On the computational complexity of cut-reduction

Klaus Aehlig Arnold Beckmann
Department of Computer Science
University of Wales Swansea
Singleton Park, Swansea SA2 8PP, UK
{k.t.aehlig|a.beckmann}@swansea.ac.uk

February 15, 2009

Abstract

Using appropriate notation systems for proofs, cut-reduction can often be rendered feasible on these notations. 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 and Related Work

Since Gentzen's invention of the "Logik Kalkül" LK and the proof of his "Hauptsatz" [Gen35a, Gen35b], cut-elimination has been studied in many papers on proof theory. Mints' invention of continuous normalisation [Min78, KMS75] isolates operational aspects of normalisation, that is, the manipulations on (infinitary) propositional derivations. These operational aspects are described independently of the system's proof theoretic complexity, but at the expense of introducing the void logical rule of *repetition* to balance derivation trees.

$$\frac{\Gamma}{\Gamma}\,(\text{Rep})$$

Note that this rule is both logically valid and preserves the sub-formula property, which, in particular, means that it does not harm computational tasks related to derivations as long as it does not occur too often.

It is well-known that, using (Rep), the cut-elimination operator becomes a primitive recursive function which is continuous w.r.t. the standard metric on infinitary trees: the normalisation procedure requires only as much information of the input as it produces output, using (Rep) as the last inference rule of the normal derivation, if the result cannot immediately be determined ("please wait").

In fact, there is a quite tight relationship between the use of repetition constructors and steps of computation. One can associate some repetition rules with β-reductions in the simply-typed lambda calculus. In this way, bounds on the sum of the number of computation steps and the size of the output can be obtained [AJ05] that strengthen earlier results [Bec01]. Using the ω-rule [Sch51], this method can also be applied to Gödel's [Göd58] system T.

In this article, we will re-examine this situation. We will show that the cut-reduction operator can be understood as a polynomial time operation in a natural way, see Observation 8.16. We will work with proof notations which give implicit descriptions of (infinite) propositional proofs. A proof notation system will be 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. For this, we build on Buchholz' technically very smooth approach to notation systems for continuous cut-elimination [Buc91, Buc97]. Our main result of the first part of this article, in particular, implies the following statement, as can be seen from Corollary 8.15. Let $2_n(x)$ denote the n-fold iteration of exponentiation 2^x.

> Let d be some propositional derivation, and assume that all sub-proofs of d can be denoted with notations of size bounded by s, and that the height of d is h. Then, all sub-proofs of the derivation obtained from d by reducing the complexity of cut-formulae by k can be denoted by notations of size bounded by $2_{k-1}(2h) \cdot s$.

Observe that the size of notations is (iterated) exponential only in the height of the original derivation. In the second part of this article we will identify situations occurring in proof-theoretical investigations of Bounded Arithmetic where this height is bounded by an iterated logarithm of some global size parameter, making these sizes feasible.

Bounded Arithmetic has been introduced by Buss [Bus86] as theories of arithmetic with a strong connection to computational complexity. For sake of simplicity of this introduction, we will concentrate only on the Bounded Arithmetic theories S^i_2. These theories are given as first order theories of arithmetic in a language which suitably extends that of Peano Arithmetic where induction is restricted in two ways. First, logarithmic induction is considered which only 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, which defines a kind of logarithm on natural numbers. As in these theories exponentiation will not be a total function, this is a proper restriction. Second, the properties which can be inducted on, must be described by a suitably restricted ("bounded") formula. The class of formulae used here are the Σ^{b}_i-formulae which exactly characterise Σ^p_i, that is, properties of the i-th level of the polynomial time hierarchy of predicates. The main ingredients of the theories S^i_2 are the instances of logarithmic induction for Σ^{b}_i formulae.

A multifunction is a total relation. Let a multifunction f be called Σ^{b}_j-definable in S^i_2, if its graph can be expressed by a Σ^{b}_j-formula φ, such that the totality of f, which renders as $(\forall x)(\exists y)\varphi(x, y)$, is provable from the S^i_2-axioms in first-order logic. The main results characterising definable multifunctions of Bounded Arithmetic are the following.

- Buss [Bus86] has characterised the Σ^{b}_i-definable functions of S^i_2 as $\mathrm{FP}^{\Sigma^b_{i-1}}$, the i-th level of the polynomial time hierarchy of functions.
- Krajíček [Kra93] has characterised the $\Sigma^{\mathrm{b}}_{i+1}$-definable multifunctions of S^i_2 as the class $\mathrm{FP}^{\Sigma^b_i}[wit, O(\log n)]$ of multifunctions which can be computed in polynomial time using a witness oracle from Σ^p_i, where the number of oracle queries is restricted to $O(\log n)$ many (n being the length of the input).
- Buss and Krajíček [BK94] have characterised the $\Sigma^{\mathrm{b}}_{i-1}$-definable multifunctions of S^i_2 as projections of solutions to problems from $\mathrm{PLS}^{\Sigma^b_{i-2}}$, 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 in the following way. First, we will define a suitable notation system $\mathcal{H}_{\mathrm{BA}}$ for propositional derivations which are obtained by translating Bounded Arithmetic proofs. The propositional translation used here is well-known in proof-theoretic investigations; the translation has been described by Tait [Tai68], and later was independently discovered by Paris and Wilkie [PW85]. In the Bounded-Arithmetic world it is known as the *Paris-Wilkie translation.*

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 multifunctions for Bounded Arithmetic. Assume that $(\forall x)(\exists y)\varphi(x, y)$, describing the totality of some multifunction, is provable in some Bounded Arithmetic theory. Fix a particularly nice formal proof p of this. Given $a \in \mathbb{N}$ we want to describe a procedure which finds some b such that $\varphi(\underline{a}, \underline{b})$ holds. Invert the proof p of $(\forall x)(\exists y)\varphi(x, y)$ to a proof of $(\exists y)\varphi(x, y)$ where x is a fresh variable, then substitute $\underline{a}$ for all occurrences of

x. This yields a proof of $(\exists y)\varphi(\underline{a}, y)$. Adding an appropriate number of cut-reduction operators we obtain a proof with all cut-formulae of (at most) the same logical complexity as φ. It should be noted that a notation $h(a)$ for this proof can be computed in time polynomial in the length of a.

The general local search problem which finds a witness for $(\exists y)\varphi(\underline{a}, y)$ can now be characterised as follows. The set of solutions are those notations of a suitable size, which denote a derivation having the property that the derived sequent is equivalent to $(\exists y)\varphi(\underline{a}, y)\vee\psi_1\vee\cdots\vee\psi_l$ where all ψ_i are "simple enough" and false. An initial solution is given by $h(a)$. A neighbour to a solution h is a solution which denotes an immediate sub-derivation of the derivation denoted by h, if this exists, and h otherwise. The cost of a notation is the height of the denoted derivation. The search task is to find a notation in the set of solutions which is a fixpoint of the neighbourhood function. Obviously, a solution to the search task must exist. In fact, any solution of minimal cost has this property. Now consider any solution to the search problem. It must have the property, that none of the immediate sub-derivations is in the solution space. This can only happen if the last inference derives $(\exists y)\varphi(\underline{a}, y)$ from a true statement $\varphi(\underline{a}, \underline{b})$ for some $b \in \mathbb{N}$. Thus b is a witness to $(\exists y)\varphi(\underline{a}, y)$, and we can output b as a solution to our original witnessing problem.

Depending on the complexity of logarithmic induction present in the Bounded Arithmetic theory we started with, and the level of definability, we obtain local search problems defined by functions of some level of the polynomial time hierarchy, and different bounds to the cost function. For example, if we start with the Σ^{b}_i-definable functions of S^i_2, we obtain a local search problem defined by properties in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}$, where the cost function is bounded by $|a|^{O(1)}$. Thus, by following the canonical path through the search problem which starts at the initial value and iterates the neighbourhood function, we obtain a path of polynomial length, which describes a procedure in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}$ to compute a witness.

Other research related to our investigations is a paper by Buss [Bus04] which also makes use of the Paris-Wilkie translation to obtain witnessing results by giving uniform descriptions of translated proofs. However, Buss' approach does not explicitely involve cut-elimination. Dynamic ordinal analysis [Bec03, Bec06] characterises the heights of propositional proof trees obtained via the Paris-Wilkie translation and cut-reduction. Therefore, it is not surprising that the bounds obtained by dynamic ordinal analysis coincide with the bounds on cost functions we are exploiting here.

The potential of our approach to the characterisations of definable search problems via notation systems is that it leads to characterisations of so far uncharacterised definable search problems, most notably the Σ^{b}_1-definable search problems in S^i_2 for $i \geq 3$. Research on this topic will be reported at a different place.

This article is organised as follows. First, we introduce (in Section 2) the general notion of a proof system, of which probably the most important example is (infinitary) propositional logic (Section 3). We then introduce (in Section 4) the concept of a notation for a propositional formula and (in Section 5) that of a notation system for proofs. Recalling (in Section 6) cut-elimination for propositional logic, we then (in Section 7) show how a notation system for propositional proof can be extended to one where cut-elimination is available.

