A New Path Order for Exponential Time*

Martin Avanzini
Institute of Computer Science
University of Innsbruck, Austria
martin.avanzini@uibk.ac.at

Naohi Eguchi
School of Information Science
Japan Advanced Institute of Science and Technology, Japan
n-eguchi@jaist.ac.jp

Abstract

In this note we present the *Exponential Path Order* EPO*. Inspired by a novel term rewriting characterisation of the exponential time functions FEXP, this order is carefully trimmed so that we believe that compatibility of TRSs implies exponentially bounded runtime complexity. Moreover, the order is complete in the sense that every exponential time function can be expressed by a TRS compatible with an instance of EPO*.

The work on EPO* is still unfinished, but we strongly believe that above mentioned results are correct.

1 Introduction

Bellantoni and Cook [4] define the class $\mathcal{B}$ as the least class of functions containing certain initial functions, and which is closed under the schemes of *safe recursion on notation* and *safe composition*. In this spirit, exactly the *polytime computable functions* FP are generated, i.e., the class $\mathcal{B}$ coincides with the class FP. Unlike previous definitions of FP (for instance, the recursion-theoretic characterisation given by Cobham [5]), the class $\mathcal{B}$ is defined without explicitly referring to any externally imposed resource bounds. Instead, the strength of the recursion scheme is broken by a syntactic separation of arguments positions into *safe* and *normal* ones. To highlight this separation, we write $f(\vec{x};\vec{y})$ instead of $f(\vec{x},\vec{y})$ for normal arguments $\vec{x}$ and safe arguments $\vec{y}$. Suppose functions g, h_0 and h_1 as well as functions h, $\vec{r}$ and $\vec{s}$ are definable in $\mathcal{B}$. Then a new function f is defined either by *safe recursion on notation* via the equations

$$\begin{aligned} f(\epsilon,\vec{x};\vec{y}) &= g(\vec{x};\vec{y}) \\ f(zi,\vec{x};\vec{y}) &= h_i(z,\vec{x};\vec{y},f(z,\vec{x};\vec{y})),\ i\in\{0,1\} \end{aligned} \tag{SNR}$$

or by *safe composition* via the equation

$$f(\vec{x};\vec{y}) = h(\vec{r}(\vec{x};);\vec{s}(\vec{x};\vec{y}))\ . \tag{SNC}$$

The purpose of the separation is to disallow recursion on recursively computed results: The recursion parameter in (SNR) is taken from a normal argument position, whereas the recursively computed result $f(z,\vec{x};\vec{y})$ is substituted into a safe argument position of the stepping function h_i. For instance, it is not possible to define an exponentiation like function exp via the equations

$$\mathsf{double}(\epsilon) = \epsilon \qquad \mathsf{double}(zi) = \mathsf{double}(z)ii \qquad \mathsf{exp}(\epsilon) = 1 \qquad \mathsf{exp}(zi) = \mathsf{double}(\mathsf{exp}(z))\ .$$

Since the function double is defined by recursion, the single argument of double needs to be normal. Consequently, the function double cannot be used in the definition of exp. The additional restrictions on argument positions imposed by scheme (SNC) ensures that safe arguments cannot influence normal ones.

Inspired by the results of Bellantoni and Cook, Arai and the second author define in [1] the class $\mathcal{N}$ as the least class containing the initial functions of $\mathcal{B}$ and that is closed under the (modified) scheme of *safe composition*

$$f(\vec{x};\vec{y}) = h(x_{i_1},\dots,x_{i_k};\vec{s}(\vec{x};\vec{y}))\ , \tag{SNC$_2$}$$

* This research is supported by FWF (Austrian Science Fund) projects P20133.

and *safe nested recursion on notation*

$$f(\vec{\epsilon},\vec{x};\vec{y}) = g(\vec{x};\vec{y}) \qquad \text{(SNRN)}$$
$$f(\vec{z},\vec{x};\vec{y}) = h_{\tau(\vec{z})}(\vec{v_1},\vec{x};\vec{y},f(\vec{v_1},\vec{x};\vec{t}_{\tau(\vec{z})}(\vec{v_2},\vec{x};\vec{y},f(\vec{v_2},\vec{x};\vec{y})))) \ .$$

