A Characterisation of Definable NP Search Problems in Peano Arithmetic

Arnold Beckmann*

Department of Computer Science
Swansea University
Swansea SA2 8PP, UK
a.beckmann@swansea.ac.uk

Abstract. The complexity class of $\prec$-*bounded local search problems with goals* is introduced for well-orderings $\prec$, and is used to give a characterisation of definable NP search problems in Peano Arithmetic.

1 Introduction

A *search problem* in general is just a binary relation R. The search task is to find, given x as input, some y satisfying $R(x, y)$. Search problems play a special role in complexity theory. Usually, they are ignored, that is, studied through corresponding decision problems. Often this leads to satisfying results, for example when the reduction is given by a natural self-reduction which produces a polynomially equivalent decision problem. However, there are situations where this approach is unsatisfying as the decision problem is not computationally equivalent. This is particularly important if we are concerned with *total search problems*, that is, search problems which satisfy $(\forall x)(\exists y)R(x, y)$.

Total NP *search problems* are those where R is polynomial time computable and polynomially bounded — the latter means that $R(x, y)$ always implies that the length of y is polynomially bounded in the length of x. Johnson, Papadimitriou and Yannakakis [JPY88] have initiated the study of total NP search problems, and in particular identified several natural subclasses of total NP search problems depending on the mathematical principle needed to proof their totality.

The totality of NP search problems, or in general the totality of definable (multi-)functions, is also an important theme in the study of logical theories, like fragments of arithmetic, in particular Bounded Arithmetic. Bounded Arithmetic has been introduced by Buss [Bus86] as first-order theories of arithmetic with a strong connection to computational complexity. These theories can be given as restrictions of Peano Arithmetic in a suitable language. A main goal in the study of Bounded Arithmetic is to give natural descriptions of the class of total search problems / (multi-)functions whose totality can be shown within some theory of

* Supported in part by EPSRC grant EP/D03809X/1, and by a grant from the John Templeton Foundation.

Bounded Arithmetic [Bus86,Kra93,BK94,Pol99]. Recently, some advances have been made in providing characterisations for missing pairs of level of definability and theories of Bounded Arithmetic [KST07,ST07,Pud06,BB08]. In particular, characterisations have been obtained using a machinery which originated from the proof-theoretic study of Peano Arithmetic, using so called *proof notations for continuous cut-elimination* [AB08,BB09].

At this point, it natural to ask whether it is also possible to obtain natural descriptions of those total NP search problems whose totality can be proven in stronger theories than Bounded Arithmetic.[1] The present paper is a first contribution to this programme by studying the definable NP search problems of Peano Arithmetic, and characterising them in terms of a kind of generalised local search problems which we denote *α-bounded local search problems* for $\alpha < \epsilon_0$. Of course, it is not surprising that α ranges over ordinal notations for the ordinal ϵ_0, as ϵ_0 is the well-known proof-theoretic ordinal for Peano Arithmetic, first implicitly established by Gentzen [Gen36] in his consistency proof for Peano Arithmetic.

The next section will briefly introduce Peano Arithmetic in a way suitable for our proof-theoretic investigations. Section 3 defines the search problem classes of α-bounded local search. This is followed in Section 4 by the definition of an ordinal notation system of order-type ϵ_0. Section 5 briefly reviews necessary definitions and results on notations and cut-reduction for Peano Arithmetic from [AB08]. This is followed by the section defining the search problems which come from proofs in Peano Arithmetic, and stating our main result concerning the characterisation of definable NP search problems in terms of α-bounded local search for $\alpha < \epsilon_0$.

2 Peano Arithmetic

Our definition of Peano Arithmetic is based on Bounded Arithmetic, as we want to make use of the machinery developed in [AB08]. Also, we want to obtain in later sections notation systems which have polynomial time computable ingredients, which in particular means that closed terms in the language must evaluate in polynomial time. Thus, allowing symbols for stronger functions than polynomial time computable ones is problematic.

Our proof-theoretic investigations are very much independent of the exact choice of the language. Therefore, we will be very liberal and allow symbols for all polynomial time computable functions. We introduce Bounded Arithmetic very briefly, and in a slightly nonstandard way similar to [AB08]. The reader

[1] This question has also been formulated in a draft of a book by Pavel Pudlák. The author would like to thank Pavel Pudlák for discussing this question during a one week visit of the author at the Mathematical Institute of the Academy of Sciences of the Czech Republic. The author would also like to thank Jan Krajíček, Pavel Pudlák and Neil Thapen for their hospitality during his stay.

interested in the general theory of Bounded Arithmetic is kindly referred to the literature [Bus86].

For $a \in \mathbb{N}$ let $|a|$ denote the length of the binary representation of a. We will use $|\cdot|$ also as a symbol for a unary function in the next definition. This will never lead to confusion.

