On the computational complexity of cut-reduction

Klaus Aehlig Arnold Beckmann

Department of Computer Science
Swansea University
Singleton Park, Swansea SA2 8PP, UK
E-mail: {k.t.aehlig|a.beckmann}@swansea.ac.uk

Abstract

Using appropriate notation systems for proofs, cut-reduction can often be rendered feasible on these notations, and explicit bounds can be given. Developing a suitable notation system for Bounded Arithmetic, and applying these bounds, all the known results on definable functions of certain such theories can be reobtained in a uniform way.

1 Introduction

Since Gentzen's invention of the "Logik Kalkül" LK and his proof of the "Hauptsatz" [8, 9], cut-elimination has been a topic of almost any paper on proof theory. Mints introduced the concept of continuous cut-elimination [12, 11] by isolating operational aspects of normalisation of (infinitary) propositional derivations. These operational aspects are described at the expense of introducing the void logical rule of repetition to balance derivation trees.

$$\frac{\Gamma}{\Gamma}\ (\mathcal{R})$$

Note that this rule is both logically valid and preserves the sub-formula property.

In this article, we re-examine this situation. We will show that the cut-reduction operator can be understood as a polynomial time operation. We work with proof notations which give implicit descriptions of (infinite) propositional proofs. A proof notation system is a set which is equipped with some functions, most importantly two which compute the following tasks. Given a notation h, compute the last inference $\mathrm{tp}(h)$ in the denoted proof. Given a notation h and a number $i \in \mathbb{N}$, compute a notation $h[i]$ for the i-th immediate sub-derivation of the derivation denoted by h.

Implicit proof notations given in this way uniquely determine a propositional derivation tree, by exploring the derivation tree from its root and determining the inference at each node of the tree. The cut-reduction operator will be defined on such implicitly described derivation trees, following Bucholz' approach [3, 4]. We show (in Corollary 7.6) that the size of the notation for the k-times cut-reduced proof grows only $(k-1)$ times exponentially in the height of original proof.

In the second part of this article we apply these bounds to Bounded Arithmetic. Bounded Arithmetic has been introduced by Buss [5] 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. The restrictions of Peano Arithmetic in S^i_2 are twofold. First, only logarithmic induction is considered which inducts over a logarithmic part of the universe of discourse.

$$\varphi(0) \wedge (\forall x)(\varphi(x) \rightarrow \varphi(x+1)) \rightarrow (\forall x)\varphi(|x|) \ .$$

Here, $|x|$ denotes the length of the binary representation of the natural number x. Second, the properties which can be inducted on, must be described by a suitably restricted bounded formula, more precisely a Σ^{b}_i-formula.

An important goal in Bounded Arithmetic is to give good descriptions of the functions that are definable in a certain theory by a certain class of formulae. Buss [5] has characterised the Σ^{b}_i-definable functions of S^i_2 as $\mathrm{FP}^{\Sigma^p_{i-1}}$, the i-th level of the polynomial time hierarchy of functions. Krajíček [10] has characterised the $\Sigma^{\mathrm{b}}_{i+1}$-definable multi-functions of S^i_2 as the class $\mathrm{FP}^{\Sigma^p_i}[wit, \mathcal{O}(\log n)]$ of multi-functions which can be computed in polynomial time using a witness oracle from Σ^p_i, where the number of oracle queries is restricted to $\mathcal{O}(\log n)$ many (n being the length of the input). Buss and Krajíček [7] have characterised the $\Sigma^{\mathrm{b}}_{i-1}$-definable multi-functions of S^i_2 as projections of solutions to problems from $\mathrm{PLS}^{\Sigma^p_{i-2}}$, which is the class of polynomial local search problems relativised to Σ^p_{i-2}-oracles.

We will re-obtain all these definability characterisations by one unifying method, using the results from the first part of this article. First, we will define a suitable notation system $\mathcal{H}_{\mathrm{BA}}$ for propositional derivations which are obtained by translating [16, 13] Bounded Arithmetic proofs.

Applying the machinery from the first part we obtain a notation system $\mathcal{CH}_{\mathrm{BA}}$ of cut-elimination for $\mathcal{H}_{\mathrm{BA}}$. $\mathcal{CH}_{\mathrm{BA}}$ will have the property that its implicit descriptions, most notably the functions $\mathrm{tp}(h)$ and $h[i]$ mentioned above, will be polynomial time computable. This allows us to formulate a general local search problem on $\mathcal{CH}_{\mathrm{BA}}$ which is suitable to characterise definable multi-functions for Bounded Arithmetic. A more detailled work out of the notation system for Bounded Arithmetic, including full proofs, can be found in a technical report [1].

Other research related to our investigations is an article by Buss [6] which also makes use of the same propositional translation to obtain witnessing results by giving uniform descriptions of translated proofs; however, his approach does not explicitely involve cut-elimination.

The potential of our approach to the characterisations of definable search problems via notation systems is that it may lead to characterisations of so far uncharacterised definable search problems, most notably the Σ^{b}_1-definable search problems in S^i_2 for $i \geq 3$.

2 Proof Systems

Let S be a set. The set of all subsets of S will be denoted by $\mathfrak{P}(S)$, the set of all finite subsets of S will be denoted by $\mathfrak{P}_{\mathrm{fin}}(S)$. Let $\mathcal{F}$ be a set (of *formulae*), $\approx$ a binary relation on $\mathcal{F}$ (*identity between formulae*), and $\mathrm{rk}\colon \mathfrak{P}(\mathcal{F}) \times \mathcal{F} \to \mathbb{N}$ a function (*rank*).

Definition 2.1 (Sequent)**.** A *sequent* over $\mathcal{F}, \approx, \mathrm{rk}$ is a finite subset of $\mathcal{F}$. We use $\Gamma, \Delta, \ldots$ as syntactic variables to denote sequents. With $\approx\Delta$ we denote the set $\{A \in \mathcal{F} \colon (\exists B \in \Delta) A \approx B\}$.

We usually write $A_1, \ldots, A_n$ for $\{A_1, \ldots, A_n\}$ and A, Γ, Δ for $\{A\} \cup \Gamma \cup \Delta$, etc. We always write $\mathcal{C}\text{-rk}(A)$ instead of $\mathrm{rk}(\mathcal{C}, A)$.

We repeat standard Buchholz notation for proof systems [4].

Definition 2.2. A *proof system* $\mathfrak{S}$ *over* $\mathcal{F}, \approx, \mathrm{rk}$ is given by a set of formal expressions called *inference symbols* (syntactic variable $\mathcal{I}$), and for each inference symbol $\mathcal{I}$ an ordinal $|\mathcal{I}| \leq \omega$, a sequent $\Delta(\mathcal{I})$ and a family of sequents $(\Delta_\iota(\mathcal{I}))_{\iota<|\mathcal{I}|}$.

Proof systems may have inference symbols of the form Cut_C for $C \in \mathcal{F}$; these are called "cut inference symbols" and their use will (in Definition 2.4) be measured by the $\mathcal{C}$-cut rank.

Notation 2.3. By writing $(\mathcal{I})\ \dfrac{\ldots \Delta_\iota \ldots (\iota < I)}{\Delta}$ we declare $\mathcal{I}$ as an inference symbol with $|\mathcal{I}| = I$, $\Delta(\mathcal{I}) = \Delta$, $\Delta_\iota(\mathcal{I}) = \Delta_\iota$. If $|\mathcal{I}| = n$ we write $\dfrac{\Delta_0\ \Delta_1\ \ldots\ \Delta_{n-1}}{\Delta}$ instead of $\dfrac{\ldots \Delta_\iota \ldots (\iota < I)}{\Delta}$.