To abstract away from unneeded details we consider (in Section 8) notation systems as abstract reduction systems, with "stepping to a sub-proof" as reduction relation. In this setting we then prove our bounds on the size of our notations.

Finally, we apply our results to Bounded Arithmetic. We first introduce Bounded Arithmetic (in Section 9) and notations for formulae (Section 10) and proofs (Section 11). Using the bounds obtained earlier we then (in Section 12) obtain the mentioned characterisations of the definable functions.

2 Proof Systems

Following Buchholz [Buc97], we present a generic concept of a Tait-style proof system. A proof system essentially is a set of rules that tells how to derive finite sets of formulae. These finite sets of formulae ("sequents") are to be read disjunctively.

Even in the generic setting, we want an abstract notion of cut-rank. Therefore, we require our formulae to come with some structure, including a notion of rank. As our main example in mind is infinitary propositional logic, we take formulae as a quite abstract notation system—otherwise complexity issues would be hard to define in the presence of infinite objects. As equality for infinite objects usually is undecidable, we require formulae to come with an intensional equality, i.e., we want to know when two formulae are given to us as the same object. Note that in the main example (discussed in Section 10) in this article propositional formulae are given by arithmetical formulae denoting them. In this situation, extensional equality, i.e., equality of the denoted propositional formula is undecidable, but just using equality of the notations does not identify enough formulae.

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 $\mathfrak{P}_{\mathrm{fin}}(S)$.

Definition 2.1 (Notation System for Formulae)**.** A *notation system for formulae* is a triple $\langle \mathcal{F}, \approx, \mathrm{rk} \rangle$ where $\mathcal{F}$ is a set (of *formulae*), $\approx$ an equivalence 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 2.2 (Sequent)**.** A *sequent* over $\langle \mathcal{F}, \approx, \mathrm{rk} \rangle$ 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 $\mathcal{C}$-$\mathrm{rk}(A)$ instead of $\mathrm{rk}(\mathcal{C}, A)$.

Definition 2.3. A *proof system* $\mathfrak{S}$ *over* $\langle \mathcal{F}, \approx, \mathrm{rk}\rangle$ 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 2.5) be measured by the $\mathcal{C}$-cut-rank.

Notation 2.4. 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 2.5 (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 $\mathcal{C}$-rk$(\mathcal{I})$, the *cut-rank of* $\mathcal{I}$, to be $\mathcal{C}$-rk$(C) + 1$ if $\mathcal{I}$ is of the form $\mathcal{I} = \mathrm{Cut}_C$ with $C \notin \mathcal{C}$, and 0 otherwise.

Remark 2.6. The reason why the notion introduced in Definition 2.5 is called "quasi derivation", rather than "derivation" is that some proof systems might require additional constraints for a proof to be correct. Most prominently, formal systems of (Bounded) Arithmetic might require an Eigenvariable condition, see Definition 11.3.

Definition 2.7. $d \vdash_\approx \Gamma$ is defined as $\Gamma(d) \subseteq {\approx}\Gamma$.

Lemma 2.8. *For* $d = \mathcal{I}(d_\iota)_{\iota<|\mathcal{I}|}$ *we have*

$$d \vdash_\approx \Gamma \quad \Leftrightarrow \quad \Delta(\mathcal{I}) \subseteq {\approx}\Gamma \quad \textit{and} \quad (\forall \iota < |\mathcal{I}|)\ d_\iota \vdash_\approx \Gamma, \Delta_\iota(\mathcal{I})$$

Proof. If $d \vdash_\approx \Gamma$, that is, $\Gamma(d) \subseteq {\approx}\Gamma$, then $\Delta(\mathcal{I}) \subseteq \Gamma(d) \subseteq {\approx}\Gamma$. Moreover, for $\iota < |\mathcal{I}|$, we have $\Gamma(d_\iota) \setminus {\approx}\Delta_\iota(\mathcal{I}) \subseteq \Gamma(d)$ and hence $\Gamma(d_\iota) \subseteq \Gamma(d) \cup {\approx}\Delta_\iota(\mathcal{I}) \subseteq {\approx}(\Gamma \cup \Delta_\iota(\mathcal{I}))$.

If, on the other hand, $\Delta(\mathcal{I}) \subseteq {\approx}\Gamma$ and $(\forall \iota < |\mathcal{I}|)\ d_\iota \vdash_\approx \Gamma, \Delta_\iota(\mathcal{I})$, then $\Gamma(d) = \Delta(\mathcal{I}) \cup \bigcup_{\iota<|\mathcal{I}|}(\Gamma(d_\iota) \setminus {\approx}\Delta_\iota(\mathcal{I})) \subset {\approx}\Gamma \cup \bigcup_{\iota<|\mathcal{I}|}({\approx}(\Gamma \cup \Delta_\iota(\mathcal{I})) \setminus {\approx}\Delta_\iota(\mathcal{I})) \subseteq {\approx}\Gamma$. ∎

3 The Infinitary Proof System for Propositional Logic

The most prominent logic proof systems are designed for is propositional logic. It is standard proof-theoretical practise to translate more complicated systems, like arithmetic, into propositional logic, using infinitary rules, like the ω-rule [Sch51].

In this section we introduce the infinitary calculus for (infinitary) propositional logic. We will never work with this calculus directly, but only access (parts of) it via notations. Nevertheless it is an important source of inspiration. In particular, cut-elimination is best understood thinking about infinite proofs. Therefore, we introduce this system and relate each concept we work with to infinitary logic, by means of appropriate translations.

Definition 3.1. The set of all infinitary formulae $\mathcal{L}_\infty$ is defined inductively as follows. If $c \in \{\top, \bot, \bigwedge, \bigvee\}$ and $A_\iota \in \mathcal{L}_\infty$ for $\iota < |c|$ then $c(A_\iota)_{\iota<|c|} \in \mathcal{L}_\infty$. Here $|\top| = |\bot| = 0$ and $|\bigwedge| = |\bigvee| = \omega$. The rank $\mathcal{C}\text{-rk}(A)$ is defined to be $\mathcal{C}\text{-rk}(c(A_\iota)_{\iota<|c|}) = \sup\{(\mathcal{C}\text{-rk}(A_\iota) + 1) \mid \iota < |c| \text{ and } C \notin \mathcal{C}\}$.

Convention 3.2. $\mathcal{C}$-rk will only be used for $\mathcal{C}$ which are closed under taking subformulae and intensional equal formulae.

Remark 3.3. By our convention, the $\mathcal{C}$-rk in particular has the property that $\mathcal{C}\text{-rk}(C) = 0$ if $C \in \mathcal{C}$.

Notation 3.4. We denote $\top()$ by $\top$, and $\bot()$ by $\bot$.

Definition 3.5. $\neg$ denotes the operation on $\mathcal{L}_\infty$ which computes negation according to the de Morgan rules, i.e.,

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

where $\neg(\top) = \bot$, $\neg(\bot) = \top$, $\neg(\bigwedge) = \bigvee$, and $\neg(\bigvee) = \bigwedge$.

Definition 3.6. The set of all infinitary formulae of finite rank is denoted with $\mathcal{F}_\infty$. The identity between $\mathcal{F}_\infty$-formulae is the "true" set-theoretic equality.

The logical rules associated with infinitary propositional logic are the obvious ones, i.e., to derive a disjunction, it suffices to derive one disjunct, and to derive a conjunction, all the (infinitely many) conjuncts have to be derived.

Definition 3.7. 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:

(Ax) $\quad \dfrac{}{\top}$

$(\bigwedge_A)$ $\quad \dfrac{\ldots \quad A_\iota \quad \ldots \quad (\iota < \omega)}{A}$ for $A = \bigwedge(A_\iota)_{\iota<\omega} \in \mathcal{F}_\infty$

$(\bigvee_A^i)$ $\quad \dfrac{A_i}{A}$ for $A = \bigvee(A_\iota)_{\iota<\omega} \in \mathcal{F}_\infty$ and $i < \omega$

(Cut$_C$) $\dfrac{C \qquad \neg C}{\emptyset}$ for $C \in \mathcal{F}_\infty$

(Rep) $\dfrac{\emptyset}{\emptyset}$

Definition 3.8. The $\mathfrak{S}_\infty$-*derivations* are the $\mathfrak{S}_\infty$-quasi derivations.

As mentioned earlier, proofs of propositional logic are infinite, ω-branching trees labelled with the inference symbols. As the underlying structure is always the full tree, it suffices to describe the labelling, which is best done by a function from the paths in the tree to the set of inference symbols. The set of paths in the full ω-branching tree is the set $\mathbb{N}^{<\omega}$ of the finite sequences of natural numbers.