Definition 1 (Language of Bounded Arithmetic). The language $\mathcal{L}_{\mathrm{BA}}$ of Bounded Arithmetic *contains as nonlogical symbols* $\{=, \leq\}$ *for the binary relation "equality" and "less than or equal", and a symbol for each polynomial time computable function. In particular,* $\mathcal{L}_{\mathrm{BA}}$ *includes a constant* c_a *for* $a \in \mathbb{N}$ *whose interpretation in the standard model* $\mathbb{N}$ *is* $c_a^{\mathbb{N}} = a$*, and unary function symbols* $|\cdot|$ *whose standard interpretation is given by* $|\cdot|^{\mathbb{N}} \colon a \mapsto |a|$*. We will often write* $\underline{a}$ *instead of* c_a*, and* 0 *for* c_0*.*

Atomic formulas *are of the form* $s = t$ *or* $s \leq t$ *where* s *and* t *are terms.* Literals *are expressions of the form* A *or* $\neg A$ *where* A *is an atomic formula. Formulas are build up from literals by means of* $\wedge$, $\vee$, $(\forall x)$, $(\exists x)$*. The* negation $\neg C$ for a formula C *is defined via de Morgan's laws. Negation extends to sets of formulas in the usual way by applying it to their members individually.*

We will use the following abbreviations.

Definition 2. *The expression* $A \rightarrow B$ *denotes* $\neg A \vee B$*. Bounded quantifiers are introduced as follows:* $(\forall x \leq t)A$ *denotes* $(\forall x)(x \leq t \rightarrow A)$*,* $(\exists x \leq t)A$ *denotes* $(\exists x)(x \leq t \wedge A)$*, where* x *may not occur in* t*.*

Definition 3 (Bounded Formulas). *The set of* bounded $\mathcal{L}_{\mathrm{BA}}$-formulas *is the set of* $\mathcal{L}_{\mathrm{BA}}$*-formulas consisting of literals and being closed under* $\wedge$, $\vee$, $(\forall x \leq t)$, $(\exists x \leq t)$*.*

Definition 4. *The set* $\mathrm{s}\Sigma_1^{\mathrm{b}}$ *consists of all literals and all formulas of the form* $(\exists x \leq s)A(x)$ *where* A *is a literal.* A*,* s *and* t *may depend on other variables not mentioned here.*

Definition 5. *As axioms we allow all disjunctions of literals, i.e., all disjunctions* A *of literals such that* A *is true in* $\mathbb{N}$ *under any assignment. Let us denote this set of axioms by* $\overline{\mathrm{BASIC}}$*.*

The set $\overline{\mathrm{BASIC}}$ is not recursive. Although this is nonstandard for usual formulation of Bounded Arithmetic [Bus86], it is quite normal for the type of proof theoretic invistigations we are after, i.e. using notations for infinitary derivations. It comes from the fact that the complexity of the set of axioms (measured by its arithmetic complexity) of a formal system does not influence the complexity of cut-elimination (measured by the ordinal height of infinitary derivation trees) in the corresponding infinitary propositional derivations.

Definition 6. *Let* $\mathrm{Ind}(A, z, t)$ *denote the expression*

$$A_z(0) \wedge (\forall z \leq t)(A \rightarrow A_z(z+1)) \rightarrow A_z(t) \ .$$

Definition 7. *Let* S^1_2 *denote the theory consisting (of universal closures) of formulas in* $\overline{\mathrm{BASIC}}$ *and (of universal closures) of formulas of the form* $\mathrm{Ind}(A, z, |t|)$ *with* $A \in \mathrm{s}\Sigma^{\mathrm{b}}_1$, *$z$ a variable and t an* $\mathcal{L}_{\mathrm{BA}}$*-term.*

Let PA *denote the theory consisting (of universal closures) of formulas in* $\overline{\mathrm{BASIC}}$ *and (of universal closures) of formulas of the form* $\mathrm{Ind}(A, z, t)$ *with A an* $\mathcal{L}_{\mathrm{BA}}$ *formula (not necessarily bounded), z a variable and t an* $\mathcal{L}_{\mathrm{BA}}$*-term.*

Definition 8. *Let* Σ^{b}_1 *be the set of formulas φ such that there exist* $\psi \in \mathrm{s}\Sigma^{\mathrm{b}}_1$ *with* $\mathrm{S}^1_2 \vdash \varphi \leftrightarrow \psi$.

Let Δ^{b}_1 *be the set of formulas φ such that there exist formulas σ, π with* $\sigma, \neg\pi \in \mathrm{s}\Sigma^{\mathrm{b}}_1$ *and* $\mathrm{S}^1_2 \vdash (\varphi \leftrightarrow \sigma) \wedge (\varphi \leftrightarrow \pi)$.

3 Bounded Local Search with Goals

A binary relation $R \subseteq \mathbb{N} \times \mathbb{N}$ is called *polynomially bounded* iff there is a polynomial p such that $(x, y) \in R$ implies $|y| \leq p(|x|)$. R is called *total* if for all x there exists a y with $(x, y) \in R$.

Definition 9 (Total and Definable NP Search Problems). *Let $R \subseteq \mathbb{N} \times \mathbb{N}$ be a polynomially bounded, total relation which is polynomial time computable. The* (total) NP search problem *associated with R is this: Given input $x \in \mathbb{N}$, return a $y \in \mathbb{N}$ such that $(x, y) \in R$. R is called* definable in a theory T, *if there exists a* $\mathrm{s}\Sigma^{\mathrm{b}}_1$*-formula $(\exists y)\varphi(x, y)$ (the bound to y is implicit in φ) with all free variables shown, such that $(x, y) \in R$ iff $\mathbb{N} \vDash \varphi(x, y)$, and such that $T \vdash (\forall x)(\exists y)\varphi(x, y)$.*

A binary relation $\prec$ on $\mathbb{N} \times \mathbb{N}$ is a *polynomial time computable well-ordering,* if it satisfies the conditions that it is polynomial time computable as a binary relation, that it is a total order, and that it is well-founded, i.e. does not contain infinite descending sequences.