In the recursion scheme, recursion is preformed simultaneously on multiple arguments. The functions $h_{\tau(\vec{z})}$ and $\vec{t}_{\tau(\vec{z})}$ are previously defined functions, chosen in terms of $\tau(\vec{z}) \in \Sigma_0^k$ where k is the length of $\vec{z}$ (Σ_0^k refers to the set of binary strings of length k). Further, $\vec{v_1}$ and $\vec{v_2}$ are unique predecessors of $\vec{z}$ defined in terms of τ, satisfying $\vec{v_1} <_{\mathsf{lex}} \vec{z}$ and $\vec{v_2} <_{\mathsf{lex}} \vec{z}$. Observe that the scheme (SNRN) is a syntactic extension of the scheme (SNR). Note also, that the modification of safe composition is necessary as FEXP is, opposed to FP, *not* closed under composition. For missing details on the definition of the scheme (SNRN) we kindly refer the reader to [1]. Instead, we give an illustrating example. The simplest exponentially growing function definable by safe nested recursion on notation is

$$f(\epsilon;y) = y1 \qquad\qquad f(xi;y) = f(x;f(x;y)) \quad (i = 0,1) \ .$$

Then it can be verified that $f(x;y) = y1^{2^{|x|}}$ where 1^n denotes n times concatenation of the symbol 1. Let FEXP denote the class of *functions computable in exponential time*. In [1] it is proved that the class $\mathcal{N}$ coincides with FEXP [1]. As a consequence, we conclude $f \in \mathsf{FEXP}$ for above defined function f.

Hofbauer has shown that *multiset path orders (MPOs for short)* induce primitive recursive bounds on the length of derivations (see [7]). The schemes of safe recursion on notation suitably tames primitive recursion so that only polytime computable functions are generated. Combining those two observation, Moser and the first author have shown that the separation of safe and normal argument positions can suitable tame MPO so that the induced *innermost runtime complexity* is polynomially bounded (see [2] for the polynomial path order POP*, a restriction of MPO). Here the runtime complexity of a TRS measures the maximal number of rewrite steps as a function in the size of the initial term, where the initial terms are *basic* terms, i.e., of the form $f(t_1,\dots,t_n)$ for constructor terms t_i. More precisely, define the *derivation height* of a terminating term t with respect to a finitely branching and well-founded relation $\to$ as

$$\mathrm{dh}(t,\to) = \max\{\ell \mid t \to t_1 \to \cdots \to t_\ell\} \ .$$

Let $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$ denote the innermost rewrite relation as induced by a TRS $\mathcal{R}$. Then the *innermost runtime complexity* of a terminating TRS $\mathcal{R}$ is defined as

$$\mathrm{rc}^{\mathrm{i}}_{\mathcal{R}}(n) = \max\{\mathrm{dh}(t,\xrightarrow{\mathrm{i}}_{\mathcal{R}}) \mid t \text{ is basic and } |t| \leqslant n\} \ .$$

In this paper we present the *exponential path order* EPO*, a miniaturisation of the *lexicographic path order (LPO for short)*. We hope that the same idea exploited in [2] carries over to this miniaturisation, in the sense that by enforcing the scheme of safe nested recursion on notation the multiple recursive bound on derivation lengths induced by LPOs (see [8]) can be broken down to exponential bounds on innermost derivations, whenever the starting term is basic. Hence we aim for an order that

1. induces exponential bounds on the innermost runtime complexity of compatible TRSs, and
2. that is complete in the sense that every exponential time function is expressible by a TRS compatible with EPO*.

Up to now, we are able to verify the second property. The first property still needs further investigations.

[1] A function may even be defined by safe nested recursion on notation with more than two nested recursive calls. In fact, at least three layers are needed to capture the class FEXP, compare [1]. We avoid this complication to simplify the presentation.