Definition 2.4 (Inductive definition of $\mathfrak{S}$-quasi derivations)**.** If $\mathcal{I}$ is an inference symbol of $\mathfrak{S}$, and $(d_\iota)_{\iota<|\mathcal{I}|}$ is a sequence of $\mathfrak{S}$-quasi derivations, then $d := \mathcal{I}(d_\iota)_{\iota<|\mathcal{I}|}$ is an $\mathfrak{S}$-*quasi derivation* with endsequent

$$\Gamma(d) := \Delta(\mathcal{I}) \cup \bigcup_{\iota<|\mathcal{I}|} (\Gamma(d_\iota) \setminus \approx\Delta_\iota(\mathcal{I})) ,$$

last inference $\mathrm{last}(d) := \mathcal{I}$, subderivations $d(\iota) := d_\iota$ for $\iota < |\mathcal{I}|$, height

$$\mathrm{hgt}(d) := \sup \{\mathrm{hgt}(d_\iota) + 1 \colon \iota < |\mathcal{I}|\} ,$$

size (provided $\mathfrak{S}$ has inference symbols of finite arity only)

$$\mathrm{sz}(d) := (\sum_{\iota<|\mathcal{I}|} \mathrm{sz}(d_\iota)) + 1 ,$$

and cut rank

$$\mathcal{C}\text{-crk}(d) := \sup(\{\mathcal{C}\text{-rk}(\mathcal{I})\} \cup \{\mathcal{C}\text{-crk}(d_\iota) \colon \iota < |\mathcal{I}|\}) .$$

Here we define the cut-rank of $\mathcal{I}$ to be $\mathcal{C}\text{-rk}(C) + 1$ if $\mathcal{I}$ is of the form $\mathcal{I} = \mathrm{Cut}_C$, and 0 otherwise.

3 The infinitary proof system

Definition 3.1. Let $\mathbb{C} = \{\top, \bot, \bigwedge, \bigvee\}$ be the set of (symbols for) connectives for infinitary logic. Their arity is given by $|\top| = |\bot| = 0$ and $|\bigwedge| = |\bigvee| = \omega$. We define a negation of the connectives according to the de Morgan laws: $\neg(\top) = \bot$, $\neg(\bot) = \top$, $\neg(\bigwedge) = \bigvee$, and $\neg(\bigvee) = \bigwedge$.

Definition 3.2. The set of all infinitary formulae $\mathcal{L}_\infty$ is inductively defined by the clause: if $c \in \mathbb{C}$ and $A_\iota \in \mathcal{L}_\infty$ for $\iota < |c|$ then $c(A_\iota)_{\iota<|c|} \in \mathcal{L}_\infty$. Let $\mathcal{C} \subseteq \mathcal{L}_\infty$, then the $\mathcal{C}$-rank of a formula $A = c(A_\iota)_{\iota<|c|} \in \mathcal{L}_\infty$, $\mathcal{C}\text{-rk}(A)$, is defined by recursion on the definition of $A \in \mathcal{L}_\infty$:

$$\mathcal{C}\text{-rk}(A) = \begin{cases} 0 & \text{if } A \in \mathcal{C} \\ \sup_{\iota<|c|}(\mathcal{C}\text{-rk}(A_\iota) + 1) & \text{otherwise.} \end{cases}$$

We denote $\top()$ by $\top$ and $\bot()$ by $\bot$.

Definition 3.3. Negation $\neg$ is the operation on $\mathcal{L}_\infty$ which computes negation according to the de Morgan rules, i.e.,

$$\neg \left(c(A_\iota)_{\iota<|c|}\right) := \neg(c)\left(\neg(A_\iota)\right)_{\iota<|c|}$$

Definition 3.4. The set of all infinitary formulae of finite rank is denoted with $\mathcal{F}_\infty$. The identity between $\mathcal{F}_\infty$-formulae is true equality, i.e., equality of the underlying set theory.

Definition 3.5. The *infinitary proof system* $\mathfrak{S}_\infty$ is the proof system over $\mathcal{F}_\infty$ which is given by the following set of inference symbols.

$$(\bigwedge_A)\ \frac{\dots A_\iota \dots \quad (\iota < \omega)}{A} \text{ for } A = \bigwedge(A_\iota)_{\iota<\omega}$$

$$(\bigvee_A^i)\ \frac{A_i}{A} \text{ for } A = \bigvee(A_\iota)_{\iota<\omega}$$

$$(\mathrm{Ax})\ \frac{}{\top} \qquad (\mathrm{Cut}_C)\ \frac{C \qquad \neg C}{\emptyset} \qquad (\mathrm{Rep})\ \frac{\emptyset}{\emptyset}$$

The $\mathfrak{S}_\infty$-*derivations* are the $\mathfrak{S}_\infty$-quasi derivations.

4 Semiformal proof systems

Definition 4.1. A *notation system for (infinitary) formulae* is a set $\mathcal{F}$ of *formulae*, together with four functions $\mathrm{tp}\colon \mathcal{F} \to \{\top, \bot, \bigwedge, \bigvee\}$, $\cdot[\cdot]\colon \mathcal{F} \times \mathbb{N} \to \mathcal{F}$, $\neg\colon \mathcal{F} \to \mathcal{F}$, and $\mathrm{rk}\colon \mathfrak{P}(\mathcal{F}) \times \mathcal{F} \to \mathbb{N}$ called *outermost connective*, *subformula*, *negation* and *rank*, and a relation $\approx \subseteq \mathcal{F} \times \mathcal{F}$ called *intensional equality*, such that $\mathrm{tp}(\neg(f)) = \neg(\mathrm{tp}(f))$, $\neg(f)[n] = \neg(f[n])$, $\mathcal{C}\text{-rk}(f) = \mathcal{C}\text{-rk}(\neg f)$, $\mathcal{C}\text{-rk}(f[n]) < \mathcal{C}\text{-rk}(f)$ for $n < |\,\mathrm{tp}(f)|$, and $f \approx g$ implies $\mathrm{tp}(f) = \mathrm{tp}(g)$, $f[n] \approx g[n]$, $\neg(f) \approx \neg(g)$ and $\mathcal{C}\text{-rk}(f) = \mathcal{C}\text{-rk}(g)$.

It should be noted that if $\mathcal{F}$ is a notation system for formulae, then so is $\mathcal{F}/\approx$ in the obvious way; moreover, in $\mathcal{F}/\approx$ the intensional equality is true equality in the quotient. The reason why we nevertheless explicitly consider an (intensional) equality relation is that we are interested in the computational complexity of notation systems and therefore prefer to take notations as the strings that arise naturally, rather than working on the quotient.

Definition 4.2. Let $\mathcal{F} = (\mathcal{F}, \mathrm{tp}, \cdot[\cdot], \mathrm{rk}, \approx)$ be a notation system for infinitary formulae. The *interpretation* $[\![f]\!]_\infty$ *of* $f \in \mathcal{F}$ is inductively defined as

$$[\![f]\!]_\infty = \mathrm{tp}(f)([\![f[\iota]]\!]_\infty)_{\iota<|\,\mathrm{tp}(f)|}$$

We note that $f \approx g$ implies $[\![f]\!]_\infty = [\![g]\!]_\infty$, and that for $\mathcal{C} \subseteq \mathcal{F}$ and $\mathcal{C}^\infty := \{[\![f]\!]_\infty \colon f \in \mathcal{C}\}$ we have $\mathcal{C}^\infty\text{-rk}([\![f]\!]_\infty) \le \mathcal{C}\text{-rk}(f)$.

Let $\mathcal{F} = (\mathcal{F}, \mathrm{tp}, \cdot[\cdot], \mathrm{rk}, \approx)$ be a notation system for infinitary formulae.

Definition 4.3. The *semiformal proof system* $\mathfrak{S}_\mathcal{F}$ *over* $\mathcal{F}$ is the proof system over $\mathcal{F}$ which is given by the following set of inference symbols.

$$(\bigwedge_C)\ \frac{\dots C[n] \dots \quad (n \in \mathbb{N})}{C} \qquad (\bigvee_C^i)\ \frac{C[i]}{C}$$

$$(\mathrm{Ax}_A)\ \frac{}{A} \qquad (\mathrm{Cut}_C)\ \frac{C \qquad \neg C}{\emptyset} \qquad (\mathrm{Rep})\ \frac{\emptyset}{\emptyset}$$

The rules Ax_A, $\bigwedge_C$ and $\bigvee_C^i$ require that $\mathrm{tp}(A) = \top$, $\mathrm{tp}(C) = \bigvee$ and $\mathrm{tp}(C) = \bigvee$, respectively.