We now define the class of $\prec$-bounded local search problems with goals. It will be defined similar to *polynomial local search (PLS) problems* as introduced by Johnson, Papadimitriou, and Yannakakis [JPY88], and in particular Π^{p}_k-PLS with Π^{p}_ℓ-goals from [BB08,BB09]. The main difference will be that the set of possible solutions is not required to be polynomially bounded. We discuss below immediate consequences of this, after we have given the next definition.

Definition 10 ($\prec$-BLS Problems with Goals). *Let $\prec$ be a polynomial time computable well-ordering. A* $\prec$-bounded local search ($\prec$-bls) problem with goal *is a tuple $L = (S, G, d, N, c, i)$ consisting of, for a given input x, a set $S(x)$ of* possible solutions*, a* goal set *$G(x)$ with a polynomial bound d, a* neighbourhood function *$N(x, s)$ mapping a solution s to another solution, a function $c(x, s)$ computing the* cost of a solution *s according to the well-ordering $\prec$, and a function $i(x)$ computing an* initial solution*, such that the functions N, c and i and the predicates F and G are polynomial time computable, and the following six* *conditions are satisfied:*

$$\prec \textit{ is a total order.} \tag{3.1}$$

$$(\forall x, s)(s \in G(x) \ \rightarrow\ |s| \leq d(|x|)) \tag{3.2}$$

$$(\forall x)(i(x) \in S(x)) \tag{3.3}$$

$$(\forall x, s)(s \in S(x) \ \rightarrow\ N(x, s) \in S(x)) \tag{3.4}$$

$$(\forall x, s)(N(x, s) = s \ \vee\ c(x, N(x, s)) \prec c(x, s)) \tag{3.5}$$

$$(\forall x, s)(s \in G(x) \ \leftrightarrow\ (N(x, s) = s \ \wedge\ s \in S(x))) \tag{3.6}$$

The search task is, for a given input x, to find some s with $s \in G(x)$.

If the well-ordering is understood from the context, we often refer to it by its ordertype given as an ordinal, and e.g. speak of α-bounded local search problems with goals.

We have introduced F and G as sets. They will usually be given via a corresponding relation, e.g. "$s \in S(a)$" in terms of $S(a, s)$.

The following fact is obvious.

Fact 11. *Any $\prec$-bls problem with goal defines a total* NP *search problem in the sense of Definition 9.*

The next observation is almost obvious, and uses the fact that the set of possible solutions is not necessarily polynomially bounded.

Observation 12. *Any total* NP *search problem can be defined by some $<$-bls problem with goal, where $<$ is the natural ordering on $\mathbb{N}$.*

Proof. The proof is based on a simple padding idea. As the set of possible solutions is not required to be polynomially bounded, we first increase the size of a possible solution to reach a possible solution which is exponentially bigger that the polynomially bound of our goal set. At this point it is feasible to directly search for a solution in the goal set.

To be more precise, let R be a total binary relation, which is polynomially bounded using some polynomial d. We define a $<$-bls problem with goal $L = (S, G, d, N, c, i)$ which defines the NP search problem associated with R in the sense of Definition 10: let $b := 2^{d(|x|)}$ (which implies $d(|x|) < |b|$) and define $G(x) := \{y \colon |y| \leq d(|x|) \text{ and } R(x, y)\}$, $S(x) := \{\langle x, b, n\rangle : n \in \mathbb{N}\} \cup G(x)$, $N(x, \langle x, b, n\rangle) := \left\langle x, b, n^2 + 2\right\rangle$ if $|n| < b$, $N(x, \langle x, b, n\rangle) := y$ if $|n| \geq b$ and y smallest with $R(x, y)$ (observe that in this case $y < b \leq |n| \leq |\langle x, b, n\rangle|$, thus it is feasible to search for y,) $N(x, s) := s$ otherwise, $i(x) := \langle x, b, 0\rangle$ and $c(x, \langle x, b, n\rangle) := 1 + (b \dot{-} |n|)$, $c(x, s) := 0$ otherwise. $\square$

The previous fact and observation show that the general formulation of $\prec$-bls problems with goals cannot be used to make any meaningful assertions about total NP search problems. That is, they do not lead to a meaningful combinatorial description of a kind of local search problem, which expresses the totality of the overall search problem in some natural way. If we study the previous proof we can see why this is the case: in order to obtain that the neighbourhood function as defined in the previous proof is a well-defined function (that is, is total) we have to know for the step $N(x, \langle x, b, n\rangle) := y$ if $|n| \geq b$ and y smallest with $R(x, y)$, that a y with $R(x, y)$ exists, which means that at this point we already have to invest that the R defines a total NP search problem. And for the *proof* of existence it does not help that n is very big.