2 Exponential Path Order EPO*

In this section we present the *exponential path order* EPO*. We fix a finite but else arbitrary signature $\mathcal{F}$, partitioned into defined symbols $\mathcal{D}$ and constructors $\mathcal{C}$. We use $\succsim = \succ \uplus \approx$ to denote an *admissible* quasi-precedence, i.e. a quasi-precedence where constructors are minimal. The separation of safe and normal argument positions is taken into account by the notion of *safe mapping*. A safe mapping safe is a function that associates with every n-ary function symbol f the set of *safe argument positions*. For constructors $f \in \mathcal{C}$ we require that all argument positions are safe. The argument positions not included in $\mathsf{safe}(f)$ are called *normal* and denoted by $\mathsf{nrm}(f)$. To simplify the presentation, we write $f(t_{i_1},\dots,t_{i_k};t_{j_1},\dots,t_{j_l})$ for the term $s = f(t_1,\dots,t_n)$ with $\mathsf{safe}(f) = \{i_1,\dots,i_k\}$ and $\mathsf{nrm}(f) = \{j_1,\dots,j_l\}$. We use $\approx_{\mathsf{s}}$ to denote term equivalence as induced by $\succsim$. Moreover, we suppose $\approx_{\mathsf{s}}$ respects the separation of argument positions, compare [3].

The *exponential path order* EPO* $>_{\mathsf{epo}\star}$ is based on an auxiliary order $\sqsupset_{\mathsf{epo}\star}$ defined as follows.

$$(1)\quad \frac{s_i \sqsupseteq_{\mathsf{epo}\star} t}{f(s_1,\dots,s_l;s_{l+1},\dots,s_m) \sqsupset_{\mathsf{epo}\star} t}\ f \in \mathcal{C} \text{ and some } 1 \leqslant i \leqslant m$$

$$(2)\quad \frac{s_i \sqsupseteq_{\mathsf{epo}\star} t}{f(s_1,\dots,s_l;s_{l+1},\dots,s_m) \sqsupset_{\mathsf{epo}\star} t}\ f \in \mathcal{D} \text{ and some } 1 \leqslant i \leqslant l$$

Here $\sqsupseteq_{\mathsf{epo}\star} := \sqsupset_{\mathsf{epo}\star} \cup \approx_{\mathsf{s}}$. The split into two orders is necessary, as we must carefully control composition and recursion of functions according to the schemes (SNC_2) and (SNRN). Note that due to the restrictive definition of case (2), one can show $f(\vec{x};\vec{y}) \sqsupset_{\mathsf{epo}\star} x_i$, but one cannot show $f(\vec{x};\vec{y}) \sqsupset_{\mathsf{epo}\star} y_i$. Based on the auxiliary order $\sqsupset_{\mathsf{epo}\star}$, we define for $s = f(s_1,\dots,s_l;s_{l+1},\dots,s_m)$ the *exponential path order* EPO* $>_{\mathsf{epo}\star}$ as follows.

$$(1)\quad \frac{s_i \geq_{\mathsf{epo}\star} t}{f(s_1,\dots,s_l;s_{l+1},\dots,s_m) >_{\mathsf{epo}\star} t}\ \text{for some } 1 \leqslant i \leqslant m$$

$$(2)\quad \frac{s \sqsupset_{\mathsf{epo}\star} t_1 \ \cdots\ s \sqsupset_{\mathsf{epo}\star} t_k \quad s >_{\mathsf{epo}\star} t_{k+1} \ \cdots\ s >_{\mathsf{epo}\star} t_n}{f(s_1,\dots,s_l;s_{l+1},\dots,s_m) >_{\mathsf{epo}\star} g(t_1,\dots,t_k;t_{k+1},\dots,t_n)}\ \text{for } f \succ g$$

$$(3)\quad \frac{s_1 = t_1 \ \cdots\ s_1 = t_{i-1} \quad s_i \sqsupset_{\mathsf{epo}\star} t_i \quad s \sqsupset_{\mathsf{epo}\star} t_{i+1} \ \cdots\ s \sqsupset_{\mathsf{epo}\star} t_l \quad s >_{\mathsf{epo}\star} t_{k+1} \ \cdots\ s >_{\mathsf{epo}\star} t_n}{f(s_1,\dots,s_l;s_{l+1},\dots,s_m) >_{\mathsf{epo}\star} g(t_1,\dots,t_k;t_{k+1},\dots,t_n)}$$

