma has size $|M_{i,0}| \leq m^{2^{i-1}}$). □

Theorem 5.5. *There is a function $g(w,n) : \mathbb{N}^2 \longrightarrow \mathbb{N}$ in the fourth level, $\mathcal{E}_4$, of the Grzegorczyk hierarchy such that $\mathrm{vs}_\Lambda(w,n) \leq g(w,n)$.*

Proof. The right-hand side of the recurrence equation of Lemma 5.4 involves composition of addition, multiplication and exponentiation, applied to limited recursion on the function $\mathtt{bl}\,(m,k,w)$ being defined. As addition, multiplication and exponentiation are at the first, second, and third levels of the Grzegorczyk hierarchy, hence a fortiori in $\mathcal{E}_3$, the function $g(w,n) = \mathtt{bl}\,(n,n,m)$ is in $\mathcal{E}_4$. □

We are currently unable to exhibit a λ-term with a peak of size n such that the corresponding valley size is more than singly exponential. The reader should note that while performing *projections* of the reductions in a peak across each other may yield reductions of extreme length, the projections will usually be equivalent to much shorter reductions and will only give rise to 'small' values of $\mathrm{vs}_\Lambda(m,n)$.

6 Conclusion and Conjectures

We have performed the first fundamental study of the size of Church-Rosser diagrams in TRSs and λ-calculus. For orthogonal TRSs, bounds on valleys turn out to be exponential in a constant dependent on the rewrite system, and thus potentially tractable; for non-orthogonal systems, we showed that for every computable total function, there are TRSs with valley sizes majorizing the function. For λ-calculus, we gave an upper bound on valley sizes. Our inability to construct terms that saturate the upper bounds derived in Section 5 suggests that $\mathrm{vs}_\Lambda(m,n)$ may be in $\mathcal{E}_3$. We conjecture that the dependence on term size $|s|$ in the bound given for arbitrary TRSs in Section 3.1 can be removed; we are currently unable to do so. Finally, the question of valley sizes for higher-order rewriting systems must be investigated; bounds for such systems will automatically lead to bounds for deduction systems in first- and higher order logics, as well as for higher-order functional programs.

Acknowledgements. We would like to thank Richard Statman and Andrzej Filinski for providing valuable answers to some of our questions.

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
2. H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics.* North-Holland, rev. edition, 1985.
3. R. de Vrijer. A direct proof of the finite developments theorem. *Journal of Symbolic Logic*, 50(2):339–343, 1985.
4. J. Endrullis, H. Geuvers, and H. Zantema. Degrees of undecidability in term rewriting. In *Proceedings of the 23rd international workshop on Computer Science Logic (CSL 2009)*, volume 5771 of *Lecture Notes in Computer Science*, pages 255–277, 2009.
5. M. Fernandez. *Models of Computation: An Introduction to Computability Theory.* Undergraduate topics in computer science. Springer, 2009.
6. A. Grzegorczyk. Some classes of recursive functions. *Rozpr. Mat.*, 4:1–45, 1953.
7. N. D. Jones. *Computability and Complexity from a Programming Perspective.* The MIT Press, 1997.
8. Z. Khasidashvili. The longest perpetual reductions in orthogonal expression reduction systems. In *Proceedings of the 3rd International Conference on Logical Foundations of Computer Science*, volume 813 of *Lecture Notes in Computer Science*, pages 191–203. Springer-Verlag, 1994.
9. J. W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 1–116. Oxford University Press, 1992.
10. K.-I. Ko and D.-Z. Du. *Theory of Computational Complexity.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, 2000.
11. P. Odifreddi. *Classical Recursion Theory, volume II*, volume 143 of *Studies in Logic and the Foundations of Mathematics.* North-Holland, 1999.
12. C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
13. H. Rogers Jr. *Theory of Recursive Functions and Effective Computability.* The MIT Press, 1987.
14. M. Sipser. *Introduction to the Theory of Computation.* Thomson Course Technology, 2nd edition, 2006.
15. R. Statman. The typed lambda calculus is not elementary recursive. *Theoretical Computer Science*, 9:73–81, 1979.
16. Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 2003.
17. H. Xi. Upper bounds for standardizations and an application. *Journal of Symbolic Logic*, 64(1):291–303, 1999.

A Omitted Definitions, Proofs, and Remarks

A.1 Definitions for λ-Calculus

Definition A.1. *We write $p \parallel q$ if p and q are incomparable in this order.*

The set of one-hole contexts $C[]$ is defined exactly as for TRSs. If the hole in the one-hole context $C[]$ occurs at position q, we write $C[]_q$. The relation $\rightarrow_\beta$ on $\Lambda \times \Lambda$ is given by $P \rightarrow_\beta Q$ if there is a one-hole context $C[]_q$ such that $P = C[(\lambda x.M)\, N]_q$ and $Q = C[M\{N/x\}]_q$.