One way to ensure that the description of a $\prec$-bounded local search problem stays in some sense "purely combinatorial", is to require that all its conditions can be formalised in some weak theory suitable for formalising combinatorics. We follow this line of thought in the following definition by taking as such theory S^1_2.

Definition 13 (Formalised $\prec$-BLS Problems in S^1_2). *A $\prec$-bls problem with goal in the sense of Definition 10 is* formalised in S^1_2 *provided the predicates S, G and $\prec$ are given by Δ^{b}_1-formulas, and the defining conditions* (3.1)–(3.6) *are provable in S^1_2.*

4 Ordinal Notations for ϵ_0

Let $<$ denote the 'real' semantic concept of ordinal orderings. Recall the Cantor normal form for ordinals; i.e., every ordinal $\alpha > 0$ can be written uniquely in the form

$$\alpha = \omega^{\alpha_1} + \omega^{\alpha_2} + \omega^{\alpha_3} + \cdots + \omega^{\alpha_k},$$

where $k \geq 1$ and $\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \cdots \geq \alpha_k$. This is the basis for the well-known representation of ordinals less than ϵ_0: namely, write an ordinal $\alpha < \epsilon_0$ as a term in Cantor normal form, recursively writing the exponents of ω in the same form. We repeat the definition of compact representations for ordinals less than ϵ_0 as given in [BBP03].

Definition 14. *We simultaneously and inductively define a set of expressions, called* normal compact forms for ordinals less than ϵ_0*, and a binary relation $\prec_{\epsilon_0}$ on normal compact forms, as follows, where "=" denotes identity on strings:*

1. *If $\alpha_1, \ldots, \alpha_k$ are normal compact forms, and $n_1, \ldots, n_k \in \mathbb{N} \setminus \{0\}$, then the expression $\omega^{\alpha_1} \cdot n_1 + \cdots + \omega^{\alpha_k} \cdot n_k$ is a normal compact form. For $k = 0$ this is the empty word which we denote by* 0.
2. *$\omega^{\alpha_1} \cdot n_1 + \cdots + \omega^{\alpha_k} \cdot n_k \prec_{\epsilon_0} \omega^{\beta_1} \cdot m_1 + \cdots + \omega^{\beta_\ell} \cdot m_\ell$ holds if and only if there is some i with $0 \leq i \leq \min\{k, \ell\}$, such that $\alpha_j = \beta_j$ and $n_j = m_j$ for $j = 1, \ldots, i$, and one of the following cases is satisfied:*
   (a) either $i = k < \ell$; or
   (b) $i < \min\{k, \ell\}$ and $\alpha_{i+1} \prec_{\epsilon_0} \beta_{i+1}$; or
   (c) $i < \min\{k, \ell\}$, $\alpha_{i+1} = \beta_{i+1}$ and $n_{i+1} < m_{i+1}$.

We also write $\alpha \prec_{\epsilon_0} \epsilon_0$ to indicate that α is a normal compact form.

It can be shown (cf. [BBP03]) that S^1_2 can formalise the notion of normal compact forms by using standard sequence coding methods to define the Gödel number of a normal compact form. We assume that some efficient method of sequence coding is used for Gödel numbers so that the length of the Gödel number of a basic form α is proportional to the number of symbols in α.

In this way, the set of normal compact forms and the relation $\prec_{\epsilon_0}$ can be seen to be polynomial time computable based on their inductive definitions, and that the bounded arithmetic theory S^1_2 can Δ^{b}_1-define the syntactic concepts of normal compact forms and the relation $\prec_{\epsilon_0}$, see [BBP03] for more details.

It is also easy to see that the operations $\alpha, \beta \mapsto \alpha + \beta$ of addition and $\alpha \mapsto 3^\alpha$ of exponentiation to base 3 on ordinals can be represented on normal compact forms by polynomial time computable functions. Also observe that the embedding of $\mathbb{N}$ into normal compact forms, given by $n \mapsto \omega^0 \cdot n$, is polynomial time computable.

Finally, we show that S^1_2 can prove that $\prec_{\epsilon_0}$ is a total ordering on normal compact forms, satisfying transitivity and trichotomy.

Theorem 15. *Let α be a normal compact form. The α-bls problems with goals are definable* NP *search problems in* PA.

Proof. Let $L = (S, G, N, c, i)$ be an α-bls problem with goal. Let x be given. The set $A := \{c(x, s) \colon s \in S(x)\}$ is a non-empty subset of $\{\beta \colon \beta \prec_{\epsilon_0} \alpha\}$ by (3.3) and (3.4), and can be expressed by a Σ_1 formula. PA proves transfinite induction up to $\alpha \prec_{\epsilon_0} \epsilon_0$ for Σ_1 properties [Poh09]. Thus, arguing in PA, we can choose some $c \in A$ which is $\prec_{\epsilon_0}$-minimal. Pick $s \in S(x)$ with $c(x, s) = c$, and let $s' := N(x, s)$. Then $s' \in S(x)$ by (3.4). By construction $c(x, s') \not\prec_{\epsilon_0} c(x, s)$, hence (3.5) shows $s' = N(x, s) = s$. Hence, (3.6) shows $s \in G(x)$. □

5 Notation System for Peano Arithmetic

In [AB08], a general framework has been developed which is suitable to characterise definable search problems / (multi-)functions in Bounded Arithmetic. This framework is based on *notations for propositional proofs*. In principle, the same framework can also be used to characterise the definable NP search problems of Peano Arithmetic. The main difference between the notation system for Bounded Arithmetic and that for Peano Arithmetic is that heights of propositional proofs can become infinite in the case of Peano Arithmetic, and therefore have to be bounded by ordinals.