With a $\mathfrak{S}_\infty$-derivation $d = \mathcal{I}(d_\iota)_{\iota<|\mathcal{I}|}$ we associate a function from $\mathbb{N}^{<\omega}$ to $\mathfrak{S}_\infty$ as follows. $d(\langle\rangle) := \mathrm{last}(d)$ and

$$d(\langle\iota\rangle \frown s) := \begin{cases} d_\iota(s) & \text{if } i < |\mathcal{I}| \\ \mathrm{Ax} & \text{otherwise} \end{cases}$$

4 Notation System for Infinitary Formulae

Formulae of propositional logic are, as seen, built from $\top$ (“true”) and $\bot$ (“false”) by ω-branching conjunctions and disjunctions. In order to reasonably speak about effectiveness and complexity we consider (as we did already in Section 2) abstract notations for formulae; in Section 7 we will consider notations for derivations as well. A notation for a propositional formula essentially is anything which allows to compute the outermost connective and notations of subformulae.

Definition 4.1. A *notation system* $\langle\mathcal{F}, \mathrm{tp}, \cdot[\cdot], \neg, \mathrm{rk}, \approx\rangle$ *for (infinitary) propositional formulae* is a notation system $\langle\mathcal{F}, \approx, \mathrm{rk}\rangle$ for formulae together with functions $\mathrm{tp}\colon \mathcal{F} \to \{\top, \bot, \bigwedge, \bigvee\}$, $\cdot[\cdot]\colon \mathcal{F} \times \mathbb{N} \to \mathcal{F}$, and $\neg\colon \mathcal{F} \to \mathcal{F}$, called *outermost connective*, *sub-formula*, and *negation*, respectively, 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 $f \notin \mathcal{C}$ and $n < |\,\mathrm{tp}(f)|$, 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. Note that the latter would require us to compute canonical representations anyway and so would just push the problem to a different place.

It should also be noted that the intensional equality is truly intensional. Two formulae are only equal, if they are given to us as being equal. The obvious extensional equality would be the largest bisimulation, that is, the largest relation $\sim \subset \mathcal{F} \times \mathcal{F}$ satisfying $f \sim g \to \mathrm{tp}(f) = \mathrm{tp}(g) \wedge f[n] \sim g[n] \wedge \mathcal{C}\text{-rk}(f) = \mathcal{C}\text{-rk}(g) \wedge \neg f \sim \neg g$. However, like most extensional concepts, the largest bisimulation is undecidable in almost all interesting cases and therefore not suited for an investigation of effective notations.

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

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

Observation 4.3. *The following properties hold.*

1. $f \sim g \quad \Leftrightarrow \quad [\![f]\!]_\infty = [\![g]\!]_\infty$,
2. $f \approx g \quad \Rightarrow \quad [\![f]\!]_\infty = [\![g]\!]_\infty$.

5 Semiformal Proof Systems

Definition 5.1. Let $\mathcal{F} = \langle \mathcal{F}, \mathrm{tp}, \cdot[\cdot], \neg, \mathrm{rk}, \approx \rangle$ be a notation system for infinitary propositional formulae. The *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.

(Ax_A) $\quad \dfrac{}{A}$ for $A \in \mathcal{F}$ with $\mathrm{tp}(A) = \top$

$(\bigwedge_C)$ $\quad \dfrac{\ldots \quad C[n] \quad \ldots \quad (n \in \mathbb{N})}{C}$ for $C \in \mathcal{F}$ with $\mathrm{tp}(C) = \bigwedge$

$(\bigvee_C^i)$ $\quad \dfrac{C[i]}{C}$ for $C \in \mathcal{F}$ with $\mathrm{tp}(C) = \bigvee$ and $i \in \mathbb{N}$

(Cut_C) $\quad \dfrac{C \qquad \neg C}{\emptyset}$ for $C \in \mathcal{F}$ with $\mathrm{tp}(C) \in \{\top, \bigwedge\}$

(Rep) $\quad \dfrac{\emptyset}{\emptyset}$

Abbreviations
For $\mathrm{tp}(C) \in \{\bot, \bigvee\}$ let (Cut_C) $\dfrac{C \qquad \neg C}{\emptyset}$ denote $(\mathrm{Cut}_{\neg C})$ $\dfrac{\neg C \qquad C}{\emptyset}$.

Definition 5.2. 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 5.3. Let $d \vdash^\alpha_{\mathcal{C},m} \Gamma$ denote that d is an $\mathfrak{S}_\mathcal{F}$-derivation with $\Gamma(d) \subseteq {\approx}\Gamma$, $\mathcal{C}\text{-crk}(d) \leq m$, and $\mathrm{hgt}(d) \leq \alpha < \omega$.

It should be noted that Definition 5.3 adds weakening in a strong form: the very same proof is also a proof of the weakened sequent. This is a deliberate choice, as weakening never contains any computationally relevant information.

Observation 5.4. *If* $d \vdash^{\alpha}_{\mathcal{C},m} \Gamma$ *with* $\mathcal{I} = \mathrm{last}(d)$ *and* $\iota < |\mathcal{I}|$*, then* $d(\iota) \vdash^{\alpha_\iota}_{\mathcal{C},m} \Gamma, \Delta_\iota(\mathcal{I})$ *for some* $\alpha_\iota < \alpha$*.*

Definition 5.5. 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

$$\begin{aligned}
[\![\mathrm{Ax}_A]\!]_\infty &:= \mathrm{Ax} \\
[\![\textstyle\bigwedge_A]\!]_\infty &:= \textstyle\bigwedge_{[\![A]\!]_\infty} \\
[\![\textstyle\bigvee^i_A]\!]_\infty &:= \textstyle\bigvee^i_{[\![A]\!]_\infty} \\
[\![\mathrm{Cut}_C]\!]_\infty &:= \mathrm{Cut}_{[\![C]\!]_\infty} \\
[\![\mathrm{Rep}]\!]_\infty &:= \mathrm{Rep}
\end{aligned}$$

Observation 5.6. $\Gamma([\![d]\!]_\infty) \subseteq [\![\Gamma(d)]\!]_\infty$

Proof. Induction on d. The "$\subseteq$", instead of the expected "$=$" is due to the fact, that only formulae are removed from the conclusion that are intensionally equal; compare also Observation 4.3. ∎

6 Cut Elimination for Semiformal Systems

Let $\mathcal{F} = \langle \mathcal{F}, \mathrm{tp}, \cdot[\cdot], \neg, \mathrm{rk}, \approx \rangle$ be a notation system for infinitary formulae, and $\mathfrak{S}_{\mathcal{F}}$ the semiformal proof system over $\mathcal{F}$. We define Mints' continuous cut-reduction operator [Min78, KMS75] following the description given by Buchholz [Buc91]. The only modification is our explicit use of intensional equality. For this section we will always assume that $\mathcal{C}$ is closed under taking subformulae and intensional equal formulae.

Theorem 6.1 (and Definition of the Inversion Operator)**.** *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:* $d \vdash^{\alpha}_{\mathcal{C},m} \Gamma, C \quad \Rightarrow \quad \mathbb{I}^k_C(d) \vdash^{\alpha}_{\mathcal{C},m} \Gamma, C[k]$.

Proof by induction on the build-up of d:
Case 1. $\mathrm{last}(d) \in \{\bigwedge_D : D \approx C\}$. Then

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

is a derivation as required.

Case 2. $\mathcal{I} := \mathrm{last}(d) \notin \{\bigwedge_D : D \approx C\}$. Then

$$\mathbb{I}^k_C(d) := \mathcal{I}(\mathbb{I}^k_C(d(i)))_{i<|\mathcal{I}|}$$

is a derivation as required. ∎

Theorem 6.2 (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:* $\quad d_0 \vdash^{\alpha}_{\mathcal{C},m} \Gamma, C \quad \& \quad d_1 \vdash^{\beta}_{\mathcal{C},m} \Gamma, \neg C$ $\quad \& \quad \mathcal{C}\text{-rk}(C) \leq m \quad \Rightarrow \quad \mathbb{R}_C(d_0, d_1) \vdash^{\alpha+\beta}_{\mathcal{C},m} \Gamma.$

Proof by induction on the build-up of d_1: Let $\mathcal{I} = \mathrm{last}(d_1)$.

Case 1. $\Delta(\mathcal{I}) \cap {\approx}\{\neg C\} = \emptyset$. Then $\Delta(\mathcal{I}) \subseteq {\approx}\Gamma$ and $d_1(i) \vdash^{\beta_i}_{\mathcal{C},m} \Gamma, \neg C, \Delta_i(\mathcal{I})$ with $\beta_i < \beta$ for all $i < |\mathcal{I}|$. By induction hypothesis we obtain $\mathbb{R}_C(d_0, d_1(i)) \vdash^{\alpha+\beta_i}_{\mathcal{C},m} \Gamma, \Delta_i(\mathcal{I})$ for $i < |\mathcal{I}|$. Hence

$$\mathbb{R}_C(d_0, d_1) := \mathcal{I}(\mathbb{R}_C(d_0, d_1(i)))_{i<|\mathcal{I}|}$$

is a derivation as required.