For Cut_C we require $\mathrm{tp}(C) \in \{\top, \bigwedge\}$. For other C we use Cut_C as an obvious abbreviation for $\mathrm{Cut}_{\neg C}$ with both premises exchanged.

The $\mathfrak{S}_\mathcal{F}$-*derivations* are the $\mathfrak{S}_\mathcal{F}$-quasi derivations.

Later in our applications, we will be concerned only with derivations of finite height, for which we can formulate slightly sharper upper bounds on cut-reduction than in the general (infinite) case (2^α versus 3^α). Thus, from now on we will restrict attention to derivations of finite height only.

Definition 4.4. Let $d \vdash^\alpha_{\mathcal{C},m} \Gamma$ denote that d is an $\mathfrak{S}_\mathcal{F}$-derivation with $\Gamma(d) \subseteq {\approx}\Gamma$, $\mathcal{C}\text{-crk}(d) \le m$, and $\mathrm{hgt}(d) \le \alpha < \omega$.

Definition 4.5. The *interpretation* $[\![d]\!]_\infty$ *of a* $\mathfrak{S}_\mathcal{F}$-*derivation* $d = \mathcal{I}(d_\iota)_{\iota<|\mathcal{I}|}$ is defined as

$$[\![d]\!]_\infty := [\![\mathcal{I}]\!]_\infty([\![d_\iota]\!]_\infty)_{\iota<|\mathcal{I}|}$$

where $[\![\mathcal{I}]\!]_\infty$ is defined by setting the interpretation of Ax_A, $\bigwedge_A$, $\bigvee_A^i$, Cut_C and Rep to be Ax, $\bigwedge_{[\![A]\!]_\infty}$, $\bigvee^i_{[\![A]\!]_\infty}$, $\mathrm{Cut}_{[\![C]\!]_\infty}$, and Rep, respectively.

Induction on d immediately gives $\Gamma([\![d]\!]_\infty) \subseteq [\![\Gamma(d)]\!]_\infty$. The "$\subseteq$", instead of the expected "$=$" is due to the fact, that only formulae are removed from the conclusion that are intensionally equal.

Let $\mathfrak{S}_\mathcal{F}$ the semiformal proof system over $\mathcal{F}$. We define Mints' continuous cut-reduction operator [12, 11] following the description given by Buchholz [3]. The only modification is our explicit use of intensional equality.

Theorem 4.6 (and Definition). *Let* $C \in \mathcal{F}$ *with* $\mathrm{tp}(C) = \bigwedge$*, and* $k < \omega$ *be given. We define an operator* $\mathbb{I}^k_C$ *such that* $d \vdash^\alpha_{\mathcal{C},m} \Gamma, C$ *implies* $\mathbb{I}^k_C(d) \vdash^\alpha_{\mathcal{C},m} \Gamma, C[k]$.

Proof. We argue by the buildup of d. If $\mathrm{last}(d) \in \{\bigwedge_D \colon D \approx C\}$ we set

$$\mathbb{I}^k_C(d) = \mathrm{Rep}(\mathbb{I}^k_C(d(k)))$$

and otherwise we set $\mathbb{I}^k_C(d) := \mathcal{I}(\mathbb{I}^k_C(d(i)))_{i<|\mathcal{I}|}$. ∎

Theorem 4.7 (and Definition). *Let* $C \in \mathcal{F}$ *with* $\mathrm{tp}(C) \in \{\top, \bigwedge\}$ *be given. We define an operator* $\mathbb{R}_C$ *such that for* $\mathcal{C}\text{-rk}(C) \le m$ *we have that* $d_0 \vdash^\alpha_{\mathcal{C},m} \Gamma, C$ *and* $d_1 \vdash^\beta_{\mathcal{C},m} \Gamma, \neg C$ *imply* $\mathbb{R}_C(d_0, d_1) \vdash^{\alpha+\beta}_{\mathcal{C},m} \Gamma$.

Proof. We argue by induction on d_1. Let $\mathcal{I} = \mathrm{last}(d_1)$.

If $\Delta(\mathcal{I}) \cap {\approx}\{\neg C\} \neq \emptyset$. we note that $\mathcal{I}$ has to be of the form $\mathcal{I} = \bigvee^k_D$ for some $k \in \mathbb{N}$ and $D \approx \neg C$. So we can set

$$\mathbb{R}_C(d_0, d_1) = \mathrm{Cut}_{C[k]}(\mathbb{I}^k_C(d_0), \mathbb{R}_C(d_0, d_1(0)))\ .$$

Otherwise we can just set $\mathbb{R}_C(d_0, d_1) = \mathcal{I}(\mathbb{R}_C(d_0, d_1(i)))_{i<|\mathcal{I}|}$ and obtain a derivation as desired. □

Theorem 4.8 (and Definition)**.** *We define an operator* $\mathbb{E}$ *such that:* $d \vdash^{\alpha}_{\mathcal{C},m+1} \Gamma$ *implies* $\mathbb{E}(d) \vdash^{2^{\alpha}-1}_{\mathcal{C},m} \Gamma$.

Proof. We argue by induction on the buildup of d.

If $\mathrm{last}(d) = \mathrm{Cut}_C$ then $\mathcal{C}$-rk$(C) \leq m$ and, without loss of generality, $\mathrm{tp}(C) \in \{\top, \bigwedge\}$. We set

$$\mathbb{E}(d) = \mathrm{Rep}(\mathbb{R}_C(\mathbb{E}(d(0)), \mathbb{E}(d(1))))$$

which is as desired.

Otherwise we set $\mathbb{E}(d) = \mathcal{I}(\mathbb{E}(d(i)))_{i<|\mathcal{I}|}$. □

Immediately from the definition we note that the operators $\mathbb{I}$, $\mathbb{R}$, and $\mathbb{E}$ only inspects the last inference symbol of a derivation to obtain the last inference symbol of the transformed derivation. It should be noted that this continuity would not be possible without the repetition rule.

5 Notations for derivations and cut-elimination

Let $\mathcal{F}$ be a notation system for formulae, and $\mathfrak{S}_{\mathcal{F}}$ the semiformal proof system over $\mathcal{F}$ from Definition 4.3.

Definition 5.1. A *notation system for* $\mathfrak{S}_{\mathcal{F}}$ is a set $\mathcal{H}$ of *notations* and functions $\mathrm{tp}\colon \mathcal{H} \to \mathfrak{S}_{\mathcal{F}}$, $\cdot[\cdot]\colon \mathcal{H} \times \mathbb{N} \to \mathcal{H}$, $\Gamma\colon \mathcal{H} \to \mathfrak{P}_{\mathrm{fin}}(\mathcal{F})$, $\mathrm{crk}\colon \mathfrak{P}(\mathcal{F}) \times \mathcal{H} \to \mathbb{N}$, and $\mathrm{o}, |\cdot|\colon \mathcal{H} \to \mathbb{N} \setminus \{0\}$ called *denoted last inference*, *denoted sub-derivation*, *denoted end-sequent*, *denoted cut-rank*, *denoted height* and *size*, such that $\mathcal{C}$-crk$(h[n]) \leq \mathcal{C}$-crk(h), $\mathrm{tp}(h) = \mathrm{Cut}_C$ implies $\mathcal{C}$-rk$(C) < \mathcal{C}$-crk(h), $\mathrm{o}(h[n]) < \mathrm{o}(h)$ for $n < |\mathrm{tp}(h)|$, and the following local faithfulness property holds for $h \in \mathcal{H}$:

$$\Delta(\mathrm{tp}(h)) \cup \bigcup_{\iota < |\mathrm{tp}(h)|} \Big(\Gamma(h[\iota]) \setminus \approx\Delta_\iota(\mathrm{tp}(h))\Big) \subseteq \approx\Gamma(h) \enspace .$$

The local faithfulness property suffices to ensure the following Proposition.