Due to the lack of space, we will only briefly introduce proof notations, and mainly state the differences between those for Bounded Arithmetic and those needed for Peano Arithmetic. The reader interested in more details is kindly referred to [AB08].

A proof notation system is a set (of proof terms) which is equipped with some functions, most prominently a function computing the last inference $\mathrm{tp}(h)$ of a proof named by some notation h, and a function that, given a notation h and a natural number i computes some notation $h[i]$ for the i'th subproof of the derivation named by h. So, a proof notation completely determines an explicit propositional derivation tree; the tree can be reconstructed by exploring it from its root and determining the inference at each node of the tree.

The cut-reduction operator can be defined on the names for derivation trees. Using continuous cut-elimination, these transformations will be particularly simple on the names; note that, using names, for derivations it makes sense to ask about the complexity of getting the i'th subderivation, or about the size of the name, even if it denotes an infinite object. It has been shown [AB08] that the cut-reduction operator on proof notations can be understood as a polynomial time operation. Continuous normalisation for infinitary propositional proofs has been invented by Mints [Min78,KMS75]. The approach in [AB08] is build on Buchholz' technical very smooth approach to notation systems for continuous cut-elimination [Buc91,Buc97].

In [AB08], a notation system $\mathcal{H}_{\mathrm{BA}}$ has been defined which denotes propositional derivations obtained by translating [Tai68,PW85] Bounded Arithmetic proofs. Applying the machinery of notations for continuous cut-elimination, a notation system $\mathcal{CH}_{\mathrm{BA}}$ of cut-elimination for $\mathcal{H}_{\mathrm{BA}}$ has been obtained which has the property that its implicit descriptions, most notably the functions mentioned above, will be polynomial time computable.

To obtain a similar notation system for Peano Arithmetic we can proceed as follows. Let $\mathcal{F}_{\mathrm{PA}}$ be the set of closed formulas in $\mathcal{L}_{\mathrm{BA}}$. We define the outermost connective function $\mathrm{tp}(f)$ for $f \in \mathcal{F}_{\mathrm{PA}}$ to be $\top$ or $\perp$ for true or false literals, respectively, $\bigwedge$ for universally quantified formulas and conjunctions, and $\bigvee$ for existentially quantified formulas and disjunctions. The sub-formula function $f[n]$ for $f \in \mathcal{F}_{\mathrm{PA}}$ and $n \in \mathbb{N}$ is defined in the obvious way, where for finite conjunctions and disjunctions the last conjunct or disjunct is treated as if it were repeated infinitely often. The rank $\mathrm{rk}(A)$ of a formula A in $\mathcal{F}_{\mathrm{PA}}$ is defined in the usual way measuring its depth: $\mathrm{rk}(A) := 0$ for atomic formulas A, for $A = B \wedge C$ or $A = B \vee C$ let $\mathrm{rk}(A) := 1 + \max\{\mathrm{rk}(B), \mathrm{rk}(C)\}$. If $A = (\forall x)B$ or $A = (\exists x)B$, let $\mathrm{rk}(A) := 1 + \mathrm{rk}(B)$.

As closed terms are evaluated to numbers when translating PA-proofs into propositional ones, notations have to be considered modulo the natural intentional equivalence relation $\approx_{\mathbb{N}}$ which identifies terms with the same value. As our definition of $\mathcal{L}_{\mathrm{BA}}$ only contains function symbols for polynomial time computable functions, $\approx_{\mathbb{N}}$ will be polynomial time decidable if the depth of expressions is restricted, and the number of function symbols representing polynomial time functions is also restricted to a finite subset.

Let PA^∞ denote the propositional proof system over $\mathcal{F}_{\mathrm{PA}}$. The last inference of a derivation in PA^∞ can be of the form (Ax_A) for $A \in \mathcal{F}_{\mathrm{PA}}$ with $\mathrm{tp}(A) = \top$ indicating an axiom, $(\bigwedge_C)$ for $C \in \mathcal{F}_{\mathrm{PA}}$ with $\mathrm{tp}(C) = \bigwedge$ indication an application of a $\bigwedge$-inference with main formula C, $(\bigvee_C^i)$ for $C \in \mathcal{F}_{\mathrm{PA}}$ with $\mathrm{tp}(C) = \bigvee$ and $i \in \mathbb{N}$ indicating an application of a $\bigvee$-inference with main formula C and side formula $C[i]$, (Cut_C) for $C \in \mathcal{F}_{\mathrm{PA}}$ with $\mathrm{tp}(C) \in \{\top, \bigwedge\}$ indicating an application of a cut inference, and the void repetition inference (Rep) which neither introduces nor discharges a formula.

The *finitary proof system* $\mathrm{PA}^\star$ is some particularly nice formal proof system for first order logic, which includes also some special rules for induction. It is mainly given by the same inference symbols as $\mathrm{BA}^\star$ in [AB08].