for $f \approx g$ and some $1 \leqslant i \leqslant \min(l,k)$

It is easy to see that $\sqsupset_{\mathsf{epo}\star} \subseteq >_{\mathsf{epo}\star} \subseteq >_{\mathsf{lpo}}$, hence compatibility of a TRS with $>_{\mathsf{epo}\star}$ implies termination, and moreover, a multiply recursive bound on the length of derivations [8]. Note that in order to show $s >_{\mathsf{epo}\star} g(t_1,\dots,t_k;t_{k+1},\dots,t_n)$ by case (2), we need to prove $s \sqsupset_{\mathsf{epo}\star} t_i$ for $1 \leqslant i \leqslant k$ instead of $s >_{\mathsf{epo}\star} t_i$. By the above observation on $>_{\mathsf{epo}\star}$, we can only compare normal arguments of s with t_i. This is in accordance with the scheme (SNC_2).

In [6], the second author introduces the *exponential path order* EPO, a restriction of LPO that induce exponential bounds on the innermost runtime complexity of TRSs. Although the order is complete for FEXP in principle, its application is very restricted on naturally formulated TRSs. The idea of the current research on EPO* is to lift this limitation. Besides the order EPO, a term rewriting characterisation $\mathcal{R}_{\mathcal{N}}$ of the class FEXP is presented [6]. This characterisation is inspired by the schemes from [1], we use it below to show completeness of EPO*. The scheme of rewrite rules $\mathcal{R}_{\mathcal{N}}$ consists of the rules drawn below. For clarification of this scheme of rewrite rules, we kindly refer the reader to [6]. Binary words are formed from the constructor symbols $\varepsilon, \mathrm{S}_0$ and S_1. The function symbols $\mathrm{O}^{k,l}, \mathrm{I}^{k,l}_r, \mathrm{P}, \mathrm{C}$ correspond to the initial functions of the class $\mathcal{N}$. The function symbols $\mathrm{SUB}[g, i_1, \dots, i_k, \vec{h}]$ are used to denote the function obtained by composing functions g and $\vec{h}$ according to the scheme (SNC_2). Finally, function symbols $\mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)]$ correspond to the functions defined by safe nested recursion on notation in accordance to scheme (SNRN). We highlight the separation of safe and normal argument positions directly in the rules.

$$
\begin{array}{rll}
(1) & \mathrm{O}^{k,l}(\vec{x};\vec{y}) \to \epsilon & \\
(2) & \mathrm{I}_r^{k,l}(\vec{x};\vec{y}) \to x_r & \text{for } 1 \leqslant r \leqslant k \\
(3) & \mathrm{I}_r^{k,l}(\vec{x};\vec{y}) \to y_{r-k} & \text{for } k < r \leqslant l+k \\
(4) & \mathrm{P}(;\epsilon) \to \epsilon & \\
(5) & \mathrm{P}(;\mathrm{S}_i(;x)) \to x & \\
(6) & \mathrm{C}(;\epsilon, y_0, y_1) \to y_0 & \\
(7) & \mathrm{C}(;\mathrm{S}_i(;x), y_0, y_1) \to y_1 & \\
(8) & \mathrm{SUB}[g, i_1, \dots, i_k, \vec{h}](\vec{x};\vec{y}) \to g(x_{i_1}, \dots, x_{i_k}; \vec{h}(\vec{x};\vec{y})) & \\
(9) & \mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)](\vec{\epsilon}, \vec{x};\vec{y}) \to g(\vec{x};\vec{y}) & \\
(10) & \mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)](\mathrm{S}_{i_1}(;z_1), \dots, \mathrm{S}_{i_k}(;z_k), \vec{x};\vec{y}) \to & \\
& \quad h_w(\vec{v_1}, \vec{x};\vec{y}, \mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)](\vec{v_1}, \vec{x};\vec{a})) & \\
& \quad\quad [\vec{t_w}(\vec{v_2}, \vec{x};\vec{y}, \mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)](\vec{v_2}, \vec{x};\vec{b}))/\vec{a}] & \\
& \quad\quad [\vec{s_w}(\vec{v_3}, \vec{x};\vec{y}, \mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)](\vec{v_3}, \vec{x};\vec{y}))/\vec{b}] \quad i_j \in \{0,1\}, & \text{for } 1 \leqslant j \leqslant k
\end{array}
$$

