A New Path Order that Induces Tight Polynomial Derivation Lengths* (Extended Abstract)

Martin Avanzini[1], Naohi Eguchi[2] and Georg Moser[1]

[1] Institute of Computer Science, University of Innsbruck, Austria
{martin.avanzini,georg.moser}@uibk.ac.at
[2] Mathematical Institute, Tohoku University, Japan
eguchi@math.tohoku.ac.jp

Abstract. We present a new termination ordering, the *small Polynomial Path Order* (sPOP*). The ordering sPOP* provides a new order-theoretic characterisation of the polynomial time functions. It can be shown that if a directed equational system, i.e., a term rewrite system, is oriented by sPOP* using recursion up to depth d, then the maximal length of (innermost) rewriting sequences in the system starting with terms in a specific form is bounded by a polynomial of degree d. This would open a way to characterise sub-polytime complexity classes.

1 Introduction

S. Bellantoni and S. Cook [4], or D. Leivant [5] independently introduced a notion of predicative recursion as known as *safe recursion* (in [4]) or *tiered recursion* (in [5]). The predicative recursion enables us to represent every polynomial time function as a finite set of equations, and hence as a term rewrite system (TRS for short), in contrast to classical recursion-theoretic characterisations by means of bounded recursion. To the best of the authors' knowledge, term rewriting approaches to computational complexity started with a work [3] by A. Beckmann and A. Weiermann, where a non-trivial closure condition of the class of the polytime functions was shown. This was made possible by analysing the maximal lengths of rewriting sequences, which is known as *derivational complexity*, in the TRSs induced by equations defining polytime functions. An immediate motivation of this work comes from a work [2] by the first and third authors, where the term-rewriting characterisation of the polytime functions in [3] is generalised by a termination ordering, the Polynomial Path Order (POP* for short). We present a syntactic restriction sPOP* of POP* that was introduced in a technical report [1] by the authors. Technically the ordering sPOP* is a miniaturisation of the multiset path order. The advantage of sPOP* compared to POP* is that we can read off degrees of polynomial bounds on derivational complexity of TRSs compatible with sPOP*.

* The first and third authors are partially supported by FWF (Austrian Science Fund) project I-608-N18 and by a grant of the University of Innsbruck. The second author is supported by a grant from the John Templeton Foundation for the project *Philosophical Frontiers in Reverse Mathematics*.

2 Main Definitions and Motivating Examples

Let $\mathcal{R}$ be a TRS. The function symbols of $\mathcal{R}$ are partitioned into *defined* and *constructor* ones. We call $\mathcal{R}$ a *constructor TRS* if the arguments of left-hand side of every rewriting rule in $\mathcal{R}$ only contain constructor symbols (and variables). We assume that any TRS contains at least one constructor constant symbol like $\mathsf{0}$ or ϵ. We also assume that the arguments of every function symbol are separated into *normal* and *safe* ones. We write $f(t_1, \dots, t_k\,; t_{k+1}, \dots, t_{k+l})$ to denote a term with k normal and l safe arguments. To precisely assess the complexity of $\mathcal{R}$, the ordering sPOP* allows recursive definitions only on a subset of defined symbols, the so called *recursive symbols*. Symbols that are not recursive are called *compositional*. Fix a (quasi-)precedence $\succcurlyeq$ on the symbols of $\mathcal{R}$. We require that the equivalence $\sim$ underlying $\succcurlyeq$ respects the separation of constructors, recursive and compositional symbols. We denote by $\stackrel{s}{\sim}$ the restriction of term equivalence $\sim$ that takes the partitioning of argument positions into account, cf. [1] for the precise definition. The following definition introduces small polynomial path orders $>_{\mathsf{spop*}}$. Below we set $\geqslant_{\mathsf{spop*}} := \stackrel{s}{\sim} \cup >_{\mathsf{spop*}}$ and also write $>_{\mathsf{spop*}}$ for the product extension of $>_{\mathsf{spop*}}$. Let $\boldsymbol{s} = \langle s_1, \dots, s_k \rangle$ and $\boldsymbol{t} = \langle t_1, \dots, t_k \rangle$. Then $\boldsymbol{s} \geqslant_{\mathsf{spop*}} \boldsymbol{t}$ if $s_i \geqslant_{\mathsf{spop*}} t_i$ for all $i = 1, \dots, k$ and $\boldsymbol{s} >_{\mathsf{spop*}} \boldsymbol{t}$ if $\boldsymbol{s} \geqslant_{\mathsf{spop*}} \boldsymbol{t}$ and $s_{i_0} >_{\mathsf{spop*}} t_{i_0}$ for some $i_0 \in \{1, \dots, k\}$. We write $f(s_1, \dots, s_k\,; s_{k+1}, \dots, s_{k+l}) \rhd_{\mathsf{n}} t$ if t is a subterm of s_i (modulo $\stackrel{s}{\sim}$) for some $i \in \{1, \dots, k\}$.