Proposition 5.2. $\Gamma(h[j]) \subseteq \approx\Big(\Gamma(h) \cup \Delta_j(\mathrm{tp}(h))\Big)$

Definition 5.3. Let $\mathcal{H} = (\mathcal{H}, \mathrm{tp}, \cdot[\cdot], \mathrm{o}, |\cdot|)$ be a notation system for $\mathfrak{S}_{\mathcal{F}}$. The *interpretation* $[\![h]\!]$ *of* $h \in \mathcal{H}$ is inductively defined to be the $\mathfrak{S}_{\mathcal{F}}$-derivation $[\![h]\!] = \mathrm{tp}(h)([\![h[n]]\!])_{n<|\mathrm{tp}(h)|}$.

We note that for $h \in \mathcal{H}$ we have $\mathrm{last}([\![h]\!]) = \mathrm{tp}(h)$, $[\![h]\!](\iota) = [\![h[\iota]]\!]$ for $\iota < |\mathrm{tp}(h)|$, and $\Gamma([\![h]\!]) \subseteq \approx\Gamma(h)$

We now extend a notation system $\mathcal{H}$ for $\mathfrak{S}_{\mathcal{F}}$ to a notation system for cut-elimination on $\mathcal{H}$, by adding notations for the operators $\mathbb{I}$, $\mathbb{R}$ and $\mathbb{E}$ from the previous section.

If $\mathcal{H}$ is a notation system we define a notation system $\mathcal{CH}$ for cut-elimination for $\mathcal{H}$ by extending $\mathcal{H}$ by derivations $\mathsf{I}^k_C h$ for $\mathrm{tp}(C) = \bigwedge$, $\mathsf{R}_C h_0 h_1$ for $\mathrm{tp}(C) \in \{\top, \bigwedge\}$, and $\mathsf{E}h$; in all these cases of this inductive definition the h, h_0, h_1 can be taken from $\mathcal{CH}$.

The functions tp, $\cdot[\cdot]$, Γ, crk and o are defined as to make the new symbols I, R, and E match the operators $\mathbb{I}$, $\mathbb{R}$, and $\mathbb{E}$, respectively. The size $|\cdot|$ is defined in the obvious way, that is, $|\mathsf{I}^k_C h| = |\mathsf{E}h| = |h| + 1$ and $|\mathsf{R}_C h_0 h_1| = |h_0| + |h_1| + 1$.

The *interpretation* $[\![h]\!]$ is extended inductively from $\mathcal{H}$ to $\mathcal{CH}$ by defining $[\![\mathsf{I}^k_C h]\!] = \mathbb{I}^k_C([\![h]\!])$, $[\![\mathsf{R}_C h_0 h_1]\!] = \mathbb{R}_C([\![h_0]\!], [\![h_1]\!])$, and $[\![\mathsf{E}h]\!] = \mathbb{E}([\![h]\!])$.

We note that also for $h \in \mathcal{CH}$ we have $\mathrm{last}([\![h]\!]) = \mathrm{tp}(h)$, $[\![h]\!](\iota) = [\![h[\iota]]\!]$ for $\iota < |\mathrm{tp}(h)|$, and $\mathcal{C}$-crk$([\![h]\!]) \leq \mathcal{C}$-crk$(h)$.

It should be observed that for the computation of Γ, the cut-elimination operators I^k_C, R_C and E behave as if there were the following inference symbols.

$$(\mathsf{I}^k_C)\ \frac{C}{C[k]} \qquad (\mathsf{R}_C)\ \frac{C \qquad \neg C}{\emptyset} \qquad (\mathsf{E})\ \frac{\emptyset}{\emptyset}$$

6 An Abstract Notion of Notation

We are now interested in studying the size needed by the notations for sub-derivations of derivations obtained by the cut-elimination operator. To avoid losing the simple idea in a blurb of notation, we abstract our problem to a simple term-rewriting system.

Definition 6.1. An *abstract system of proof notations* is a set $\mathcal{D}$ of "derivations", together with two functions $|\cdot|, o(\cdot)\colon \mathcal{D} \to \mathbb{N} \setminus \{0\}$, called "size" and "height", and a relation $\to \subseteq \mathcal{D} \times \mathcal{D}$ called "reduction to a sub-derivation", such that $d \to d'$ implies $o(d') < o(d)$.

Observation 6.2 (and Definition)**.** *Let* $\mathcal{F}$ *be a notation system for formulae and* $\mathfrak{S}_{\mathcal{F}}$ *the semiformal proof system over* $\mathcal{F}$*. A notation system* $\mathcal{H} = (\mathcal{H}, \mathrm{tp}, \cdot[\cdot], \mathrm{o}, |\cdot|)$ *for* $\mathfrak{S}_{\mathcal{F}}$ *gives rise to an abstract system of proof notations by letting* $\mathcal{D} = \mathcal{H}$ *and defining* $d \to d'$ *iff there exists an* $n < |\mathrm{tp}(d)|$ *with* $d' = d[n]$.

Definition 6.3. If $\mathcal{D}$ is an abstract system of proof notations, then $\widetilde{\mathcal{D}}$, the "cut elimination closure", is the abstract notation system inductively defined to extend $\mathcal{D}$ and contain derivations $\mathsf{I}d$, $\mathsf{E}d$, and $\mathsf{R}de$ for $d, e \in \widetilde{\mathcal{D}}$. Here I, E and R are new symbols. The size is extended in the obvious way, that is $|\mathsf{I}d| = |\mathsf{E}d| = 1 + |d|$ and $|\mathsf{R}de| = 1 + |d| + |e|$. The height is extended following the properties of the operators I, E, and R. In other words, we set $o(\mathsf{I}d) = o(d)$, $o(\mathsf{R}de) = o(d) + o(e)$, and $o(Ed) = 2^{o(d)} - 1$.

The relation $\to$ is inductively defined as follows.

$$\frac{d \to d' \text{ in } \mathcal{D}}{d \to d'} \qquad \frac{d \to d'}{\mathsf{I}d \to \mathsf{I}d'} \qquad \frac{e \to e'}{\mathsf{R}de \to \mathsf{R}de'}$$

$$\frac{d \to d'}{\mathsf{E}d \to \mathsf{E}d'} \qquad \frac{}{\mathsf{R}de \to \mathsf{I}d} \qquad \frac{d \to d' \qquad d \to d''}{\mathsf{E}d \to \mathsf{R}(\mathsf{E}d')(\mathsf{E}d'')}$$

We immediately note hat $\widetilde{\mathcal{D}}$ is an abstract system of proof notations if $\mathcal{D}$ is one.

Let $\mathcal{F}$ be a notation system for formulae, $\mathfrak{S}_{\mathcal{F}}$ the semi-formal proof system over $\mathcal{F}$, $\mathcal{H}$ a notation system for $\mathfrak{S}_{\mathcal{F}}$, $\mathcal{CH}$ the notation system for cut-elimination on $\mathcal{H}$ with denoted height o and size $|\cdot|$, and let $\mathcal{D}$ be the abstract system of proof notations associated with $\mathcal{H}$ according to Observation 6.2.

Definition 6.4. The abstraction $\overline{h}$ of $h \in \mathcal{CH}$ is obtained by dropping all sub- and superscripts. We denote the set of abstractions for $h \in \mathcal{CH}$ by $\overline{\mathcal{CH}}$.

The set of abstractions $\overline{\mathcal{CH}}$ for $\mathcal{CH}$ is a subsystem of the cut-elimination closure $\widetilde{\mathcal{H}}$ of $\mathcal{H}$ in the following sense. Let $\to$ denote the reduction to sub-derivation relation of $\widetilde{\mathcal{H}}$, and define a reduction to sub-derivation relation $\rightsquigarrow$ of $\overline{\mathcal{CH}}$ in the obvious way by $\overline{h} \rightsquigarrow \overline{h'}$ iff there exists an $n < |\operatorname{tp}(h)|$ with $h' = h[n]$. Then $\overline{\mathcal{CH}} = \widetilde{\mathcal{H}}$ and $\rightsquigarrow \subseteq \to$.

7 Size Bounds

We now prove a bound on the size of (abstract) notations for cut-elimination. By induction on the buildup of $\widetilde{\mathcal{D}}$ we assign every element a measure that bounds the size of all derivations reachable from it via iterated use of the $\to$-relation.