Case 2. $\Delta(\mathcal{I}) \cap {\approx}\{\neg C\} \neq \emptyset$. Then $\mathrm{tp}(C) \neq \top$, because otherwise there is some $D \in \Delta(\mathcal{I})$ with $\mathrm{tp}(D) = \bot$, but this is not satisfied by any of the inference symbols of the semiformal system $\mathfrak{S}_{\mathcal{F}}$. Hence $\mathrm{tp}(C) = \bigwedge$. We obtain that $\mathcal{I} = \bigvee^k_D$ for some $k \in \mathbb{N}$ and $D \approx \neg C$, and $d_1(0) \vdash^{\beta_0}_{\mathcal{C},m} \Gamma, \neg C, \neg C[k]$ with $\beta_0 < \beta$. By induction hypothesis we obtain $\mathbb{R}_C(d_0, d_1(0)) \vdash^{\alpha+\beta_0}_{\mathcal{C},m} \Gamma, \neg C[k]$. The Inversion Theorem 6.1 shows $\mathbb{I}^k_C(d_0) \vdash^{\alpha}_{\mathcal{C},m} \Gamma, C[k]$. Now either $C[k] \in \mathcal{C}$ or $\mathcal{C}\text{-rk}(C[k]) < \mathcal{C}\text{-rk}(C) \leq m$, hence

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

is a derivation as required. □

Theorem 6.3 (and Definition)**.** *We define an operator $\mathbb{E}$ such that:* $d \vdash^{\alpha}_{\mathcal{C},m+1} \Gamma \quad \Rightarrow \quad \mathbb{E}(d) \vdash^{2^{\alpha}-1}_{\mathcal{C},m} \Gamma.$

Proof by induction on the build-up of d:

Case 1. $\mathrm{last}(d) = \mathrm{Cut}_C$. Then $\mathcal{C}\text{-rk}(C) \leq m$ and $d(0) \vdash^{\alpha_0}_{\mathcal{C},m+1} \Gamma, C$ and $d(1) \vdash^{\alpha_0}_{\mathcal{C},m+1} \Gamma, \neg C$ with $\alpha_0 < \alpha$. By induction hypothesis we obtain $\mathbb{E}(d(0)) \vdash^{2^{\alpha_0}-1}_{\mathcal{C},m} \Gamma, C$ and $\mathbb{E}(d(1)) \vdash^{2^{\alpha_0}-1}_{\mathcal{C},m} \Gamma, \neg C$.

Case 1.1. $\mathrm{tp}(C) \in \{\top, \bigwedge\}$, then by the last Theorem $\mathbb{R}_C(\mathbb{E}(d(0)), \mathbb{E}(d(1))) \vdash^{2 \cdot 2^{\alpha_0}-2}_{\mathcal{C},m} \Gamma$, and

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

is a derivation as required.

Case 1.2. $\mathrm{tp}(C) \notin \{\top, \bigwedge\}$, then $\mathbb{R}_{\neg C}(\mathbb{E}(d(1)), \mathbb{E}(d(0))) \vdash^{2 \cdot 2^{\alpha_0}-2}_{\mathcal{C},m} \Gamma$. Continue as before.

Case 2. $\mathcal{I} := \mathrm{last}(d) \neq \mathrm{Cut}_C$. Then

$$\mathbb{E}(d) := \mathcal{I}(\mathbb{E}(d(i)))_{i<|\mathcal{I}|}$$

is as required. □

Remark 6.4. 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.

7 Notations for Derivations and Cut-Elimination

As already mentioned in the introduction to Section 4, we are interested in arguing about complexity of proof transformations. For this question to make sense we need a finite representation of infinite proofs. Again, we take a flexible approach. Any form of finite notation is fine, as long as it is easy to compute the last rule of inference and notations for the subderivations.

Definition 7.1. Let $\langle \mathcal{F}_{\mathrm{BA}}, \mathrm{tp}, \cdot[\cdot], \neg, \mathrm{rk}, \approx_{\mathbb{N}} \rangle$ be a notation system for formulae, and $\mathfrak{S}_{\mathcal{F}}$ the propositional proof system over $\mathcal{F}$ from Definition 5.1.

A *notation system* $\mathcal{H} = \langle \mathcal{H}, \mathrm{tp}, \cdot[\cdot], \Gamma, \mathrm{crk}, \mathrm{o}, |\cdot| \rangle$ *for* $\mathfrak{S}_{\mathcal{F}}$ consists of 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 $\mathrm{o}, |\cdot|\colon \mathcal{H} \to \mathbb{N}\backslash\{0\}$ called *denoted last inference*, *denoted sub-derivation*, *denoted end-sequent*, *denoted cut-rank*, *denoted height* and *size*, such that $\mathcal{C}\text{-crk}(h[n]) \leq \mathcal{C}\text{-crk}(h)$, $\mathrm{tp}(h) = \mathrm{Cut}_C$ implies $\mathcal{C}\text{-rk}(C) < \mathcal{C}\text{-crk}(h)$ for $C \notin \mathcal{C}$, $\mathrm{o}(h[\iota]) < \mathrm{o}(h)$ for $\iota < |\,\mathrm{tp}(h)|$, and the following local faithfulness property holds for $h \in \mathcal{H}$:

$$\Delta(\mathrm{tp}(h)) \subseteq {\approx}\Gamma(h) \quad \text{and} \quad \forall \iota < |\,\mathrm{tp}(h)|\ h[\iota] \vdash_{\approx} \Gamma(h), \Delta_\iota(\mathrm{tp}(h)) \ .$$

The local faithfulness property immediately implies the following Proposition.

Proposition 7.2.

$$\Gamma(h[\iota]) \subseteq {\approx}\Big(\Gamma(h) \cup \Delta_\iota(\mathrm{tp}(h))\Big)$$

Definition 7.3. Let $\mathcal{H} = \langle \mathcal{H}, \mathrm{tp}, \cdot[\cdot], \Gamma, \mathrm{crk}, \mathrm{o}, |\cdot| \rangle$ be a notation system for $\mathfrak{S}_{\mathcal{F}}$. The *interpretation* $[\![h]\!]$ *of* $h \in \mathcal{H}$ is inductively defined as the following $\mathfrak{S}_{\mathcal{F}}$-derivation:

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

Observation 7.4. *For $h \in \mathcal{H}$ we have*

$$\begin{aligned} \mathrm{last}([\![h]\!]) &= \mathrm{tp}(h) \\ [\![h]\!](\iota) &= [\![h[\iota]]\!] \qquad \textit{for } \iota < |\,\mathrm{tp}(h)| \\ \Gamma([\![h]\!]) &\subseteq {\approx}\Gamma(h) \end{aligned}$$

We now extend a notation system $\mathcal{H}$ for $\mathfrak{S}_{\mathcal{F}}$ to 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.

Definition 7.5. The *notation system* $\mathcal{CH}$ *for cut-elimination on* $\mathcal{H}$ is given by the set of terms $\mathcal{CH}$ which is inductively defined by

- $\mathcal{H} \subseteq \mathcal{CH}$,
- $h \in \mathcal{CH}$, $C \in \mathcal{F}$ with $\mathrm{tp}(C) = \bigwedge$, $k < \omega \quad \Rightarrow \quad \mathsf{I}_C^k h \in \mathcal{CH}$,
- $h_0, h_1 \in \mathcal{CH}$, $C \in \mathcal{F}$ with $\mathrm{tp}(C) \in \{\top, \bigwedge\} \quad \Rightarrow \quad \mathsf{R}_C h_0 h_1 \in \mathcal{CH}$,

- $h \in \mathcal{CH} \quad \Rightarrow \quad \mathsf{E}h \in \mathcal{CH}$,

where $\mathsf{I}, \mathsf{R}, \mathsf{E}$ are new symbols, and by functions $\mathrm{tp}\colon \mathcal{CH} \to \mathfrak{S}_{\mathcal{F}}$, $\cdot[\cdot]\colon \mathcal{CH} \times \mathbb{N} \to \mathcal{CH}$, $\Gamma\colon \mathcal{CH} \to \mathfrak{P}_{\mathrm{fin}}(\mathcal{F})$, $\mathrm{crk}\colon \mathfrak{P}(\mathcal{F}) \times \mathcal{CH} \to \mathbb{N}$, $\mathrm{o}\colon \mathcal{CH} \to \mathbb{N} \setminus \{0\}$ and $|\cdot|\colon \mathcal{CH} \to \mathbb{N}$ defined by recursion on the build-up of $h \in \mathcal{CH}$:

- If $h \in \mathcal{H}$ then all functions are inherited from $\mathcal{H}$.
- $h = \mathsf{I}^k_C h_0$: Let $\Gamma(h) := \{C[k]\} \cup (\Gamma(h_0) \setminus \approx\{C\})$, $\mathcal{C}\text{-crk}(h) := \mathcal{C}\text{-crk}(h_0)$, $\mathrm{o}(h) := \mathrm{o}(h_0)$, and $|h| := |h_0| + 1$.

  Case 1. $\mathrm{tp}(h_0) \in \{\bigwedge_D \colon D \approx C\}$. Then let $\mathrm{tp}(h) := \mathrm{Rep}$, and $h[0] := \mathsf{I}^k_C h_0[k]$.

