Formalizing Termination Proofs under Polynomial Quasi-interpretations

Naohi Eguchi*

Department of Mathematics and Informatics, Faculty of Science
Chiba University, Japan
neguchi@g.math.s.chiba-u.ac.jp

Abstract. It is known that (i) programs can be executed in polynomial space if they are compatible with lexicographic path orders (LPOs) and admit polynomial quasi-interpretations (PQIs), and (ii) LPO-termination proofs can be formalized in the Σ^0_2-induction fragment of Peano arithmetic. We show that LPO-termination proofs can be formalized in the second order system U^1_2 of bounded arithmetic if the compatible programs admit PQIs. This together with a well-known characterization of polyspace functions by U^1_2 yields an alternative proof of the fact (i).

1 Introduction

A termination order $\prec$ enables an embedding of every rewriting sequence $t_0 \to_{\mathcal{R}} t_1 \to_{\mathcal{R}} \cdots$ in a compatible term rewrite system $\mathcal{R}$ into a $\prec$-descending sequence $t_0 \succ t_1 \succ \cdots$. In the seminal work [4] it was discussed, depending on the choice of a termination order $\prec$, what mathematical axiom is necessary (and sufficient) to show that such a $\prec$-descending sequence does not form an infinite one. As shown in [4], in case of multiset path orders (MPOs for short), the Σ^0_1-induction fragment of Peano arithmetic (known as $\mathrm{I}\Sigma_1$) is optimal in a certain sense whereas, in case of lexicographic path orders (LPOs for short), the Σ^0_2-induction fragment (i.e. $\mathrm{I}\Sigma_2$) is optimal. In more recent works [2,7,3], these termination orders are combined with *polynomial quasi-interpretations* (PQIs for short) mainly to deduce the polynomial space complexity. It seems natural to ask in what formal systems termination proofs by MPOs or LPOs under PQIs can be optimally performed.

We present an answer in case of LPOs. It is shown that, given a rewrite system $\mathcal{R}$ under natural constraints, if $\mathcal{R}$ is compatible with an LPO and $\mathcal{R}$ admits a PQI, then an $\mathcal{R}$-normal form of an arbitrary basic term can be found in a second order system U^1_2 of bounded arithmetic. As a consequence, every function defined by such a rewrite system is computable in polynomial space. This result is optimal since, as shown in [2, Theorem 1], the set of functions defined by LPO-compatible, polynomially quasi-interpretable rewrite systems is exactly the same as the set of polynomial-space computable functions.

* The author is supported by Grants-in-Aid for JSPS fellows ($25 \cdot 726$).

2 Lexicographic path orders and quasi-interpretations

Let $\mathcal{V}$ be a countably infinite set of variables and $\mathcal{F}$ a finite set of function symbols that is partitioned into a set $\mathcal{D}$ of defined symbols and a set $\mathcal{C}$ of constructors. We write $\mathcal{T}(\mathcal{F},\mathcal{V})$ to denote the set of terms over $\mathcal{F}$ and $\mathcal{T}(\mathcal{F})$ to denote the set of closed terms. In addition we write $\mathcal{B}(\mathcal{F},\mathcal{V})$ to denote the set $\{f(t_1,\ldots,t_k) \mid f \in \mathcal{D} \ \&\ t_1,\ldots,t_k \in \mathcal{T}(\mathcal{C},\mathcal{V})\}$ of *basic* terms and $\mathcal{B}(\mathcal{F})$ to denote the set of closed basic terms. A *constructor* rewrite system consists of rewrite rules $l \to r$ such that $l \in \mathcal{B}(\mathcal{F},\mathcal{V})$. A rewrite system $\mathcal{R}$ is called *quasi-reducible* if for any basic $t \in \mathcal{B}(\mathcal{F})$ there exist a rule $l \to r \in \mathcal{R}$ and a substitution $\theta : \mathcal{V} \to \mathcal{T}(\mathcal{C})$ such that $t = l\theta$ holds. The *size* $\|t\|$ of a term t is defined as $\|x\| = 1$ if $x \in \mathcal{V}$ and $\|f(t_1,\ldots,t_k)\| = 1 + \sum_{j=1}^{k} \|(t_j)\|$ otherwise. Let $<_{\mathcal{F}}$ be a (strict) precedence, a well-founded partial order on $\mathcal{F}$. We always assume that every constructor is $<_{\mathcal{F}}$-minimal. For the *lexicographic path order* (LPO for short) $<_{\mathsf{lpo}}$ induced by $<_{\mathcal{F}}$, we write $s <_{\mathsf{lpo}} t = g(t_1,\ldots,t_l)$ (or equivalently $t >_{\mathsf{lpo}} s$) if one of the following three cases holds.