A small problem arises in the base case; if $d \to d'$ in $\widetilde{\mathcal{D}}$ because this holds in $\mathcal{D}$ we have no means of bounding $|d'|$ in terms of $|d|$. So we use the usual trick [2] when a global measure is needed and assign each element d of $\widetilde{\mathcal{D}}$ not a natural number but a monotone function $\vartheta(d)$ such that $|d'| \leq \vartheta(d)(s)$ for all $d \to^* d'$ whenever $s \in \mathbb{N}$ is a global bound on the size of all elements in $\mathcal{D}$.

Definition 7.1. An abstract system $\mathcal{D}$ of proof notations is called s-bounded (for $s \in \mathbb{N}$), if for all $d \in \mathcal{D}$ it is the case that $|d| \leq s$.

If $\mathcal{D}$ is an abstract system of proof notations and $d \in \mathcal{D}$, then by $\mathcal{D}_d$ we denote the set $\mathcal{D}_d = \{d' \mid d \to^* d'\} \subset \mathcal{D}$ considered an abstract system of proof notation with the structure induced by $\mathcal{D}$. Here $\to^*$ denotes the reflexive transitive closure of $\to$.

For $\mathcal{D}$ an abstract system of proof notations and $d \in \mathcal{D}$ we say that d is s-bounded if $\mathcal{D}_d$ is.

Definition 7.2. For $\mathcal{D}$ an abstract system of proof notations we define a *size function* $\vartheta(d)$ for every $d \in \widetilde{\mathcal{D}}$ as a monotone function from $\mathbb{N}$ to $\mathbb{N}$. $\vartheta(d)$ is defined by induction on the inductive definition of $\widetilde{\mathcal{D}}$ as follows.

$$\begin{aligned} \vartheta(d)(s) &= s, \text{ provided } d \in \mathcal{D} \\ \vartheta(\mathsf{I}d)(s) &= \vartheta(d)(s) + 1 \\ \vartheta(\mathsf{R}de)(s) &= \max\{|d|+1+\vartheta(e)(s)\ ,\ \vartheta(d)(s)+1\} \\ \vartheta(\mathsf{E}d)(s) &= o(d)(\vartheta(d)(s) + 2) \end{aligned}$$

Proposition 7.3. *If $\mathcal{D}$ is s-bounded then for every $d \in \widetilde{\mathcal{D}}$ we have $|d| \leq \vartheta(d)(s)$.*

Theorem 7.4. *If $\mathcal{D}$ is s-bounded, $d \in \widetilde{\mathcal{D}}$ and $d \to d'$, then $\vartheta(d)(s) \geq \vartheta(d')(s)$.*

Proof. Induction on the inductive definition of the relation $d \to d'$ in $\widetilde{\mathcal{D}}$. If $d \to d'$ because it holds in $\mathcal{D}$ then $\vartheta(d)(s) = s = \vartheta(d')(s)$.

If $\mathsf{E}d \to \mathsf{R}(\mathsf{E}d')(\mathsf{E}d'')$ thanks to $d \to d'$ and $d \to d''$ we argue as follows

$$\begin{aligned} &\vartheta(\mathsf{R}(\mathsf{E}d')(\mathsf{E}d''))(s) \\ =\ &\max\{|\mathsf{E}d'|+1+\vartheta(\mathsf{E}d'')(s)\ ,\ \vartheta(\mathsf{E}d')(s)+1\} \\ =\ &\max\{\ |d'|+2+o(d'')(\vartheta(d'')(s)+2)\ , \\ &\qquad o(d')(\vartheta(d')(s) + 2)\} \\ \leq\ &\max\{\ \vartheta(d')(s)+2+o(d'')(\vartheta(d'')(s)+2)\ , \\ &\qquad o(d')(\vartheta(d')(s) + 2)\} \\ \leq\ &\max\{\ \vartheta(d)(s)+2+o(d'')(\vartheta(d)(s)+2)\ , \\ &\qquad o(d')(\vartheta(d)(s) + 2)\} \\ \leq\ &\max\{\ \vartheta(d)(s)+2+(o(d) - 1)(\vartheta(d)(s)+2)\ , \\ &\qquad (o(d) - 1)(\vartheta(d)(s) + 2)\} \\ =\ &\vartheta(d)(s)+2+(o(d) - 1)(\vartheta(d)(s)+2) \\ =\ &o(d)(\vartheta(d)(s)+2) \\ =\ &\vartheta(\mathsf{E}d)(s) \end{aligned}$$

where for the first inequality we used Proposition 7.3, for the second the induction hypothesis, for the third that, since $d \to d'$ and $d \to d''$, both $o(d')$ and $o(d'')$ are bounded by $o(d) - 1$.

If $\mathsf{R}de \to \mathsf{R}de'$ thanks to $e \to e'$, then

$$\begin{aligned} &\vartheta(\mathsf{R}de')(s) \\ =\ &\max\{|d|+1+\vartheta(e')(s)\ ,\ \vartheta(d)(s)+1\} \\ \leq\ &\max\{|d|+1+\vartheta(e)(s)\ ,\ \vartheta(d)(s)+1\} \\ =\ &\vartheta(\mathsf{R}de) \end{aligned}$$

where for the inequality we used the induction hypothesis.

The remaining cases are trivial. ∎

Now we draw the desired consequences of our main theorem by putting things together.

Corollary 7.5. *If $\mathcal{D}$ is s-bounded, and $d \in \widetilde{\mathcal{D}}$ then $\widetilde{\mathcal{D}}_d$ is $\vartheta(d)(s)$-bounded.*

Recall that iterated exponentiation $2_n(x)$ is defined inductively by setting $2_0(x) = x$ and $2_{n+1}(x) = 2^{2_n(x)}$. An easy induction shows that the height $o(E^n d)$ of the n-times cut-reduced derivation d is bounded by $2_n(d)$.

Corollary 7.6. *If $d \in \mathcal{D}$ is s-bounded of height $o(d) = h$ for $s \geq 2$ and $h \geq 2$, then $E^k(d)$ is $2_{k-1}(2 \cdot h) \cdot s$-bounded for all $k \geq 1$.*

In Corollary 7.6 one should note that the tower of exponentiations has height only $k - 1$. Hence there is one exponentiation less than the height of the denoted proof.

We conclude this section by remarking that the cut-elimination operator can be viewed as a polynomial time computable operation. Assume we modify the size function on $\widetilde{\mathcal{D}}$ to ϑ_k by changing all ϑ to ϑ_k and defining for the last case to be $\vartheta_k(\mathsf{E}d)(s) = (k+1) \cdot (\vartheta(d)(s) + 2)$.

Then we obtain as before for an s-bounded $\mathcal{D}$, $d \in \widetilde{\mathcal{D}}$ and $k \in \mathbb{N}$, that $|d| \leq \vartheta_k(d)(s)$, and $d \to d'$ implies $\vartheta_{k+1}(d)(s) \geq \vartheta_k(d)(s)$. Hence, for $d \in \mathcal{D}$, $\mathcal{D}$ s-bounded, and $\mathsf{E}d \to^k d'$, we obtain $|d'| \leq \vartheta_k(\mathsf{E}d)(s) \leq (k+1) \cdot (s + 2)$. From this we can conclude the following observation, where $f[i_1, \ldots, i_k] := f[i_1] \ldots [i_k]$.

Observation 7.7. *The cut-reduction operator for infinitary propositional logic is a polynomial time operation in the following sense.*

Let $\mathcal{F}$ and $\mathcal{H}$ be some notation systems for infinitary formulae and the semiformal system $\mathfrak{S}_{\mathcal{F}}$. Assume that $\mathcal{F}$ and $\mathcal{H}$ are polynomial time computable, and that in addition also the functions $\mathcal{F} \times \mathbb{N}^{<\omega} \to \mathcal{F}$, $A, (i_1, \ldots, i_k) \mapsto A[i_1, \ldots, i_k]$ and $\mathcal{H} \times \mathbb{N}^{<\omega} \to \mathcal{H}$, $h, (i_1, \ldots, i_k) \mapsto h[i_1, \ldots, i_k]$ are polynomial time computable.

Then, $\mathcal{CH}$ and the function $\mathcal{H} \times \mathbb{N}^{<\omega} \to \mathcal{CH}$, $h, (i_1, \ldots, i_k) \mapsto (\mathsf{E}h)[i_1, \ldots, i_k]$ are polynomial time computable.