In (10) we use $\vec{v_1}$ and $\vec{v_2}$ for specific predecessors of the arguments to $\mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)]$, compare the scheme ($\mathsf{SNRN}$). By the results of [1], it follows that for each $f \in \mathsf{FEXP}$ there exists a finite restriction $\mathcal{R}_f \subseteq \mathcal{R}_{\mathcal{N}}$ such that $\mathcal{R}_f$ computes the function f (compare [6]).

Theorem 1. *Let $\mathcal{R}_f$ be a finite restriction of $\mathcal{R}_{\mathcal{N}}$. Then $\mathcal{R}_f \subseteq >_{\mathsf{epo}\star}$ for some instance of* EPO*.

Proof. Define $\mathrm{lh}(g)$ for symbol g appearing in $\mathcal{R}_f$ as follows. Set $\mathrm{lh}(g) = 1$ for $g \in \{\mathrm{O}^{k,l}, \mathrm{I}_r^{k,l}, \mathrm{P}, \mathrm{C}\}$. Define

$$\mathrm{lh}(\mathrm{SUB}[g, i_1, \dots, i_k, \vec{h}]) = \max\{\mathrm{lh}(g), \mathrm{lh}(\vec{h})\} + 1$$

and

$$\mathrm{lh}(\mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)]) = \max\{\mathrm{lh}(g), \mathrm{lh}(\vec{t_{w'}}), \mathrm{lh}(\vec{s_{w'}}) \mid w' \in \Sigma_0^k\} + 1 \ .$$

Define the safe mapping safe as indicated by the schemata $\mathcal{R}_{\mathcal{N}}$, and define $f > g$ in the precedence if $\mathrm{lh}(f) > \mathrm{lh}(g)$. Then it can be shown that $R_f \subseteq >_{\mathsf{epo}\star}$. We show the most interesting case, namely we orient the final rule. The general case easily follows from this. For notational reasons, we only consider two levels of nestings, that is we show

$$\mathsf{f}(\mathrm{S}_{i_1}(;z_1), \dots, \mathrm{S}_{i_k}(;z_k), \vec{x};\vec{y}) >_{\mathsf{epo}\star} h_w(\vec{v_1}, \vec{x};\vec{y}, \mathsf{f}(\vec{v_1}, \vec{x}; \vec{t_w}(\vec{v_2}, \vec{x};\vec{y}, \mathsf{f}(\vec{v_2}, \vec{x};\vec{y}))))$$

where f abbreviate $\mathrm{SNRN}[g, h_{w'}, \vec{t_{w'}}, \vec{s_{w'}}\,(w' \in \Sigma_0^k)]$. Set $u := \mathsf{f}(\mathrm{S}_{i_1}(;z_1), \dots, \mathrm{S}_{i_k}(;z_k), \vec{x};\vec{y})$. By one application of rule (1) in the definition of $>_{\mathsf{epo}\star}$ we obtain $u >_{\mathsf{epo}\star} y_i$ for $y_i \in \vec{y}$, similar one application of rule (2) of $\sqsupset_{\mathsf{epo}\star}$ yields $u \sqsupset_{\mathsf{epo}\star} x_i$ for $x_i \in \vec{x}$. Recall that terms $\vec{v_3}$ encode $<_{\mathsf{lex}}$-predecessors of words corresponding to terms $\vec{z}$ (compare the remarks below scheme (SNRN)). From this observation we see that for $\vec{v_3} = v_1, \dots, v_k$ and some $1 \leqslant i \leqslant k$,

$$\mathrm{S}_{i_1}(z_1) = v_1, \dots, \mathrm{S}_{i_{i-1}}(z_{i-1}) = v_{i-1},\ \mathrm{S}_{i_i}(z_i) \sqsupset_{\mathsf{epo}\star} v_i \text{ and } \mathrm{S}_{i_{i+1}}(z_{i+1}) \sqsupseteq_{\mathsf{epo}\star} v_{i+1}, \dots, \mathrm{S}_{i_{i+1}}(z_k) \sqsupseteq_{\mathsf{epo}\star} v_k \ .$$