Finally, let $\mathcal{H}_{\mathrm{PA}}$ be the set of closed $\mathrm{PA}^\star$-derivations. For each $h \in \mathcal{H}_{\mathrm{PA}}$ we define the denoted last inference $\mathrm{tp}(h)$ and subderivations $h[j]$ following the obvious translation into propositional logic, were induction up to 2^i is proved by a balanced tree of cuts of height i. The height $\mathrm{o}(h)$ is defined according to the above description of a tree of balanced cuts; the increase of the height caused by one application of induction can be bounded by ω. The cut-rank $\mathrm{crk}(h)$ of a derivation $h \in \mathcal{H}_{\mathrm{PA}}$ is defined as usual by strictly bounding the ranks of all cut-formulas. We write $h \vdash_{\approx_{\mathbb{N}}} \Gamma$ to indicate that Γ is a superset (modulo $\approx_{\mathbb{N}}$) of the end-sequent of the propositional derivation denoted by h.

As for $\mathcal{H}_{\mathrm{BA}}$ [AB08], we can now add notations for cut-elimination to obtain $\mathcal{CH}_{\mathrm{PA}}$. In particular, we add a symbol E which represents the reduction of cuts by one level, and which has the following properties: If $h \vdash_{\approx_{\mathbb{N}}} \Gamma$, then $\mathsf{E}h \vdash_{\approx_{\mathbb{N}}} \Gamma$, $\mathrm{crk}(\mathsf{E}h) \leq \mathrm{crk}(h) \dot{-} 1$ and $\mathrm{o}(\mathsf{E}h) = 3^{\mathrm{o}(h)}$.

As in the case of $\mathcal{H}_{\mathrm{BA}}$ it can be seen that all functions involved in $\mathcal{H}_{\mathrm{PA}}$ and $\mathcal{CH}_{\mathrm{PA}}$ are polynomial-time computable.

Theorem 16. *Assume* $\mathrm{PA} \vdash \varphi$ *with* $\mathrm{FV}(\varphi) \subseteq \{x\}$. *Then, there is some* $\mathrm{PA}^\star$-*derivation* h *such that* $\mathrm{FV}(h) \subseteq \{x\}$, $h \vdash_{\approx_{\mathbb{N}}} \varphi$, *and* $\mathrm{o}(h(\underline{a}/x)) \prec_{\epsilon_0} \omega \cdot 2$.

6 Definable NP Search Problems in Peano Arithmetic

We start by describing the idea for computing witnesses using proof trees. Assume we have a PA-proof of a formula $(\exists y)\varphi(y)$ in $\mathrm{s}\Sigma^{\mathrm{b}}_1$ and we want to compute an n such that $\varphi(n)$ is true — in case we are interested in definable search problems, such a situation is obtained from a proof of $(\forall x)(\exists y)\varphi(x,y)$ by inverting the universal quantifier to some $a \in \mathbb{N}$. Assume further, that we have applied cut-elimination to obtain a PA^∞ derivation d_0 of $(\exists y)\varphi(y)$ with $\mathrm{crk}(d_0) = 0$. Then we can define a path through d_0, represented by sub-derivations $d_1, d_2, d_3 \dots$, such that d_j is an immediate sub-derivation of d_{j+1}, and the end-sequent of d_j is of the form $(\exists y)\varphi(y), \Gamma_j$ where all formulas $A \in \Gamma_j$ are false and either atomic or instances of sub-formulas of $(\exists y)\varphi(y)$. Such a path must be finite as the height of d_j is strictly decreasing. Say it ends with some d_ℓ. In this situation we must have that last inference of d_ℓ is $\bigvee^k_{(\exists y)\varphi(y)}$ and that $\varphi(k)$ is true. Hence we found our witness.

Before we capture this idea in Definition 18 we will define a function on proof notations that computes the next step in the path described above.

Definition 17. *Let* $\mathcal{CH}^k_{\mathrm{PA}}$ *denote the set of notations* h *in* $\mathcal{CH}_{\mathrm{PA}}$ *which satisfy that the index of any function symbols occurring in* h *is bounded by* k, *and that the depths of any formula or term occurring in* h *is also bounded by* k *(the depth of constants is counted as* 0.*)*

We define a function $\mathrm{red}\colon \mathcal{CH}^k_{\mathrm{PA}} \cup \{0\} \to \mathcal{CH}^k_{\mathrm{PA}} \cup \{0\}$ *by* $h \mapsto \mathrm{red}(h)$ *with*

$$\mathrm{red}(h) := \begin{cases} 0 & \text{if } h = 0 \text{ or } \mathrm{tp}(h) = \mathrm{Ax}_A \text{ or} \\ & \mathrm{tp}(h) = \bigvee^d_{(\exists y)\varphi(\underline{a},y)} \text{ with } \varphi(\underline{a},\underline{d}) \text{ true,} \\ h[1] & \text{if } \mathrm{tp}(h) = \mathrm{Cut}_C \text{ with } C \text{ true,} \\ h[1] & \text{if } \mathrm{tp}(h) = \bigwedge_{A_0 \wedge A_1} \text{ with } A_0 \wedge A_1 \text{ of the form} \\ & \varphi(\underline{a},\underline{d}) \text{ for some } d \text{ and } A_0 \text{ true,} \\ h[0] & \text{otherwise.} \end{cases}$$

It is clear from the introduction of proof notations for PA that red is polynomial time computable.