  Case 2. Otherwise, let $\mathrm{tp}(h) := \mathrm{tp}(h_0)$, and $h[i] := \mathsf{I}^k_C h_0[i]$.
- $h = \mathsf{R}_C h_0 h_1$: Let $\mathcal{I} := \mathrm{tp}(h_1)$. We define $\Gamma(h) := (\Gamma(h_0) \setminus \approx\{C\}) \cup (\Gamma(h_1) \setminus \approx\{\neg C\})$, $\mathcal{C}\text{-crk}(h) := \max\{\mathcal{C}\text{-crk}(h_0), \mathcal{C}\text{-crk}(h_1), \mathcal{C}\text{-rk}(C)\}$, $\mathrm{o}(h) := \mathrm{o}(h_0) + \mathrm{o}(h_1)$, and $|h| := |h_0| + |h_1| + 1$.

  Case 1. $\Delta(\mathcal{I}) \cap \approx\{\neg C\} = \emptyset$: Then let $\mathrm{tp}(h) := \mathcal{I}$, and $h[i] := \mathsf{R}_C h_0 h_1[i]$.

  Case 2. Otherwise, $\Delta(\mathcal{I}) \cap \approx\{\neg C\} \neq \emptyset$. As in the proof of Theorem 6.2 we can conclude that $\mathrm{tp}(C) \neq \top$. Thus $\mathrm{tp}(C) = \bigwedge$, and $\mathcal{I} = \bigvee^k_D$ for some $k \in \mathbb{N}$ and $D \approx \neg C$. Then let $\mathrm{tp}(h) := \mathrm{Cut}_{C[k]}$ and $h[0] := \mathsf{I}^k_C h_0$, $h[1] := \mathsf{R}_C h_0 h_1[0]$.
- $h = \mathsf{E}h_0$: Let $\Gamma(h) := \Gamma(h_0)$, $\mathcal{C}\text{-crk}(h) := \mathcal{C}\text{-crk}(h_0) \dot{-} 1$, $\mathrm{o}(h) := 2^{\mathrm{o}(h_0)} - 1$, and $|h| := |h_0| + 1$.

  Case 1. $\mathrm{tp}(h_0) = \mathrm{Cut}_C$: Then let $\mathrm{tp}(h) := \mathrm{Rep}$ and
  let $h[0] := \mathsf{R}_C \mathsf{E}h_0[0]\mathsf{E}h_0[1]$ if $\mathrm{tp}(C) \in \{\top, \bigwedge\}$,
  let $h[0] := \mathsf{R}_{\neg C} \mathsf{E}h_0[1]\mathsf{E}h_0[0]$ if $\mathrm{tp}(C) \notin \{\top, \bigwedge\}$.

  Case 2. Otherwise, let $\mathrm{tp}(h) := \mathrm{tp}(h_0)$, and $h[i] := \mathsf{E}h_0[i]$.

Proof. The just defined system is a notation system for $\mathfrak{S}_{\mathcal{F}}$ in the sense of Definition 7.1. To prove this we have to show that

$$\mathrm{o}(h[n]) < \mathrm{o}(h) \qquad \text{for} \qquad n < |\,\mathrm{tp}(h)| \tag{1}$$

$$\mathcal{C}\text{-crk}(h[n]) \leq \mathcal{C}\text{-crk}(h) \tag{2}$$

$$\mathrm{tp}(h) = \mathrm{Cut}_C \text{ with } C \notin \mathcal{C} \ \Rightarrow\ \mathcal{C}\text{-rk}(C) < \mathcal{C}\text{-crk}(h) \tag{3}$$

and that the local faithfulness property for Γ holds. We start by proving (1) by induction on the build-up of $h \in \mathcal{CH}$.

If $h \in \mathcal{H}$ then (1) is inherited from $\mathcal{H}$. If $h = \mathsf{I}^k_C h_0$ then $h[n] = \mathsf{I}^k_C h_0[n']$ for some n' and (1) is immediate by induction hypothesis.

Now let us consider the case $h = \mathsf{R}_C h_0 h_1$. If $h[n] = \mathsf{R}_C h_0 h_1[n']$ for some n' then (1) is immediate by induction hypothesis. The other case is that $h[0] = \mathsf{I}^k_C h_0$ for some k. We compute

$$\mathrm{o}(h[0]) = \mathrm{o}(\mathsf{I}^k_C h_0) = \mathrm{o}(h_0) < \mathrm{o}(h_0) + \mathrm{o}(h_1) = \mathrm{o}(h)$$

since $\mathrm{o}(h_1) > 0$.

Finally, let us consider the case $h = \mathsf{E}h_0$. If $h[n] = \mathsf{E}h_0[n]$ then (1) is immediate by induction hypothesis. Otherwise, we are in the case $h[0] = \mathsf{R}_C(\mathsf{E}h_0[i])(\mathsf{E}h_0[j])$ for some C, i, j. By induction hypothesis we obtain that $\mathrm{o}(h_0[i]) \leq \mathrm{o}(h_0) - 1$ and $\mathrm{o}(h_0[j]) \leq \mathrm{o}(h_0) - 1$. Hence

$$\begin{aligned}\mathrm{o}(\mathsf{R}_C(\mathsf{E}h_0[i])(\mathsf{E}h_0[j])) = \mathrm{o}(\mathsf{E}h_0[i]) + \mathrm{o}(\mathsf{E}h_0[j]) &= 2^{\mathrm{o}(h_0[i])} - 1 + 2^{\mathrm{o}(h_0[j])} - 1 \\ &< 2 \cdot 2^{\mathrm{o}(h_0)-1} - 1 = 2^{\mathrm{o}(h_0)} - 1 = \mathrm{o}(h)\end{aligned}$$

The Properties (2) and (3) are proven by a simple induction on the build up of h. Note that in the only case of a cut not inherited from a subterm of h, that is, in the case $h = \mathsf{R}_C h_0 h_1$ with $\Delta(\mathrm{tp}(h_1)) \cap {\approx}\{\neg C\} \neq \emptyset$, the cut is on $C[k]$, that is, an immediate subformula of C and therefore, by Definition 3.1, of strictly smaller rank than C, since $C \notin \mathcal{C}$.

The local faithfulness property of Γ is shown by induction on the build-up of $h \in \mathcal{CH}$. This yields a somewhat lengthy case distinction, but in each case the argument is straight forward; moreover, a similar proof can be found in the literature [Buc91, Theorem 5.7(a)]. ◻

Remark 7.6. For the computation of Γ, the cut-elimination operators I^k_C, R_C and E behave like the following inference symbols:

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

Definition 7.7. Let $\mathcal{CH}$ be the notation system for cut-elimination on $\mathcal{H}$. The *interpretation* $[\![h]\!]$ is extended inductively from $\mathcal{H}$ to $\mathcal{CH}$ by defining

$$\begin{aligned}[\![\mathsf{I}^k_C h]\!] &= \mathbb{I}^k_C([\![h]\!]) \\ [\![\mathsf{R}_C h_0 h_1]\!] &= \mathbb{R}_C([\![h_0]\!], [\![h_1]\!]) \\ [\![\mathsf{E}h]\!] &= \mathbb{E}([\![h]\!]).\end{aligned}$$

Proposition 7.8. *For $h \in \mathcal{CH}$ we have*

$$\begin{aligned}\mathrm{last}([\![h]\!]) &= \mathrm{tp}(h) \\ [\![h]\!](\iota) &= [\![h[\iota]]\!] \quad \textit{for } \iota < |\,\mathrm{tp}(h)| \\ \mathcal{C}\text{-crk}([\![h]\!]) &\leq \mathcal{C}\text{-crk}(h) \\ \Gamma([\![h]\!]) &\subseteq {\approx}\Gamma(h)\end{aligned}$$

Proof. By induction on the build-up of $h \in \mathcal{CH}$. If $h \in \mathcal{H}$ then the assertion is inherited from $\mathcal{H}$ and Observation 7.4. The remaining cases follow from Theorems 6.1, 6.2 and 6.3. ◻

8 Size Bounds

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 mess of notation, we abstract our problem to a simple term-rewriting system.

Definition 8.1. 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 8.2 (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 $\mathcal{H} = \langle \mathcal{H}, \mathrm{tp}, \cdot[\cdot], \Gamma, \mathrm{crk}, \mathrm{o}, |\cdot| \rangle$ 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 $n < |\operatorname{tp}(d)|$ with $d' = d[n]$.*

Definition 8.3. If $\mathcal{D}$ is an abstract system of proof notations, then $\widetilde{\mathcal{D}}$, the “cut elimination closure”, is the abstract notation system extending $\mathcal{D}$ that is inductively defined by

$$\frac{d \in \mathcal{D}}{d \in \widetilde{\mathcal{D}}} \qquad \frac{d \in \widetilde{\mathcal{D}}}{\mathsf{I}d \in \widetilde{\mathcal{D}}} \qquad \frac{d \in \widetilde{\mathcal{D}} \qquad e \in \widetilde{\mathcal{D}}}{\mathsf{R}de \in \widetilde{\mathcal{D}}} \qquad \frac{d \in \widetilde{\mathcal{D}}}{\mathsf{E}d \in \widetilde{\mathcal{D}}}$$

$$|\mathsf{I}d| = |d| + 1 \qquad |\mathsf{R}de| = |d| + |e| + 1 \qquad |\mathsf{E}d| = |d| + 1$$

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

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

$$o(\mathsf{I}d) = o(d) \qquad o(\mathsf{R}de) = o(d) + o(e) \qquad o(Ed) = 2^{o(d)} - 1$$

where E, R, I are new symbols.