8 Bounded Arithmetic

We will now apply the results on the size of proof notations to Bounded Arithmetic. To keep the presentation simple we will be quite liberal about the language and the basic axioms.

Definition 8.1 (Language of Bounded Arithmetic)**.** *The language $\mathcal{L}_{\mathrm{BA}}$ of Bounded Arithmetic* contains as non-logical symbols $\{=, \leq\}$ for the binary relation *equality* and *less than or equal*, and a symbol for each ptime function. In particular, it includes a unary function symbols $|\cdot|$ whose interpretation in the standard model $\mathbb{N}$ is given by the function which computes the length of the binary representation of its argument, and a constant c_a for $a \in \mathbb{N}$ whose interpretation in $\mathbb{N}$ is $c_a^{\mathbb{N}} = a$. We will often write $\underline{n}$ instead of c_n, and 0 for c_0.

Bounded quantifiers are introduced as abbreviations. $(\forall x \leq t)A$ is short for $(\forall x)A_x(\min(x,t))$, and $(\exists x \leq t)A$ is short for $(\exists x)A_x(\min(x,t))$. Our introduction of bounded quantifiers is slightly nonstandard. It has the advantage that the usual cut-reduction procedure gives already optimal results. The standard abbreviation of bounded quantification, where e.g. $(\exists x \leq t)A$ denotes $(\exists x)(x \leq t \wedge A)$, would need a modification of cut-reduction to produce optimal bounds, as two logical connectives are to be removed for one bounded quantifier. Nevertheless, the two kind of abbreviations are equivalent over a weak base theory like BASIC, assuming BASIC includes some standard axiomatisation of $\min$ using $\leq$, for example $a \leq b \to \min(x,y) = x$ and $\min(a,b) = \min(b,a)$.

Definition 8.2 (Bounded Formulas)**.** The set BFOR of bounded $\mathcal{L}_{\mathrm{BA}}$-formulae is the set of $\mathcal{L}_{\mathrm{BA}}$-formulae consisting of literals and closed under $\wedge$, $\vee$, $(\forall x \leq t)$, $(\exists x \leq t)$.

Negation of complex formulae is an operation on formulae, according to de-Morgan laws; similarly, we use other connectives as obvious abbreviations. For a set $\mathcal{C}$ of formulas and a formula A, let the $\mathcal{C}$-rank of A, $\mathcal{C}$-rk(A), be the maximal nesting of logical connectives until a sub-formula in $\mathcal{C}$ is reached.

We now define a restricted (also called *strict*) delineation of bounded formulae.

Definition 8.3. The set $\mathrm{s}\Sigma^{\mathrm{b}}_d$ is the subset of bounded $\mathcal{L}_{\mathrm{BA}}$-formulae whose elements are of the form

$$(\exists x_1 \leq t_1)(\forall x_2 \leq t_2) \ldots (Q x_d \leq t_d)(\bar{Q} x_{d+1} \leq |t_{d+1}|)A(\vec{x})$$

with Q and $\bar{Q}$ being of the corresponding alternating quantifier shape, and A being quantifier free.

Definition 8.4. 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 BASIC.

Definition 8.5. Let $\mathrm{Ind}(A, z, t)$ denote the expression

$$A_z(0) \wedge (\forall z < t)(A \to A_z(z+1)) \to A_z(t) \ .$$

The set Φ-$\mathrm{L}^m\mathrm{IND}$ consists of all expressions of the form

$$\mathrm{Ind}(A, z, 2^{||t|_m|})$$

with $A \in \Phi$, z a variable and t an $\mathcal{L}_{\mathrm{BA}}$-term. Here $|\cdot|_m$ denotes the m-fold iteration of the function symbol $|\cdot|$.

9 Notation systems for Bounded Arithmetic

Let $\mathcal{F}_{\mathrm{BA}}$ be the set of closed formulae in BFOR. We define the outermost connective function on $\mathcal{F}_{\mathrm{BA}}$ to be $\top$

or $\perp$ for true or false literals, respectively, $\bigwedge$ for universally quantified formulae and conjunctions, and $\bigvee$ for existentially quantified formulae and disjunctions. The subformula function 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.

For t a closed term its numerical value $t^{\mathbb{N}} \in \mathbb{N}$ is defined in the obvious way. Let $\to^1_{\mathbb{N}}$ be the compatible closure of $t \mapsto \underline{t^{\mathbb{N}}}$ for t a closed term. Let $\approx_{\mathbb{N}}$ denote the reflexive, symmetric and transitive closure of $\to^1_{\mathbb{N}}$. If the depth of expressions is restricted, and the number of function symbols representing polynomial time functions is also restricted to a finite subset, then the relation $\approx_{\mathbb{N}}$ is polynomial time decidable.

From now on, we will assume that $\mathcal{F}_{\mathrm{BA}}$ implicitly contains such a constant k without explicitly mentioning it. All formulae and terms used in $\mathcal{F}_{\mathrm{BA}}$ are thus assumed to obey the abovementioned restriction on occurrences of function symbols and depth. Then all relations and functions in $\mathcal{F}_{\mathrm{BA}}$ are polynomial time computable.

Let BA^∞ denote the semiformal proof system over $\mathcal{F}_{\mathrm{BA}}$ according to Definition 4.3.

Definition 9.1. The *finitary proof system* $\mathrm{BA}^\star$ is the proof system over $\mathrm{BFOR}, \approx_{\mathbb{N}}, \mathrm{rk}$ which is given by the following set of inference symbols.

$$(\mathrm{Ax}_\Delta)\ \frac{}{\Delta} \quad \text{if } \bigvee \Delta \in \mathrm{BASIC}$$

$$(\textstyle\bigwedge_{A_0 \wedge A_1})\ \frac{A_0 \qquad A_1}{A_0 \wedge A_1} \qquad (\textstyle\bigvee^k_{A_0 \vee A_1})\ \frac{A_k}{A_0 \vee A_1}$$

$$(\textstyle\bigwedge^y_{(\forall x)A})\ \frac{A_x(y)}{(\forall x)A} \qquad (\textstyle\bigvee^t_{(\exists x)A})\ \frac{A_x(t)}{(\exists x)A}$$

$$(\mathrm{IND}^{y,t}_F)\ \frac{\neg F, F_y(y+1)}{\neg F_y(0), F_y(2^{|t|})}$$

$$(\mathrm{IND}^{y,n,i}_F)\ \frac{\neg F, F_y(y+1)}{\neg F_y(\underline{n}), F_y(\underline{n+2^i})}$$

$$(\mathrm{Cut}_C)\ \frac{C \qquad \neg C}{\emptyset}$$

In our finitary proof system Schütte's ω-rule [15] is replaced by rules with Eigenvariable conditions. Of course, the precise name of the Eigenvariable does not matter, as long as it *is* an Eigenvariable. For this reason, we think of the inference symbols $\bigwedge^y_{(\forall x)A}$, $\mathrm{IND}^{y,t}_F$, and $\mathrm{IND}^{y,n,i}_F$ in $\mathrm{BA}^\star$-quasi derivations as binding the variable y in the respective sub-derivations. Substitution is defined according to this intuition.

Definition 9.2 (Inductive definition of $\vec{x}\colon d$)**.** For $\vec{x}$ a finite list of disjoint variables and $d = \mathcal{I} d_0 \dots d_{n-1}$ a $\mathrm{BA}^\star$-quasi-derivation we inductively define the relation $\vec{x}\colon d$ that d is a $\mathrm{BA}^\star$-derivation with free variables among $\vec{x}$ as follows.

If $\vec{x}, y\colon h_0$ and $\mathcal{I} \in \{\bigwedge^y_{(\forall x)A}, \mathrm{IND}^{y,t}_F, \mathrm{IND}^{y,n,i}_F\}$ for some A, F, t, n, i, and $\mathrm{FV}(\Gamma(\mathcal{I}h_0)) \subset \{\vec{x}\}$ then $\vec{x}\colon \mathcal{I}h_0$.