1. $s \leqslant_{\mathsf{lpo}} t_i$ for some $i \in \{1,\ldots,l\}$.
2. $s = f(s_1,\ldots,s_k)$, $f <_{\mathcal{F}} g \in \mathcal{D}$ and $s_j <_{\mathsf{lpo}} t$ for each $j \in \{1,\ldots,k\}$.
3. $s = g(s_1,\ldots,s_l)$, $g \in \mathcal{D}$ and there exists $i \in \{1,\ldots,l\}$ such that $s_j = t_j$ for each $j \in \{1,\ldots,i-1\}$, $s_i <_{\mathsf{lpo}} t_i$ and $s_j < t$ for each $j \in \{i+1,\ldots,l\}$.

A *quasi-interpretation* $(\!|\cdot|\!)$ for a signature $\mathcal{F}$ is a mapping form $\mathcal{F}$ to functions over naturals fulfilling (i) $(\!|f|\!) : \mathbb{N}^k \to \mathbb{N}$ for each k-ary function symbol $f \in \mathcal{F}$, (ii) $(\!|f|\!)(m_1,\ldots,m_i,\ldots,m_k) \leq (\!|f|\!)(m_1,\ldots,m_i',\ldots,m_k)$ whenever $m_i < m_i'$, (iii) $m_j \leq (\!|f|\!)(m_1,\ldots,m_k)$ for any $j \in \{1,\ldots,k\}$, and (iv) $0 < (\!|f|\!)$ if f is a constant. A quasi-interpretation $(\!|\cdot|\!)$ for a signature $\mathcal{F}$ is extended to closed terms $\mathcal{T}(\mathcal{F})$ by $(\!|f(t_1,\ldots,t_k)|\!) = (\!|f|\!)((\!|t_1|\!),\ldots,(\!|t_k|\!))$. Such an interpretation $(\!|\cdot|\!)$ is called a quasi-interpretation for a rewrite system $\mathcal{R}$ if $(\!|l\theta|\!) \geq (\!|r\theta|\!)$ holds for each rule $l \to r \in \mathcal{R}$ and for any closed substitution θ. A rewrite system $\mathcal{R}$ *admits a (kind* 0*) polynomial quasi-interpretation* (PQI for short) if there exists a quasi-interpretation $(\!|\cdot|\!)$ for $\mathcal{R}$ such that $(\!|f|\!)$ is polynomially bounded for each $f \in \mathcal{F}$, and, for each constructor $c \in \mathcal{C}$, $(\!|c|\!)(m_1,\ldots,m_k) = d + \sum_{j=1}^{k} m_j$ holds for some constant $d > 0$. A rewrite system $\mathcal{R}$ is called an $LPO^{Poly(0)}$ one if (i) there exists an LPO $<_{\mathsf{lpo}}$ such that $l >_{\mathsf{lpo}} r$ holds for each rule $l \to r \in \mathcal{R}$ and (ii) $\mathcal{R}$ admits a kind 0 PQI.

Theorem 1 ([2]). *Every function defined by a constructor $LPO^{Poly(0)}$-rewrite system is computable in polynomial space.*

3 Formalizing LPO-termination proofs under PQIs in $\mathbf{U_2^1}$

We briefly present basics of second order bounded arithmetic following [1]. The precise formulation can be found in [5]. The non-logical language of first order bounded arithmetic consists of 0, S (the successor), $+$, $\cdot$, $|x| = \lceil \log_2(x+1) \rceil$, $\lfloor x/2 \rfloor$ (the division by two), $\#(x,y) = 2^{|x|\cdot|y|}$ (the smash) and $\leq$. In addition to these

usual symbols, we assume that the language contains $\max(x, y)$, which makes no change if an underlying system is sufficiently strong. Quantifiers of the form $(Qx \leq t)$ for some term t are called *bounded* ones. The language of second order bounded arithmetic additionally contains second order variables $X, Y, Z, \ldots$ ranging over sets and the membership relation $\in$. The classes $\Sigma^{\mathrm{b},1}_i$ and $\Pi^{\mathrm{b},1}_i$ ($i \in \mathbb{N}$) of formulas are dually defined as Σ^0_i and Π^0_i but only counting alternations of second order quantifiers. The second order system U^1_2 is axiomatized with a set BASIC of basic axioms, the schema ($\Sigma^{\mathrm{b},1}_1$-PIND) of bit-wise induction for $\Sigma^{\mathrm{b},1}_1$-formulas, $\varphi(0) \wedge \forall x\big(\varphi(\lfloor x/2 \rfloor) \to \varphi(x)\big) \to \forall x \varphi(x)$, and the axiom ($\Sigma^{\mathrm{b},1}_0$-$\mathsf{CA}$) of comprehension for $\Sigma^{\mathrm{b},1}_0$-formulas, $\forall \vec{x}\ \forall \vec{X}\ \exists Y (\forall y \leq t)\big(y \in Y \leftrightarrow \varphi(y, \vec{x}, \vec{X})\big)$.