As one easily verifies, we have $o(d) > o(d')$ for all $d, d' \in \widetilde{\mathcal{D}}$ with $d \to d'$. Therefore, the just defined system $\widetilde{D}$ is indeed an abstract system of proof notations.

Let $\mathcal{F}$ be a notation system for formulae, $\mathfrak{S}_{\mathcal{F}}$ the semiformal 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 8.2.

Definition 8.4. The abstraction $\overline{h}$ of $h \in \mathcal{CH}$ is obtained by dropping all sub- and superscripts. It can be defined by induction on the build-up of $h \in \mathcal{CH}$:

- $h \in \mathcal{H} \quad \Rightarrow \quad \overline{h} := h$,

- $h = \mathsf{I}^k_C h_0 \quad\Rightarrow\quad \overline{h} := \mathsf{I}\overline{h_0}$,
- $h = \mathsf{R}_C h_0 h_1 \quad\Rightarrow\quad \overline{h} := \mathsf{R}\,\overline{h_0}\,\overline{h_1}$,
- $h = \mathsf{E} h_0 \quad\Rightarrow\quad \overline{h} := \mathsf{E}\overline{h_0}$.

We denote the set of abstractions for $h \in \mathcal{CH}$ by $\overline{\mathcal{CH}}$.

Observation 8.5. *We note that the abstraction is over approximating in the following sense. For all $h \in \mathcal{CH}$ and $\iota < |\operatorname{tp}(h)|$ we have $\overline{h} \to \overline{h[\iota]}$.*

We now prove a bound on the size of (abstract) notations for cut-elimination. By induction on the build up 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 [AS00] 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 8.6. 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 $|d| \leq s$.

Definition 8.7. 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 as an abstract system of proof notation with the structure induced by $\mathcal{D}$. Here $\to^*$ denotes the reflexive transitive closure of $\to$.

Definition 8.8. 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 8.9. For $\mathcal{D}$ an abstract system of proof notations, we define a monotone function $\vartheta(d)\colon \mathbb{N} \to \mathbb{N}$ for every $d \in \widetilde{\mathcal{D}}$ by induction on the inductive definition of $\widetilde{\mathcal{D}}$ as follows.

- For $d \in \mathcal{D}$ we set $\vartheta(d)(s) = s$.
- $\vartheta(\mathsf{I}d)(s) = \vartheta(d)(s) + 1$
- $\vartheta(\mathsf{R}de)(s) = \max\{|d|+1+\vartheta(e)(s)\ ,\ \vartheta(d)(s)+1\}$
- $\vartheta(\mathsf{E}d)(s) = o(d)(\vartheta(d)(s) + 2)$

Proof. The monotonicity of the defined function $\vartheta(d)$ is immediately seen from the definition and the induction hypothesis. ∎

Proposition 8.10. *If $\mathcal{D}$ is s-bounded then for every $d \in \widetilde{\mathcal{D}}$ we have $|d| \leq \vartheta(d)(s)$.*

Proof. By induction on the inductive definition of $\widetilde{\mathcal{D}}$.

If $d \in \mathcal{D}$ then $\vartheta(d)(s) = s \geq |d|$, since $\mathcal{D}$ is s-bounded. We calculate $\vartheta(\mathsf{I}d)(s) = \vartheta(d)(s) + 1 \geq |d| + 1 = |\mathsf{I}d|$, where we used that $\vartheta(d)(s) \geq |d|$ by induction hypothesis. Also, $\vartheta(\mathsf{R}de)(s) \geq |d| + 1 + \vartheta(e)(s) \geq 1 + |d| + |e| = |\mathsf{R}de|$, using the induction hypothesis for e. Finally, $\vartheta(\mathsf{E}d)(s) = o(d)(\vartheta(d)(s) + 2) \geq \vartheta(d)(s) + 1 \geq |d| + 1 = |\mathsf{E}d|$, where for the first inequality we used that $o(d) \geq 1$, and for the second inequality we used the induction hypothesis. ∎

Theorem 8.11. *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 $\mathsf{I}d \to \mathsf{I}d'$ thanks to $d \to d'$ then $\vartheta(\mathsf{I}d)(s) = \vartheta(d)(s) + 1 \geq \vartheta(d')(s) + 1 = \vartheta(\mathsf{I}d')(s)$, where the inequality is due to the induction hypothesis.

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

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

where for the first inequality we used Proposition 8.10, 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 $\mathsf{E}d \to \mathsf{E}d'$ thanks to $d \to d'$ then $\vartheta(\mathsf{E}d')(s) = o(d')(\vartheta(d')(s) + 2) \leq o(d)(\vartheta(d')(s) + 2) \leq o(d)(\vartheta(d)(s) + 2) = \vartheta(\mathsf{E}d)(s)$.

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

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

where for the inequality we used the induction hypothesis.

If $\mathsf{R}de \to \mathsf{I}d$ then $\vartheta(\mathsf{R}de)(s) \geq \vartheta(d)(s) + 1 = \vartheta(\mathsf{I}d)(s)$. ∎

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

Lemma 8.12. *If $\mathcal{D}$ is s-bounded, and $d \in \widetilde{\mathcal{D}}$ then $\widetilde{\mathcal{D}}_d$ is $\vartheta(d)(s)$-bounded.*

Proof. We first show by induction on the inductive definition of the reflexive transitive closure that for every $d' \in \widetilde{\mathcal{D}}_d = \{d' \in \widetilde{\mathcal{D}} \mid d \to^* d'\}$ we have $\vartheta(d)(s) \geq$

$\vartheta(d')(s)$. The case $d = d'$ is trivial and if $d \to^* d' \to d''$ then $\vartheta(d)(s) \geq \vartheta(d')(s)$ by induction hypothesis and $\vartheta(d')(s) \geq \vartheta(d'')(s)$ by Theorem 8.11.

Now, by Proposition 8.10 we know that $\vartheta(d')(s) \geq |d'|$ for $d' \in \widetilde{\mathcal{D}}$. So, with the previous claim, for $d' \in \widetilde{\mathcal{D}}_d$ we get $\vartheta(d)(s) \geq \vartheta(d')(s) \geq |d'|$, which is the claim. ∎

Corollary 8.13. *If $d \in \mathcal{D}$ is s-bounded then $\mathsf{E}d$ is $o(d)(s+2)$-bounded and $\mathsf{EE}d$ is $2^{o(d)} \cdot o(d) \cdot (s+4)$-bounded.*

Proof. Let $d \in \mathcal{D}$ be s-bounded and $h := o(d)$. First we observe that $\widetilde{(\mathcal{D}_d)}_{d'} = \widetilde{\mathcal{D}}_{d'}$ for any $d' \in \widetilde{(\mathcal{D}_d)}$. So we can assume without loss of generality that $\mathcal{D}$ is s-bounded.

Lemma 8.12 now gives us that $\mathsf{E}d$ is $\vartheta(\mathsf{E}d)(s)$-bounded and $\mathsf{EE}d$ is $\vartheta(\mathsf{EE}d)(s)$-bounded. We calculate $\vartheta(\mathsf{E}d) = o(d)(\vartheta(d)(s) + 2) = o(d)(s + 2) \leq h(s + 2)$ and $\vartheta(\mathsf{EE}d) = o(\mathsf{E}d)(\vartheta(\mathsf{E}d)(d) + 2) = o(Ed)(h(s + 2) + 2) \leq (2^h - 1)(h(s + 2) + 2) \leq 2^h \cdot h \cdot (s + 4)$. ∎

Even though the above Corollary covers all the cases usually needed in practise, it is interesting to consider the general case. 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 $2_n(d)$.

Lemma 8.14. *$\vartheta(E^n d)(s) \leq 2_{n-1}(2 \cdot o(d)) \cdot s$ for all $n \geq 1$, $s \geq 2$ and $o(d) \geq 2$.*

Proof. Induction on n. For the case $n = 1$ we compute $\vartheta(Ed)(s) = o(d)(s+2) \leq 2o(d)s$.

For $n = 2$ we compute $\vartheta(EEd)(s) = (2^{o(d)}-1)(o(d)(s+2)+2)$. For $o(d) = 2$ and $o(d) = 3$ we directly compute that this is bounded by $2^{2o(d)}s$. For $o(d) \geq 4$ we compute $\vartheta(EEd)(s) \leq 2^{o(d)}4o(d)s \leq 2^{2o(d)}s$.