Definition 1. *Let $s = f(s_1, \dots, s_k\,; s_{k+1}, \dots, s_{k+l})$. Then $s >_{\mathsf{spop*}} t$ if one of the following holds.*

1. *$s_i \geqslant_{\mathsf{spop*}} t$ for some $i \in \{1, \dots, k+l\}$.*
2. *f is a defined symbol, $t = g(t_1, \dots, t_m\,; t_{m+1}, \dots, t_{m+n})$ for some $g \succ f$, and the following conditions hold: (i) $s \rhd_{\mathsf{n}} t_j$ for all $j \in \{1, \dots, m\}$, (ii) $s >_{\mathsf{spop*}} t_j$ for all $j \in \{m+1, \dots, m+n\}$, and (iii) at most one argument of t contains defined symbols not below f in the precedence.*
3. *f is a recursive symbol, $t = g(t_1, \dots, t_k\,; t_{k+1}, \dots, t_{k+l})$ for some $g \sim f$, and the following conditions hold: (i) $\langle s_1, \dots, s_k \rangle >_{\mathsf{spop*}} \langle t_{\pi(1)}, \dots, t_{\pi(k)} \rangle$ for some permutation π, (ii) $\langle s_{k+1}, \dots, s_{k+l} \rangle \geqslant_{\mathsf{spop*}} \langle t_{\tau(k+1)}, \dots, t_{\tau(k+l)} \rangle$ for some permutation τ.*

The *depth* $\mathsf{rd}(f)$ *of recursion* of a symbol f is inductively defined in correspondence to the rank of f in the precedence $\succcurlyeq$, but only takes recursive symbols into account: $\mathsf{rd}(f) := 1 + n$ if f is a recursive symbol and otherwise $\mathsf{rd}(f) := n$, where $n = \max\ \{0\} \cup \{\mathsf{rd}(g) \mid f \succ g\}$. Then it can be shown that sPOP* is complete for the polytime functions in the following sense.

Theorem 1 ([1]).

1. *If a TRS $\mathcal{R}$ is compatible with an instance $>_{\mathsf{spop*}}$ of sPOP*, then for any* argument normalised term *$f(t_1, \dots, t_n)$, the maximal length of innermost rewriting sequences starting with $f(t_1, \dots, t_n)$ in $\mathcal{R}$ is bounded by a polynomial of degree $\mathsf{rd}(f)$ in the sum of the sizes of arguments $t_1, \dots, t_n$. Moreover this bound is tight.*

2. *For any polytime function f there exist a confluent constructor TRS $\mathcal{R}$ defining f and a precedence $\succcurlyeq$ such that $\mathcal{R}$ is compatible with the instance $>_{\mathsf{spop*}}$ of sPOP* induced by $\succcurlyeq$.*

As a corollary of Theorem 1.1, if a confluent TRS $\mathcal{R}$ is compatible with sPOP*, then the function defined by $\mathcal{R}$ is polynomial time computable.

For a motivating example, let us consider a TRS $\mathcal{R}_{\mathsf{add}}$ consisting of $+(\mathsf{0}\,;y) \to y$ and $+(\mathsf{s}(\,;x)\,;y) \to \mathsf{s}(+(x\,;y))$. The TRS $\mathcal{R}_{\mathsf{add}}$ defines the addition of natural numbers built from the constructors $\mathsf{0}$ and s. Define a precedence $\succcurlyeq$ by $+ \succ \mathsf{s} \sim \mathsf{0}$. Then it can be verified that the TRS $\mathcal{R}_{\mathsf{add}}$ is compatible with the instance $>_{\mathsf{spop*}}$ induced by $\succcurlyeq$ with the maximal depth 1 of recursion. For another example, consider a TRS $\mathcal{R}_{\mathsf{mul}}$ consisting of $\mathcal{R}_{\mathsf{add}}$, $\times(\mathsf{0}, y\,;) \to \mathsf{0}$ and $\times(\mathsf{s}(\,;x), y\,;) \to +(y\,;\times(x, y\,;))$. The TRS $\mathcal{R}_{\mathsf{mul}}$ defines the multiplication of natural numbers. Define another precedence $\succcurlyeq$ by $\times \succ + \succ \mathsf{s} \sim \mathsf{0}$. Then it can be verified that the TRS $\mathcal{R}_{\mathsf{mul}}$ is compatible with the instance $>_{\mathsf{spop*}}$ induced by $\succcurlyeq$ with the maximal depth 2 of recursion. For more examples we refer the readers to [1].

3 Conclusion

The contribution of this work is introduction of a new termination ordering, the small polynomial path order sPOP*. The ordering sPOP* has been implemented in an open source *Tyrolean Complexity Tool* $\mathsf{T_CT}$[3]. We can read off degrees of polynomial bounds on derivational complexity of TRSs compatible with sPOP*. An important observation is that only recursion may increase degrees of polynomial bounds on derivational complexity of TRSs compatible with sPOP*. Hence this work would open a way to characterise complexity classes that are not closed under composition. We conclude this extended abstract with a conjecture that a function f can be represented as a confluent constructor TRS compatible with sPOP* that make use of recursion up to d if and only if f can be computed by a register machine (operating over binary words) in a step bounded by a polynomial of degree d in the maximal binary length of inputs.

References

1. M. Avanzini, N. Eguchi, and G. Moser. A New Order-theoretic Characterisation of the Polytime Computable Functions. *CoRR*, cs/CC/1201.2553, 2012. Available at `http://www.arxiv.org/`.
2. M. Avanzini and G. Moser. Complexity Analysis by Rewriting. In *Proc. of 9th FLOPS*, volume 4989 of *LNCS*, pages 130–146, 2008.
3. A. Beckmann and A. Weiermann. A Term Rewriting Characterization of the Polytime Functions and Related Complexity Classes. *Arch. Math. Log.*, 36:11–30, 1996.
4. S. Bellantoni and S. Cook. A New Recursion-Theoretic Characterization of the Polytime Functions. *Comput. Complexity*, 2(2):97–110, 1992.
5. D. Leivant. A Foundational Delineation of Computational Feasibility. In *Proc. of 6th LICS*, pages 2–11. IEEE Computer Society, 1991.

[3] Available at `http://cl-informatik.uibk.ac.at/software/tct`.