Theorem 2 ([5]). *A function is $\Sigma^{\mathrm{b},1}_1$-definable in U^1_2 if and only if it is computable in polynomial space.*

In most interesting examples of PQIs, interpreting polynomials consist of $+$, $\cdot$, $\max^k_{j=1} x_j$ together with additional constants. Thus we formalize PQIs limiting interpreting polynomial terms to those built up only from 0, S, $+$, $\cdot$ and max to make the formalization easier. Let $\mathcal{R}$ be a rewrite system admitting a PQI $(\!| \cdot |\!)$, $t = g(t_1, \ldots, t_l) \in \mathcal{B}(\mathcal{F})$, $s \in \mathcal{T}(\mathcal{C}) \cup \mathcal{B}(\mathcal{F})$ and $t \to^*_{\mathcal{R}} s$. If $s \in \mathcal{T}(\mathcal{C})$, then $\|s\| \leq (\!|s|\!) \leq (\!|t|\!)$ holds. If $s = f(s_1, \ldots, s_k) \in \mathcal{B}(\mathcal{F})$, then $\|s_j\| \leq (\!|s_j|\!) \leq (\!|s|\!) \leq (\!|t|\!)$ holds for each $j \in \{1, \ldots, k\}$. On the other hand, there exists a constant d depending only on $\mathcal{C}$ and $(\!|\cdot|\!)$ such that $(\!|t_j|\!) \leq d \cdot \|t_j\|$ holds for each $j \in \{1, \ldots, l\}$. Hence $(\!|t|\!) \leq p(\|t\|)$ holds for some polynomial p. These observations motivate us to introduce a finite restriction $<_\ell$ of $<_{\mathsf{lpo}}$ ($\ell \in \mathbb{N}$) adopting the one $<_\ell$ in [4].

Definition 1. *Let $\mathcal{T}_\ell(\mathcal{C})$ denote a set $\{t \in \mathcal{T}(\mathcal{C}) \mid \|t\| \leq \ell\}$ of constructor terms and $\mathcal{B}_\ell(\mathcal{F})$ a set $\{f(t_1, \ldots, t_k) \in \mathcal{B}(\mathcal{F}) \mid \|t_1\|, \ldots, \|t_k\| \leq \ell\}$ of basic terms. Then we write $s <_\ell t$ if $s <_{\mathsf{lpo}} t$ and additionally $s \in \mathcal{T}_\ell(\mathcal{C}) \cup \mathcal{B}_\ell(\mathcal{F})$.*

To formalize the relation $<_\ell$, we assume that ℓ denotes a term built up from variables and 0 by S, $|\cdot|$, $+$ and $\cdot$. Since ℓ contains no smash $\#$ in particular, $2^{p(\ell)}$ can be regarded as a first order term for any polynomial $p(x)$. Choosing a suitable encoding $\ulcorner \cdot \urcorner$ for terms, $|\ulcorner t \urcorner|$ is polynomially bounded in the size $\|t\|$ of t, and hence $\ulcorner t \urcorner \leq 2^{p(\|t\|)}$ for some polynomial $p(x)$. Therefore any quantifier of the form $(Qs <_\ell t)$ can be treated as a bounded one.

In [4], to show the termination of an LPO-compatible rewrite system $\mathcal{R}$, given a term t, a well-founded tree T rooted at t containing all the possible $<_{\mathsf{lpo}}$-descending chains starting with t is constructed in $\mathrm{I}\Sigma_2$. Unfortunately the same construction does not work in weak systems as U^1_2 due to an exponential gap between the height and the size of T. Instead of constructing such a derivation tree, we construct a *minimal function graph* G (in the sense of [6,7]), or a *cache* in other words, which enables one to avoid such the exponential gap and to simulate a normalizing (innermost) rewriting $t \to^*_{\mathcal{R}} t^*$ by successively replacing a basic subterm s with its normal form s^* such that $\langle s, s^* \rangle \in G$.