If $\vec{x}\colon h_0$ and $\mathrm{FV}((\exists x)A), \mathrm{FV}(t) \subseteq \{\vec{x}\}$ then $\vec{x}\colon \bigvee^t_{(\exists x)A} h_0$.

If $\vec{x}\colon h_0$, $\vec{x}\colon h_1$ and $\mathrm{FV}(C) \subseteq \{\vec{x}\}$ then $\vec{x}\colon \mathrm{Cut}_C h_0 h_1$.

If $\mathrm{FV}(\Delta) \subseteq \{\vec{x}\}$ then $\vec{x}\colon \mathrm{Ax}_\Delta$,

If $\vec{x}\colon h_0$, $\vec{x}\colon h_1$ and $\mathcal{I} = \bigwedge_{A_0 \wedge A_1}$ with $\mathrm{FV}(A_0 \wedge A_1) \subset \{\vec{x}\}$ then $\vec{x}\colon \mathcal{I}h_0h_1$.

If $\vec{x}\colon h_0$ and $\mathcal{I} = \bigvee^k_{A_0 \vee A_1}$ with $\mathrm{FV}(A_0 \vee A_1) \subset \{\vec{x}\}$ then $\vec{x}\colon \mathcal{I}h_0$.

A $\mathrm{BA}^\star$-derivation is a $\mathrm{BA}^\star$-quasi derivation h such that for some $\vec{x}$ it holds $\vec{x}\colon h$. We call a $\mathrm{BA}^\star$-derivation h *closed*, if $\emptyset\colon h$.

We note that if $\vec{x}\colon h$ then $\mathrm{FV}(\Gamma(h)) \subseteq \{\vec{x}\}$. In particular $\mathrm{FV}(\Gamma(h)) = \emptyset$ for closed h.

Moreover, if $\vec{x}\colon h$ and y is a variable and t a closed term, then $\vec{x} \setminus \{y\}\colon h(t/y)$ and moreover $\Gamma(h(t/y)) \subseteq (\Gamma(h))(t/y)$.

Let $\mathcal{H}_{\mathrm{BA}}$ be the set of closed $\mathrm{BA}^\star$-derivations. For each $h \in \mathcal{H}_{\mathrm{BA}}$ 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 size function $|\cdot|$ on $\mathcal{H}_{\mathrm{BA}}$ is given by $|h| := \mathrm{sz}(h)$ and the height $\mathrm{o}(h)$ is defined according to the above description of a tree of balanced cuts; to bound the length induction is carried out on, a monotone polynomial bounding term for the whole derivation is extracted first. Observe that, using the auxiliary induction inference symbols $(\mathrm{IND}^{y,n,i}_F)$, the translation of induction can be denoted in such a way that the size of $h[i]$ is always bounded by the size of h.

In this way we obtain a notation system for BA^∞ in the sense of Definition 5.1. We note that all the involved functions are polynomial-time computable.

10 Computational content of proofs

We will now show how the results on bounding the lengths of proof notations can be used to obtain characterisations of definable functions.

Assume we have a proof of a statement $(\forall x)(\exists y)\varphi(x,y)$. For any given $n \in \mathbb{N}$ we can use inversion to get a proof of $(\exists y)\varphi(\underline{n}, y)$. The task now is to find a witnessing k for the existential formula. After reducing the cut-rank so that the ranks of remaining cuts match the rank of φ, we can define a path $d = d_0, d_1, d_2, \dots$ through the derivation d of $(\exists y)\varphi(\underline{n}, y)$ such that always $d_{\ell+1} = d_\ell(i)$ for some i, and $\Gamma(d_\ell)$ is of the shape $\Gamma(d_\ell) = (\exists y)\varphi(y), \Gamma_\ell$ where all formulae $A \in \Gamma_\ell$ are false and of rank at most that of φ. As d is well-founded, such a path must be finite. It is easy to note that it has to end with a $\bigvee^k_{(\exists y)\varphi(y)}$-inference for which $\varphi(\underline{k})$ is true. Hence we found our witness.

Such a path can be seen as a canonical path in a local search problem on a specific subset of the BA^∞ derivations. Using notations for these proofs, the above procedure becomes effective and even feasible in many cases. Instantiating this general procedure by different formula complexities and sets of proof notations we reobtain—but in a uniform way!—characterisations of the definable functions of various theories of Bounded Arithmetic.

Our first step in the technical development is to note that all the formulae we deal with are bounded. In other words, even though, say, universal formulae have infinitely many subformulae, only finitely many carry non-trivial information. In fact, it is easy to define, for every derivation h, monotone terms $\mathrm{bd}(h)$ that bounds all the indices ever needed to access a subformula or subderivation, and $\mathrm{ibd}(h)$ that bounds the length of any induction that has to be considered. We also note, that the size of the conclusion of a derivation is polynomially (in fact, linearly) bounded in the size of the notation of a derivation. Finally, we can compute in polynomial time the list $\mathrm{deco}(h)$ of formulae that decorate any inference symbol which occurs in h.

For $s \in \mathbb{N}$ a size parameter we define $\mathcal{H}^s_{\mathrm{BA}} := \{h \in \mathcal{H}_{\mathrm{BA}} : |h| \leq s\}$. Then $\mathcal{H}^s_{\mathrm{BA}}$ is an s-bounded, abstract system of proof notations, because we observe that $h \in \mathcal{H}_{\mathrm{BA}}$ and $h \to h'$ implies $|h'| \leq |h|$.

Remember that $\overline{h}$ for $h \in \mathcal{CH}_{\mathrm{BA}}$ denotes the abstraction of h which allows us to view $\mathcal{CH}_{\mathrm{BA}}$ as a subsystem of $\widetilde{\mathcal{H}_{\mathrm{BA}}}$. For $h \in \mathcal{CH}_{\mathrm{BA}}$ we define $\vartheta(h)(s) := \vartheta(\overline{h})(s)$. Then Theorem 7.4 now reads as follows. If $h \in \mathcal{CH}^s_{\mathrm{BA}}$ and $h \to h'$, then $\vartheta(h)(s) \geq \vartheta(h')(s)$.

Definition 10.1. We define a local search problem L parameterised by a finite set of bounded formulae $\Phi \subset \mathrm{BFOR}$, a *"logical complexity"* $\mathcal{C}$ given as a polynomial time decidable set of $\mathcal{L}_{\mathrm{BA}}$-formulae, a *size parameter* $s \in \mathbb{N}$, an *initial value function* $h_. : \mathbb{N} \to \mathcal{C}\mathrm{omp}\mathcal{H}^s_{\mathrm{BA}}$, where h_a is presented in the form $\mathsf{E}\ldots\mathsf{E}h(\underline{a}/x)$ for some $\mathrm{BA}^\star$-derivation h, and a formula $(\exists y)\varphi(x,y) \in \Phi$ with $\neg\varphi \in \mathcal{C}$. It will have the property that, for every $a \in \mathbb{N}$, $\Gamma(h_a) = \{(\exists y)\varphi(\underline{a}, y)\}$, $\mathcal{C}$-crk$(h_a) \leq 1$, $\mathrm{o}(h_a) = 2^{|a|^{O(1)}}$, $\vartheta(h_a)(s) = |a|^{O(1)}$, and $\mathrm{deco}(h_a) \subseteq \Phi_a$.

The set of *possible solutions* $F(a) \in \mathfrak{P}_{\mathit{fin}}(\mathcal{C}\mathrm{omp}\mathcal{H}^s_{\mathrm{BA}})$ is given as the set of those $h \in \mathcal{C}\mathrm{omp}\mathcal{H}^s_{\mathrm{BA}}$ which satisfy $\Gamma(h) \subseteq \{(\exists y)\varphi(\underline{a}, y)\} \cup \Delta$ for some $\Delta \subseteq \mathcal{C} \cup \neg\mathcal{C}$ such that all $A \in \Delta$ are closed and false, $\mathcal{C}$-crk$(h) \leq 1$, $\mathrm{o}(h) \leq \mathrm{o}(h_a)$, $\vartheta(h)(s) \leq \vartheta(h_a)(s)$, $\mathrm{bd}(h) \leq \mathrm{bd}(h_a)$ and $\mathrm{ibd}(h) \leq \mathrm{ibd}(h_a)$, and $\mathrm{deco}(h) \subseteq \Phi_{\mathrm{bd}(h_a)}$.