Definition 18. *We define a* parameterised α-bounded local search problem *by* $k \in \mathbb{N}$, $\alpha \prec_{\epsilon_0} \epsilon_0$, *a* $\mathrm{PA}^\star$*-derivation* h *which defines an* initial value function

$$h(\cdot)\colon \mathbb{N} \to \mathcal{CH}_{\mathrm{PA}},\ a \mapsto h(a) := \underbrace{\mathsf{E} \cdots \mathsf{E}}_{\mathrm{crk}(h)\times} h(\underline{a}/x) \enspace ,$$

and a formula $(\exists y)\varphi(x,y) \in \mathrm{s}\Sigma^{\mathrm{b}}_1$*, such that* S^1_2 *proves, for* $a \in \mathbb{N}$*, that* $h(a) \vdash_{\approx_{\mathbb{N}}} (\exists y)\varphi(\underline{a},y)$, $\mathrm{crk}(h(a)) = 0$*, and* $\mathrm{o}(h(a)) \prec_{\epsilon_0} \alpha$*. We denote such a parametrisation by* $P = \langle k, \alpha, h, (\exists y)\varphi(x,y)\rangle$*.*

This parametrisation defines an α*-bounded local search problem with goal* $L = (S, G, d, N, c, i)$ *in the following way: Let* $t(x)$ *be the bound to* y *which is implicit in* $(\exists y)\varphi(x,y)$*. An instance is given by some* $a \in \mathbb{N}$*. The goal set is defined as* $G(a) := \{y\colon \varphi(\underline{a},y)\}$*; the set of possible solutions as*

$$S(a) := G(a) \cup \{\langle t(a), h_0, \ldots, h_\ell\rangle : h_0 = h(a),\ h_\ell \neq 0 \text{ and } (\forall i{<}\ell)\, h_{i+1} = \mathrm{red}(h_i)\} \enspace ;$$

the neighbourhood function is defined as

$$N(a, \langle t(a), h_0, \ldots, h_\ell\rangle) := \begin{cases} \langle t(a), h_0, \ldots, h_\ell, \mathrm{red}(h_\ell)\rangle & \text{if } \mathrm{red}(h_\ell) \neq 0 \\ d & \text{if } \mathrm{red}(h_\ell) = 0 \text{ and } \mathrm{tp}(h_\ell) = \bigvee^d_{(\exists y)\varphi(x,y)} \end{cases}$$

$$N(a, d) := d \qquad \text{for } d \leq t(a) \enspace ;$$

the initial value function *is given by* $i(a) := \langle t(a), h(a)\rangle$*; and the* cost function *is defined as* $c(a, \langle t(a), h_0, \ldots, h_\ell\rangle) := \mathrm{o}(h_\ell)$*, and* $c(a,d) := 0$ *for* $d < D_a$*.*

Proposition 19. *The local search problem* $L = (S, G, d, N, c, i)$ *parameterised by* $P = \langle k, \alpha, h, (\exists y)\varphi(x,y)\rangle$ *from Definition 18 provides an* α*-bls problem with goal according to Definition 10 which solves* φ*.*

Theorem 20. *The definable* NP *search problems in* PA *can be characterised by* α*-bls problems with goals for* $\alpha \prec_{\epsilon_0} \epsilon_0$*.*

Proof. Assume $\mathrm{PA} \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in \mathrm{s}\Sigma^{\mathrm{b}}_1$. Inverting the $(\forall x)$ quantifier we obtain $\mathrm{PA} \vdash (\exists y)\varphi(x,y)$. By Theorem 16, we obtain some $\mathrm{PA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $h \vdash_{\approx_\mathbb{N}} (\exists y)\varphi(x,y)$, and $\mathrm{o}(h(\underline{a}/x)) \prec_{\epsilon_0} \omega \cdot 2$.

Let k be so large that it bounds all indices of function symbols occurring in h, as well as the logical depths of all formulas and terms (where constants have depth 0) occurring in h. Let $\alpha := 3_{\mathrm{crk}(h)}(\omega \cdot 2)$. Then $P = \langle k, \alpha, h, (\exists y)\varphi(x,y)\rangle$ defines a parameterised α-bls problem according to Definition 18, because the following are provable in S^1_2, using $h(a) := \underbrace{\mathsf{E}\cdots\mathsf{E}}_{\mathrm{crk}(h)\times} h(\underline{a}/x)$:

- $h(a) \vdash_{\approx_\mathbb{N}} (\exists y)\varphi(\underline{a}, y)$;
- $\mathrm{crk}(h(a)) = \mathrm{crk}(h(\underline{a}/x)) \dot{-} \mathrm{crk}(h) = \mathrm{crk}(h) \dot{-} \mathrm{crk}(h) = 0$;
- $\mathrm{o}(h(a)) = 3_{\mathrm{crk}(h)}(\mathrm{o}(h(\underline{a}/x))) \prec_{\epsilon_0} 3_{\mathrm{crk}(h)}(\omega \cdot 2) = \alpha$.