Now assume that the claim holds for $n \geq 2$. We then compute $\vartheta(EE^n d)(s) = o(E^n d)(\vartheta(E^n d)(s) + 2) \leq 2_{n-1}(2^{o(d)} - 1) \cdot (2_{n-1}(2 \cdot o(d)) \cdot s + 2) \leq 2_{n-1}(2^{o(d)} - 1) \cdot 2 \cdot 2_n(o(d)) \cdot s \leq 2_n(o(d)) \cdot 2_n(o(d)) \cdot s \leq 2_n(2 \cdot o(d)) \cdot s$ ∎

As an immediate Corollary we obtain

Corollary 8.15. *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 8.15 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 function ϑ on $\widetilde{\mathcal{D}}$ to ϑ_k by changing all ϑ to ϑ_k and defining for the last case

- $\vartheta_k(\mathsf{E}d)(s) = (k + 1) \cdot (\vartheta(d)(s) + 2)$

Then we obtain as before for $\mathcal{D}$ s-bounded, $d \in \widetilde{\mathcal{D}}$ 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, which also holds in general for infinite derivations with unrestricted (i.e. potentially infinite) heights. Let $f[i_1, \ldots, i_k] := f[i_1] \ldots [i_k]$.

Observation 8.16. *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 semi-formal system $\mathfrak{S}_{\mathcal{F}}$. Assume that $\mathcal{F}$ and $\mathcal{H}$ are polynomial time computable, and that in addition also the functions

$$\begin{aligned} \mathcal{F} \times \mathbb{N}^{<\omega} &\to \mathcal{F} \\ A, (i_1, \ldots, i_k) &\mapsto A[i_1, \ldots, i_k] \end{aligned}$$

and

$$\begin{aligned} \mathcal{H} \times \mathbb{N}^{<\omega} &\to \mathcal{H} \\ h, (i_1, \ldots, i_k) &\mapsto h[i_1, \ldots, i_k] \end{aligned}$$

are polynomial time computable.

Then, $\mathcal{CH}$ and the function

$$\begin{aligned} \mathcal{H} \times \mathbb{N}^{<\omega} &\to \mathcal{CH} \\ h, (i_1, \ldots, i_k) &\mapsto (\mathsf{E}h)[i_1, \ldots, i_k] \end{aligned}$$

are polynomial time computable.

9 Bounded Arithmetic

In the remaining sections of this article, we will apply the machinery from previous sections to re-obtain known characterisations of definable multifunctions in Bounded Arithmetic. 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 have chosen to introduce Bounded Arithmetic very briefly, and in a slightly non-standard way which suits our proof-theoretic investigations most. The reader interested in the general theory of Bounded Arithmetic is kindly referred to the literature [Bus86, Kra95].

Definition 9.1 (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 polynomial time computable function. In particular, it includes a constant c_a for $a \in \mathbb{N}$ whose interpretation in the standard model $\mathbb{N}$ is $c_a^{\mathbb{N}} = a$, unary function symbols $|\cdot|$ and $2^{|\cdot|}$ which have their standard interpretation given by $|\cdot|^{\mathbb{N}} \colon a \mapsto n$ and $2^{|\cdot|^{\mathbb{N}}} \colon a \mapsto 2^n$

where n is the length of the binary representation of a, and the binary function symbols min and # whose standard interpretation are minimisation and $\#^{\mathbb{N}}\colon a, b \mapsto 2^{n\cdot m}$ where n and m are the lengths of the binary representations of a resp. b. We will often write $\underline{a}$ instead of c_a, and 0 for c_0.

Atomic formulae 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 formulae in the usual way by applying it to their members individually.

To define a rank function which relates to the rank function for infinitary formulae, c.f. Definition 3.1, we first define an auxiliary rank function rk'. Let $\mathcal{C}$ be a set of $\mathcal{L}_{\mathrm{BA}}$-formulae (think of Σ^{b}_i), and A an $\mathcal{L}_{\mathrm{BA}}$-formula. We define $\mathcal{C}$-rk'(A) by induction on the complexity of A. If $A \in \mathcal{C} \cup \neg\mathcal{C}$, let $\mathcal{C}$-rk'$(A) := -1$. For $A \notin \mathcal{C} \cup \neg\mathcal{C}$, $\mathcal{C}$-rk'(A) is defined as follows:

- If $A = B \wedge C$ or $A = B \vee C$, let $\mathcal{C}\text{-rk'}(A) := 1 + \max\{\mathcal{C}\text{-rk'}(B), \mathcal{C}\text{-rk'}(C)\}$.
- If $A = (\forall x)B$ or $A = (\exists x)B$, let $\mathcal{C}\text{-rk'}(A) := 1 + \mathcal{C}\text{-rk'}(B)$.

Then we define the *$\mathcal{C}$-rank of A*, denoted $\mathcal{C}$-rk(A), by $\mathcal{C}\text{-rk}(A) := \max\{0, \mathcal{C}\text{-rk'}(A)\}$. Observe that $\Sigma^{\mathrm{b}}_i\text{-rk}(A) \leq \Sigma^{\mathrm{b}}_{i+1}\text{-rk}(A) + 1$.

The proof-theoretic machinery from the first part of this article is adapted from the machinery designed for analysing fragments of Peano Arithmetic, in particular Π_n-Induction. A consequence of this is that cut-elimination is adjusted to formulae of type $\bigwedge$ / Π-formulae, see the definition of (Cut_C) in Definition 5.1. In Bounded Arithmetic however, the focus is on induction with Σ^{b}_i formulae, thus it would be more natural to focus for cut-elimination on formulae of type $\bigvee$ / Σ-formula. We decided to stay with the original cut-elimination machinery so that our investigations could be more closely adapted. As a consequence, the above definition of rank in its base case does not distinguish between $\bigwedge$ and $\bigvee$-type, by considering the cut-rank above $\mathcal{C} \cup \neg\mathcal{C}$ only.

We will use the following abbreviations.

Definition 9.2 (Abbreviations)**.** The expression $A \to B$ denotes the expression $\neg A \vee B$. The expression $s < t$ denotes $\neg\, t \leq s$. Bounded quantifiers are introduced as follows: $(\forall x \leq t)A$ denotes $(\forall x)A_x(\min(x,t))$, $(\exists x \leq t)A$ denotes $(\exists x)A_x(\min(x,t))$, where x may not occur in t.

Our introduction of bounded quantifiers is slightly nonstandard. It has the advantage that the usual cut-reduction procedure already gives 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 Buss' BASIC assuming that this base theory includes some standard axiomatisation of min using $\leq$, for example $a \leq b \to \min(a,b) = a$ and $\min(a,b) = \min(b,a)$. Also, either way makes use of a nonlogical symbol ("$\leq$" versus "min").

Definition 9.3 (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 $\wedge$, $\vee$, $(\forall x \le t)$, $(\exists x \le t)$.

We now define a delineation of bounded formulae. The literature sometimes distinguishes between "strict" or "prenex" versions versus more liberal ones. We do not want to make such a distinction, and define the classes directly in their restricted form.

Definition 9.4. The set Σ_i^{b} is the smallest subset of bounded $\mathcal{L}_{\mathrm{BA}}$-formulae that is closed under taking subformulae and contains all formulae of the form

$$(\exists x_1 \le t_1)(\forall x_2 \le t_2)\dots(Qx_i \le t_i)(\bar{Q}x_{i+1} \le |t_{i+1}|)A(\vec{x})$$

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

Definition 9.5. 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 BASIC.

It should be noted that our set of basic axioms is considerably stronger than the axioms usually taken in formalisations of Bounded Arithmetic. In particular, BASIC is not computable. This shows that our upper bounds on the definable multifunctions are quite independent of the precise axiomatisation; in particular, true Π_1^0-formulae can be added ad libitum. This property is typical to many proof-theoretic investigations.

We will base our definition of Bounded Arithmetic theories on a stronger normal form of induction than usually considered in the literature. Let $|\cdot|_m$ denote the m-fold iteration of the function symbol $|\cdot|$.

Definition 9.6. 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-L^mIND 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.

Definition 9.7. With Σ_j^{b}-L^kIND we also denote the theory consisting of the (universal closures of) formulae in BASIC and Σ_j^{b}-L^kIND. Let S_2^i abbreviate the theory Σ_i^{b}-L^1IND.