Theorem 3. (in U^1_2) *Suppose that $\mathcal{R}$ is a quasi-reducible, constructor $LPO^{Poly(0)}$-rewrite system. Then, for any basic term t, there exists a minimal function graph for t, and hence t has a normal form.*

Proof (Sketch). Suppose that $\mathcal{R}$ is a rewrite system compatible with an LPO $<_{\mathsf{lpo}}$ admitting a PQI $(\!|\cdot|\!)$. Let $<_\ell$ denote a finite restriction of $<_{\mathsf{lpo}}$ and $\psi_\ell(x, y, X)$ a $\Sigma_0^{\mathrm{b},1}$-formula expressing that $X \subseteq (\mathcal{T}_\ell(\mathcal{C}) \cup \mathcal{B}_\ell(\mathcal{F})) \times \mathcal{T}_\ell(\mathcal{C})$ is a set of pairs of terms such that $\langle x, y\rangle \in X$ and one of the following two cases holds.

1. $x \in \mathcal{T}_\ell(\mathcal{C})$ and $\langle x, x\rangle \in X$.
2. $x \in \mathcal{B}_\ell(\mathcal{F})$ and, for any $\langle t, s\rangle \in X$, $\ell \geq (\!|t|\!) \geq (\!|s|\!)$ and $\exists l \to r \in \mathcal{R}$, $\exists \theta : \mathcal{V} \to \mathcal{T}_\ell(\mathcal{C})$, $\exists \langle \langle t_j, s_j\rangle \in X \mid j < \|r\|\rangle$ such that
   - $t = l\theta$, and
   - $s = \big((r\theta[t_0/s_0]) \cdots [t_{\|r\|-1}/s_{\|r\|-1}]\big)$, where $t'[v/u]$ denotes the result of replacing an occurrence of v in t' with u.

Note that a constant number of variables is enough for a fixed $\mathcal{R}$. Thus $\exists \theta : \mathcal{V} \to \mathcal{T}_\ell(\mathcal{C})$ can be regarded as a bounded quantifier. By definition, for any pair $\langle t, s\rangle \in X$, s is an $\mathcal{R}$-normal form of t (by innermost evaluation). We can assume that $\|t\| \leq |\ulcorner t \urcorner|$ holds for any term t. Hence, depending on the PQI $(\!|\cdot|\!)$, we can find a polynomial term $p(x)$ such that $(\!|t|\!) \leq p(|\ulcorner t \urcorner|)$ holds for any $t \in \mathcal{B}(\mathcal{F})$. Thus it suffices to deduce $(\forall x \in \mathcal{B}(\mathcal{F}))\exists y \exists X \psi_{p(|x|)}(x, y, X)$ in U_2^1. □

Since $\exists X \psi_{p(|x|)}(x, y, X)$ is a $\Sigma_1^{\mathrm{b},1}$-formula, Theorem 2 yields a poly-space computable function f that maps (the code $\ulcorner t \urcorner$ of) an arbitrary basic term t to (the code of) an $\mathcal{R}$-normal form of t, yielding (a variant of) Theorem 1.

Corollary 1. *Every function defined by a quasi-reducible, constructor $LPO^{Poly(0)}$-rewrite system is computable in polynomial space.*

4 Conclusion

We presented an idea how to formalize termination proofs for $\mathrm{LPO}^{\mathrm{Poly}(0)}$-rewrite systems in U_2^1. Crucially, instead of derivation trees, minimal function graphs are constructed by transfinite recursion along a finite restriction $<_\ell$ of $<_{\mathsf{lpo}}$.

References

1. Beckmann, A., Buss, S.: Improved Witnessing and Local Improvement Principles for Second-order Bounded Arithmetic. ACM Transactions on Computational Logic 15(1), 2 (2014)
2. Bonfante, G., Marion, J.-Y., Moyen, J.-Y.: On Lexicographic Termination Ordering with Space Bound Certifications. In: Perspectives of System Informatics. Lecture Notes in Computer Science, vol. 2244, pp. 482–493 (2001)
3. Bonfante, G., Marion, J.-Y., Moyen, J.-Y.: Quasi-interpretations - A Way to Control Resources. Theoretical Computer Science 412(25), 2776–2796 (2011)
4. Buchholz, W.: Proof-theoretic Analysis of Termination Proofs. Annals of Pure and Applied Logic 75(1–2), 57–65 (1995)
5. Buss, S.: Bounded Arithmetic. Bibliopolis, Napoli (1986)
6. Jones, N.D.: Computability and Complexity - from a Programming Perspective. Foundations of Computing Series, MIT Press (1997)
7. Marion, J.-Y.: Analysing the Implicit Complexity of Programs. Information and Computation 183(1), 2–18 (2003)