By Proposition 19, this defines an α-bls problem with goal which solves φ. □

Together with Theorem 15 we obtain the following

Corollary 21. *The definable* NP *search problems in* PA *are exactly characterised by α-bls problems with goals for $\alpha \prec_{\epsilon_0} \epsilon_0$.*

Conclusion

We have characterised the definable NP search problems of Peano Arithmetic in terms of α-bls problems with goals for $\alpha \prec_{\epsilon_0} \epsilon_0$. One immediate question is whether the defining conditions (3.1)–(3.6) can be turned into some independent principle, by rendering all involved polynomial time functions and predicates in a generic way using oracles (cf. [BB08,BB09]).

Further steps in this programme will be to investigate whether it can be extended to stronger theories than PA. The hope would be that for any theory for which a suitable ordinal analysis has been accomplished [Poh09], this can be turned into some feasible notation system which can form the basis of some class of $\prec$-bls problems characterising the definable NP search problems of that theory. A next step here could be to use the description of Γ_0 in [BBP03].

References

[AB08] Klaus Aehlig and Arnold Beckmann. On the computational complexity of cut-reduction, 2008. Submitted to APAL.

[BB08] Arnold Beckmann and Samuel R. Buss. Polynomial local search in the polynomial hierarchy and witnessing in fragments of bounded arithmetic. Technical Report CSR15-2008, Department of Computer Science, Swansea University, December 2008.

[BB09] Arnold Beckmann and Samuel R. Buss. Characterising definable search problems in bounded arithmetic via proof notations. Technical report, Department of Computer Science, Swansea University, January 2009.

[BBP03] Arnold Beckmann, Samuel R. Buss, and Chris Pollett. Ordinal notations and well-orderings in bounded arithmetic. *Ann. Pure Appl. Logic*, 120(1-3):197–223, 2003.

[BK94] Samuel R. Buss and Jan Krajíček. An application of Boolean complexity to separation problems in bounded arithmetic. *Proc. London Math. Soc. (3)*, 69(1):1–21, 1994.

[Buc91] Wilfried Buchholz. Notation systems for infinitary derivations. *Archive for Mathematical Logic*, 30:277–296, 1991.

[Buc97] Wilfried Buchholz. Explaining Gentzen's consistency proof within infinitary proof theory. In *Computational logic and proof theory (Vienna, 1997)*, volume 1289 of *Lecture Notes in Comput. Sci.*, pages 4–17. Springer, Berlin, 1997.

[Bus86] Samuel R. Buss. *Bounded arithmetic*, volume 3 of *Studies in Proof Theory. Lecture Notes*. Bibliopolis, Naples, 1986.

[Gen36] Gerhard Gentzen. Die Widerspruchsfreiheit der reinen Zahlentheorie. *Math. Ann.*, 112:493–565, 1936.

[JPY88] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. System Sci.*, 37(1):79–100, 1988. 26th IEEE Conference on Foundations of Computer Science (Portland, OR, 1985).

[KMS75] G. Kreisel, G.E. Mints, and S.G. Simpson. The use of abstract language in elementary metamathematics: Some pedagogic examples. In R. Parikh, editor, *Logic Colloquium*, volume 453 of *Lecture Notes in Mathematics*, pages 38–131. Springer, 1975.

[Kra93] Jan Krajíček. Fragments of bounded arithmetic and bounded query classes. *Trans. Amer. Math. Soc.*, 338(2):587–598, 1993.

[KST07] Jan Krajíček, Alan Skelley, and Neil Thapen. NP search problems in low fragments of bounded arithmetic. *J. Symbolic Logic*, 72(2):649–672, 2007.

[Min78] Grigori E. Mints. Finite investigations of transfinite derivations. *Journal of Soviet Mathematics*, 10:548–596, 1978. Translated from: Zap. Nauchn. Semin. LOMI 49 (1975). Cited after Grigori Mints. *Selected papers in Proof Theory*.Studies in Proof Theory. Bibliopolis, 1992.

[Poh09] Wolfram Pohlers. *Proof theory*. Universitext. Springer-Verlag, Berlin, 2009. The first step into impredicativity.

[Pol99] Chris Pollett. Structure and definability in general bounded arithmetic theories. *Ann. Pure Appl. Logic*, 100(1-3):189–245, 1999.

[Pud06] Pavel Pudlák. Consistency and games—in search of new combinatorial principles. In *Logic Colloquium '03*, volume 24 of *Lect. Notes Log.*, pages 244–281. Assoc. Symbol. Logic, La Jolla, CA, 2006.

[PW85] J. Paris and A. Wilkie. Counting problems in bounded arithmetic. In A. Dold and B. Eckmann, editors, *Methods in Mathematical Logic (Proceedings Caracas 1983)*, number 1130 in Lecture Notes in Mathematics, pages 317–340. Springer, 1985.

[ST07] Alan Skelley and Neil Thapen. The provable total search problems of bounded arithmetic, 2007. Typeset manuscript.

[Tai68] William W. Tait. Normal derivability in classical logic. In J. Barwise, editor, *The Syntax and Semantics of Infinitatry Languages*, number 72 in Lecture Notes in Mathematics, pages 204–236. Springer, 1968.