The *initial value function* is given by $i(a) := h_a$. The *cost function* is defined as $c(a,h) := \mathrm{o}(h)$. Finally, the *neighbourhood function* is given by setting $N(a,h)$ to be $h[j]$ if the j'th minor premise of the last rule is in the set of possible solutions, and h if no such j exists.

Proposition 10.2. *$F \in \mathrm{P}^{\mathcal{C}}$, $i, c \in \mathrm{FP}$, and $N \in \mathrm{FP}^{\mathcal{C}}[\mathit{wit}, 1]$.*

Proposition 10.3. *The following are properties of L.*

1. *$N(a,h) = h$ implies $\mathrm{tp}(h) = \bigvee^i_{(\exists y)\varphi(\underline{a},y)}$ with $\varphi(\underline{a}, \underline{i})$ true. Thus, the local search problem L defines a multi-function by mapping a to i (this is called the computed multi-function).*

2. *The search problem L in general defines a search problem in $\mathit{PLS}^{\mathcal{C}}$, assuming that we turn the neighbourhood (multi-)function into a proper function, which can easily be achieved by using an intermediate $\mathit{PLS}^{\mathcal{C}}$ search problem which looks for the smallest witness for the case $\mathrm{tp}(h) = \bigwedge_C$. Then $N \in \mathrm{FP}^{\mathcal{C}}$.*

3. *Assume $\mathrm{o}(h_a) = |a|^{O(1)}$. Then the canonical path through L, which starts at h_a and leads to a local minimum, is of polynomial length with terms of polynomial size, thus the computed multi-function is in $\mathrm{FP}^{\mathcal{C}}[\mathit{wit}, \mathrm{o}(h_a)]$.*

We now apply this general considerations to various concrete situations.

Let $i \geq 2$ and assume that $\mathrm{S}^{i-1}_2 \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in \Sigma^{\mathrm{b}}_i$, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By partial cut-elimination we obtain some $\mathrm{BA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$, $\Sigma^{\mathrm{b}}_{i-1}$-crk$(h) \leq 1$, and $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$. We define a search problem by stating its parameters as follows. $\Phi := \mathrm{deco}(h)$ is a finite set of formulae in BFOR, as the "logical complexity" we take $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$, for the size parameter we choose $s := |h|$, the initial value function is given by $h_a := h(\underline{a}/x)$, and the formula is as given, $(\exists y)\varphi(x,y)$.

As $\mathrm{o}(h_a) = O(||a||)$, Proposition 10.3, shows that the computed multi-function of this search problem is in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}[\mathrm{wit}, O(\log n)]$, which coincides with the description given by Krajíček [10].

Let $i > 0$ and assume that $\mathrm{S}^i_2 \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in \Sigma^{\mathrm{b}}_i$, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By partial cut-elimination we obtain some $\mathrm{BA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$, $\Sigma^{\mathrm{b}}_{i-1}$-crk$(h) \leq 2$, and $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$. We define a search problem by stating its parameters as follows. $\Phi := \mathrm{deco}(h)$ is a finite set of formulae in BFOR, as the "logical complexity" we take $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$, for the size parameter we choose $s := |h|$, the initial value function is given by $h_a := \mathsf{E}h(\underline{a}/x)$, and the formula is as given, $(\exists y)\varphi(x,y)$.

As $\mathrm{o}(h_a) = |a|^{O(1)}$, Proposition 10.3, shows that the computed multi-function of this search problem is in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}[\mathrm{wit}, n^{O(1)}] = \mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}[\mathrm{wit}]$. But this immediately implies that the Σ^{b}_i-definable functions of S^i_2 are in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}$, because a witness query to $(\exists z < t)\psi(u,z)$ can be replaced by $|t|$ many usual (non-witness) queries to $\chi(a, b, u) = (\exists z < t)(a \leq z < b \wedge \psi(u, z))$ using a divide and conquer strategy. This characterisation coincides with the one given by Buss [5].

Let $i > 0$ and assume that $\mathrm{S}_2^{i+1} \vdash (\forall x)(\exists y)\varphi(x, y)$ with $(\exists y)\varphi(x, y) \in \Sigma_i^{\mathrm{b}}$, $\varphi \in \Pi_{i-1}^{\mathrm{b}}$. By partial cut-elimination we obtain some $\mathrm{BA}^{\star}$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x, y)\}$, $\Sigma_{i-1}^{\mathrm{b}}$-crk$(h) \leq 3$, and $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$. We define a search problem by stating its parameters as follows. $\Phi := \mathrm{deco}(h)$ is a finite set of formulae in BFOR, as the "logical complexity" we take $\mathcal{C} := \Sigma_{i-1}^{\mathrm{b}}$, for the size parameter we choose $s := |h|$, the initial value function is given by $h_a := \mathsf{EE}h(\underline{a}/x)$, the formula is as given, $(\exists y)\varphi(x, y)$.

By Proposition 10.3, this defines a search problem in $\mathrm{PLS}^{\Sigma_{i-1}^{\mathrm{b}}}$. This coincides with the description given by Buss and Krajíček [7].

Let $i \geq 1$, $j \geq 0$, and assume that $\Sigma_{i+j}^{\mathrm{b}}$-$\mathrm{L}^{2+j}\mathrm{IND} \vdash (\forall x)(\exists y)\varphi(x, y)$ with $(\exists y)\varphi(x, y) \in \Sigma_{i+1}^{\mathrm{b}}$, $\varphi \in \Pi_i^{\mathrm{b}}$. By partial cut-elimination we obtain some $\mathrm{BA}^{\star}$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x, y)\}$, Σ_i^{b}-crk$(h) \leq j + 1$, and $\mathrm{o}(h(\underline{a}/x)) = O(|a|_{3+j})$. We define a search problem by stating its parameters as follows. $\Phi := \mathrm{deco}(h)$ is a finite set of formulae in BFOR, as the "logical complexity" we take $\mathcal{C} := \Sigma_i^{\mathrm{b}}$, for the size parameter we choose $s := |h|$, the initial value function is given by

$$h_a := \underbrace{\mathsf{E} \ldots \mathsf{E}}_{j \text{ times}} h(\underline{a}/x)$$

and the formula is, as given, $(\exists y)\varphi(x, y)$.

As $\mathrm{o}(h_a) = O(||a||)$, Proposition 10.3, 3., shows that the computed multi-function of this search problem is in $\mathrm{FP}^{\Sigma_i^{\mathrm{b}}}[\mathrm{wit}, 2_j(\mathcal{O}(\log^{2+j} n))]$, which coincides with the description given by Pollett [14].

Conclusions and Future Work

In this article we have shown that one application of cut-reduction on proof notations behaves feasibly. Explicit bounds have been obtained. We then applied these bounds to Bounded Arithmetic to reobtain all known definability results in a uniform way.

In the future, the authors will try to build on these notations to obtain new definability results for hithero uncharacterised classes.

Acknowledgements

The authors gratefully acknowledge support by the Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/D03809X/1.

References

[1] K. Aehlig and A. Beckmann. On the computational complexity of cut-reduction. Technical Report CSR15-2007, Department of Computer Science, Swansea University, Dec. 2007. `http://arxiv.org/abs/0712.1499`.

[2] K. Aehlig and H. Schwichtenberg. A syntactical analysis of non-size-increasing polynomial time computation. In *Proceedings of the Fifteenth IEEE Symposium on Logic in Computer Science (LICS '00)*, pages 84 – 91, June 2000.

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

[4] W. 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.

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

[6] S. R. Buss. Bounded arithmetic and constant depth Frege proofs. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 153–174. Dept. Math., Seconda Univ. Napoli, Caserta, 2004.

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

[8] G. Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[9] G. Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39:405–431, 1935.

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

[11] G. Kreisel, G. Mints, and S. 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.

[12] G. 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.

[13] 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.

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

[15] K. Schütte. Die unendliche Induktion in der Zahlentheorie. *Mathematische Annalen*, 122:369–389, 1951.

[16] W. 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.