Thus by one application of rule (3) of $>_{\mathsf{epo}\star}$ we are able to conclude $u >_{\mathsf{epo}\star} \mathsf{f}(\vec{v_2},\vec{x};\vec{y})$. Note that by the above inequalities, also $u \sqsupset_{\mathsf{epo}\star} v_i$ for $1 \leqslant i \leqslant k$, and thus due to rule (2) of $>_{\mathsf{epo}\star}$ we further obtain

$$u >_{\mathsf{epo}\star} \vec{t_w}(\vec{v_2},\vec{x};\vec{y},\mathsf{f}(\vec{v_2},\vec{x};\vec{y}))\ .$$

Carrying over the observations on $\vec{v_2}$ to $\vec{v_1}$, the latter inequality gives us

$$u >_{\mathsf{epo}\star} \mathsf{f}(\vec{v_1},\vec{x};\vec{t_w}(\vec{v_2},\vec{x};\vec{y},\mathsf{f}(\vec{v_2},\vec{x};\vec{y})))$$

by another application of rule (3). We conclude with a final application of rule (2). □

We want to point out that compatibility with EPO* *does not* induce exponential runtime complexity. Consider the TRS $\mathcal{R}$ consisting of the rules

$$\mathsf{d}(;x) \to \mathsf{c}(;x,x) \qquad \mathsf{f}(0;y) \to y \qquad \mathsf{f}(\mathsf{s}(;x);y) \to \mathsf{f}(x;\mathsf{d}(;\mathsf{f}(x;y)))\ .$$

Then $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$ for the precedence $\mathsf{f} > \mathsf{d}$ and safe mapping as indicated in the definition of $\mathcal{R}$. Still, we conjecture the following:

Conjecture 1. *Suppose $\mathcal{R}$ is a constructor TRS compatible with $>_{\mathsf{epo}\star}$. Then the innermost runtime complexity $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}(n)$ is bounded by an exponential.*

Our belief in this conjecture is based on the proof of completeness of exponential path orders EPO as put forward in [6]. Completeness of the order is shown by embedding derivations of finite restrictions of $\mathcal{R}_f \subset \mathcal{R}_{\mathcal{N}}$ into EPO via suitable term interpretations. The definition of the employed term interpretation takes the separation of safe and normal argument positions into account. By the close correspondence between the scheme $\mathcal{R}_{\mathcal{N}}$ with the ordering constraints imposed by $>_{\mathsf{epo}\star}$, compatibility $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$ should give enough information to embed $\mathcal{R}$ derivations into instances of EPO in a similar spirit. Whether this truly holds is subject to further research.

References

[1] Arai, T., Eguchi, N.: A new Function Algebra of EXPTIME Functions by Safe Nested Recursion. TCL 10(4) (2009)

[2] Avanzini, M., Moser, G.: Complexity Analysis by Rewriting. In: Proc. of 9th FLOPS 2008. LNCS, vol. 4989, pp. 130–146. Springer Verlag (2008)

[3] Avanzini, M., Moser, G.: Dependency Pairs and Polynomial Path Orders. In: Proc. of 20th RTA 2009. LNCS, vol. 5595, pp. 48–62. Springer Verlag (2009)

[4] Bellantoni, S., Cook, S.: A new Recursion-Theoretic Characterization of the Polytime Functions. CC 2(2), 97–110 (1992)

[5] Cobham, A.: The Intrinsic Computational Difficulty of Functions. In: Proc. 1964 LMPS 1964. pp. 24–30 (1964)

[6] Eguchi, N.: A Lexicographic Path Order with Slow Growing Derivation Bounds. MLQ 55(2), 212–224 (2009)

[7] Hofbauer, D.: Termination Proofs by Multiset Path Orderings Imply Primitive Recursive Derivation Lengths. TCS 105(1), 129–140 (1992)

[8] Weiermann, A.: Termination Proofs for Term Rewriting Systems with Lexicographic Path Ordering Imply Multiply Recursive Derivation Lengths. TCS 139, 355–362 (1995)