We say that there is a redex *at position q.*

If $p \in \mathbf{pos}(M)$ and v is the step $C[(\lambda x.M)N]_q \rightarrow_\beta C[M\{N/x\}]_q$, then the set of descendants *of p across v is defined by:*

- *if $p \prec q$, then $p/v = \{p\}$;*
- *if $p = q$, then $p/v = \emptyset$;*
- *if $q \prec p$ and $p = q \cdot 0 \cdot p'$, then $p/v = \{q \cdot p'\}$;*
- *if $q \prec p$ and $p = q \cdot 1 \cdot p'$, then $p/v = q \cdot \{p_x : M|_{p_x} = x\} \cdot p'$.*

If u is a redex at position p, we denote by u/v the set of redexes at positions p/v. The elements of this set are called the residuals *of u.*

A.2 Omitted Proofs

Proof (Lemma 3.7). As R_0 and R_1 are mutually orthogonal, R_0-reductions and R_1-reductions commute, i.e. the following diagram commutes:

$$\begin{tikzcd}
s \arrow[r, "i", "*"'] \arrow[d, "\bar{i}"', "*"] & t \arrow[d, dotted, "*"', "\bar{i}"] \\
t' \arrow[r, dotted, "*", "i"'] & s'
\end{tikzcd}$$

By the conditions of the lemma, for each peak $t \;{}_i^*\!\leftarrow s \rightarrow_i^* t'$, at least one of the two diagrams below commutes

$$\begin{tikzcd}
s \arrow[r, "i", "*"'] \arrow[d, "i"', "*"] & t \arrow[d, dotted, "*"', "i"] \\
t' \arrow[r, dotted, "*", "i"'] & s'
\end{tikzcd}
\qquad
\begin{tikzcd}
s \arrow[r, "i", "*"'] \arrow[d, "i"', "*"] & t \arrow[d, dotted, "*"', "\bar{i}"] \\
t' \arrow[r, dotted, "*", "\bar{i}"'] & s'
\end{tikzcd}$$

Consider the relation $\rightarrow_Q = \rightarrow_0^* \cup \rightarrow_1^*$. Then $\rightarrow_Q^* = (\rightarrow_0 \cup \rightarrow_1)^* = \rightarrow_{0\cup 1}^*$, and it thus suffices to prove $\rightarrow_Q$ confluent. We make a stronger claim from which confluence will follow: $\rightarrow_Q$ has the diamond property. To see this, observe that if $t \;{}_Q\!\leftarrow s \rightarrow_Q t'$, then there are the four possibilities: (i) $s \rightarrow_0^* t$ and $s \rightarrow_0^* t'$, (ii) $s \rightarrow_0^* t$ and $s \rightarrow_1^* t'$, (iii) $s \rightarrow_1^* t$ and $s \rightarrow_0^* t'$, (iv) $s \rightarrow_1^* t$ and $s \rightarrow_1^* t'$. By the assumptions, for each of these four peaks, there is a corresponding valley obtained by either of the three diagrams above. As each of the reductions in the peak is either a $\rightarrow_0^*$- or a $\rightarrow_1^*$-reduction, it is in particular a $\rightarrow_Q$-step; hence, $\rightarrow_Q$ has the diamond property, and $\rightarrow_{0\cup 1}$ thus has the Church-Rosser property. □

Proof (Proposition 5.1). By induction on n:

- $n = 0$. Only a single copy of u occurs in $M_n = M_0$, and the result follows.
- $n = n' + 1$. Let the redex contracted in $M_{n'} \to_\beta M_n$ be $C[(\lambda x.M)N]_q \to_\beta C[M\{N/x\}]_q$ and let $p \in \mathbf{pos}(M_n)$. If $p \preceq q$ or $p \parallel q$, there are at most $2^{n'} < 2^n$ residuals of u above p by the induction hypothesis. If $q \prec p$, then the number of residuals of u above p is bounded by the number of residuals of u above q plus the number of residuals of u encountered in a path through the term $M\{N/x\}$. This number is at most $2^{n'}+2^{n'} = 2\cdot 2^{n'} = 2^{n'+1} = 2^n$ (recall that even though there may be nested copies of N in $M\{N/x\}$, nestings occur by the application symbol of λ-calculus, hence no position in a copy of N is above any position in another copy of N). □

Proof (Lemma 5.2). By induction on i, observing that there are at most $|M|$ redexes in U.

- $i = 0$. Then, for every $p, q \in U$, we have $p \parallel q$. Also, contraction of a single redex can produce a term of size $\leq |M|^2$. Thus, the total size of the term obtained by contracting all redexes in U is $|M| \cdot |M|^2$, and as there are at most $|M|$ positions above or parallel to all of the redexes of U, we obtain a term of size at most $|M| + |M| \cdot |M|^2 \leq |M|^{2^2}$.
- $i = i' + 1$. By the Finite Developments Theorem [2], we may contract the redexes of U in any order we like to obtain the unique final term. In particular, we may contract the redexes in an innermost fashion. So, consider the subterms of M immediately below outermost redexes in U. The induction hypothesis yields that performing an innermost contraction, these subterms reduce to terms of size at most $|M|^{2^{2\cdot i'}}$. Contracting (the residual of) an outermost redex in U after reducing all of the subterms below it can thus yield a term of size at most $(|M|^{2^{2\cdot i'}})^2 = |M|^{2^{2\cdot i'+1}}$.
  There are at most $|M|$ outermost redexes in U, and there are at most $|M|$ positions of M parallel to or above all redexes of U. Hence, the total size of the term obtained after contracting all redexes of U is at most
  $$|M| + |M| \cdot |M|^{2^{2\cdot i'+1}} \leq |M| \cdot |M| \cdot |M|^{2^{2\cdot i'+1}} \leq |M|^{2^{2\cdot i'+2}} = |M|^{2^{2\cdot i}},$$
  concluding the proof. □