Our version of S_2^i is different from the standard version as for example defined in [Bus86], as it is adapted to suit the proof-theoretic investigations we want to pursue. Nevertheless, it is obvious that our version is an extension of the standard one, in the sense that every statement that is provable in "standard

S_2^i" (say, as defined in [Bus86]) can also be proved in the version of S_2^i as given in the previous definition. This follows from the fact that the restricted form of induction as defined in Definition 9.6 implies the usual form, because the following can be proven from BASIC alone:

$$\mathrm{Ind}(A(\min(t,z)), z, 2^{|t|}) \;\to\; \mathrm{Ind}(A(z), z, t) \ .$$

10 A Notation System for Formulae based on Bounded Arithmetic

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

$$\mathrm{tp}(A) := \begin{cases} \top & A \text{ true literal} \\ \bot & A \text{ false literal} \\ \bigwedge & A \text{ is of the form } A_0 \ \wedge\ A_1 \text{ or } (\forall x)B \\ \bigvee & A \text{ is of the form } A_0 \ \vee\ A_1 \text{ or } (\exists x)B \ , \end{cases}$$

and the sub-formula function on $\mathcal{F}_{\mathrm{BA}} \times \mathbb{N}$ by

$$A[n] := \begin{cases} A & A \text{ literal} \\ A_{\min(n,1)} & A \text{ is of the form } A_0 \ \wedge\ A_1 \text{ or } A_0 \ \vee\ A_1 \\ B_x(\underline{n}) & A \text{ is of the form } (\forall x)B \text{ or } (\exists x)B \ . \end{cases}$$

The rank and negation functions for the notation system are those defined for $\mathcal{L}_{\mathrm{BA}}$.

We didn't have much choice on how to render BFOR into a notation system for formulae. Nevertheless, the above definition already shows that we have to work with a non-trivial intensional equality. The reason is that, even though in the process of the propositional translation we can make sure that we only have closed formulae, this still is not enough; we do have other closed terms than just the canonical ones.

Consider, for example, an arithmetical derivation ending in

$$\dfrac{\begin{array}{c}\vdots \\ B(f(\underline{0}))\end{array}}{\exists x.B(x)}$$

where f is some function symbol. In the propositional translation we have to provide some witness i for the $\bigvee^i_{\exists x.B(x)}$-inference. The "obvious" choice seems to take $i = f^{\mathbb{N}}(0)$. But this would require a derivation of $(\exists x.B(x))[f^{\mathbb{N}}(0)] = B(\underline{f^{\mathbb{N}}(0)})$. The translation of the sub-derivation, on the other hand, gives us a derivation of $B(f(\underline{0}))$. So, in order to make this a correct inference in the propositional translation, we have to consider $B(f(\underline{0}))$ and $B(\underline{f^{\mathbb{N}}(0)})$ as intensionally equal. Note that both formulae are extensionally equal.

We will now define an intensional equality which provides the above described identification. For t a closed term its numerical value $t^{\mathbb{N}} \in \mathbb{N}$ is defined in the obvious way. Let $\to_{\mathbb{N}}^{1}$ denote the rewriting relation obtained from

$$\big\{(t, \underline{t^{\mathbb{N}}}) \colon t \text{ a closed term}\big\} \enspace .$$

For example,

$$(\forall x)(x \leq \lfloor \tfrac{1}{2}(\underline{5} \cdot \underline{3}) \rfloor) \to_{\mathbb{N}}^{1} (\forall x)(x \leq \underline{7}) \enspace .$$

Let $\approx_{\mathbb{N}}$ denote the reflexive, symmetric and transitive closure of $\to_{\mathbb{N}}^{1}$.

Proposition 10.1. *The just defined system $\langle \mathcal{F}_{\mathrm{BA}}, \mathrm{tp}, \cdot[\cdot], \neg, \mathrm{rk}, \approx_{\mathbb{N}} \rangle$ forms a notation system for formulae in the sense of Definition 4.1.*

Remark 10.2. It is an open problem what the complexity of $\approx_{\mathbb{N}}$ is (assuming a usual feasible arithmetisation of syntax). However, 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. I.e., let ${\approx_{\mathbb{N}}}^{k}$ denote the restriction of $\approx_{\mathbb{N}}$ to expressions of depth $\leq k$ in which at most the first k function symbols occur. Here, we do not consider constants as 0-ary function symbols, thus constants c_a may occur arbitrarily in such expressions. Terms of depth k build up from the first k function symbols (excluding constants c_a but allowing variables) represent a finite set $\mathcal{F}$ of polynomial time computable functions. Thus, deciding ${\approx_{\mathbb{N}}}^{k}$ is equivalent to testing equality between functions from $\mathcal{F}$ on particular inputs, which can be done in polynomial time.

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. We will come back to this restriction at relevant places. The next observation already makes use of this assumption.

Observation 10.3. *All relations and functions in $\mathcal{F}_{\mathrm{BA}}$ are polynomial time computable.*

Proof. Syntax like *terms* and *formulae* can be Gödelised in a feasible way as explained for example in [Bus86, Chapter 7]. Thus, we obtain that predicates and operations on syntax like $\mathcal{F}_{\mathrm{BA}}$, $\cdot[\cdot]$, $\neg$, and rk, are polynomial time computable. For tp and $\approx_{\mathbb{N}}$ we observe in addition, that, under the just fixed convention, the relation $\approx_{\mathbb{N}}$ is actually ${\approx_{\mathbb{N}}}^{k}$ for some k, and that the truth of literals can be decided in polynomial time. ∎

Definition 10.4. Let BA^{∞} denote the semiformal proof system over $\mathcal{F}_{\mathrm{BA}}$ according to Definition 5.1.

11 A Notation System for BA^∞

Definition 11.1. The *finitary proof system* $\mathrm{BA}^\star$ is the proof system over $\langle \mathrm{BFOR}, \approx_{\mathbb{N}}, \mathrm{rk}\rangle$ which is given by the following set of inference symbols.

(Ax_Δ) $\quad \dfrac{}{\Delta} \quad$ if $\bigvee \Delta \in \mathrm{BASIC}$

$(\bigwedge_{A_0\wedge A_1})$ $\quad \dfrac{A_0 \qquad A_1}{A_0 \wedge A_1}$ $\qquad$ $(\bigvee^k_{A_0\vee A_1})$ $\quad \dfrac{A_k}{A_0 \vee A_1}$ $\quad (k \in \{0,1\})$

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

$(\mathrm{IND}^{y,t}_F)$ $\quad \dfrac{\neg F, F_y(y+1)}{\neg F_y(0), F_y(2^{|t|})}$ $\qquad$ $(\mathrm{IND}^{y,n,i}_F)$ $\quad \dfrac{\neg F, F_y(y+1)}{\neg F_y(\underline{n}), F_y(\underline{n+2^i})}$ $\quad (n, i \in \mathbb{N})$

(Cut_C) $\quad \dfrac{C \qquad \neg C}{\emptyset}$

According to Definition 2.5, a $\mathrm{BA}^\star$-quasi derivation h is equipped with functions $\Gamma(h)$ denoting the endsequent of h, $\mathrm{hgt}(h)$ denoting the height of h, and $\mathrm{sz}(h)$ denoting the size of h. and $\mathcal{C}$-$\mathrm{crk}(h)$ denoting the cut-rank of h above $\mathcal{C}$.

In the following we will not need the cut-rank function which comes with $\mathrm{BA}^\star$-quasi derivations, but we will need a more general cut-rank function gcrk, which will also bound the rank of induction formulas.

Definition 11.2. Let h be a $\mathrm{BA}^\star$-derivation, $h = \mathcal{I}h_0 \dots h_{n-1}$. We define

$$\mathcal{C}\text{-gcrk}(h) := \sup(\{\mathcal{C}\text{-grk}(\mathcal{I})\} \cup \{\mathcal{C}\text{-gcrk}(h_i) \colon i < n\})$$

where $\mathcal{C}$-$\mathrm{grk}(\mathcal{I})$, the *generalised cut-rank of* $\mathcal{I}$, is $\mathcal{C}$-$\mathrm{rk}(C) + 1$ if $\mathcal{I}$ is of the form Cut_C, $\mathrm{IND}^{y,t}_C$ or $\mathrm{IND}^{y,n,i}_C$ for $C \notin \mathcal{C}$, and 0 otherwise.

Observe that Σ^{b}_i-$\mathrm{gcrk}(h) \leq \Sigma^{\mathrm{b}}_{i+1}$-$\mathrm{gcrk}(h) + 1$, which immediately follows from Σ^{b}_i-$\mathrm{gcrk}(\mathcal{I}) \leq \Sigma^{\mathrm{b}}_{i+1}$-$\mathrm{gcrk}(\mathcal{I}) + 1$. To see the latter, in the critical case when $\mathcal{I}$ is of the form Cut_C, $\mathrm{IND}^{y,t}_C$ or $\mathrm{IND}^{y,n,i}_C$ for $C \in \Sigma^{\mathrm{b}}_{i+1} \setminus \Sigma^{\mathrm{b}}_i$, we compute $\Sigma^{\mathrm{b}}_i\text{-gcrk}(\mathcal{I}) = \Sigma^{\mathrm{b}}_i\text{-rk}(C)+1 = \max\{0, \Sigma^{\mathrm{b}}_i\text{-rk'}(C)\}+1 = 0+1 = \Sigma^{\mathrm{b}}_{i+1}\text{-gcrk}(\mathcal{I})+1$.

In our finitary proof system the ω-rule [Sch51] 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. Fortunately, we don't have to make this intuition precise, as we will always substitute only closed (arithmetical) terms into $\mathrm{BA}^\star$-derivations and therefore no renaming of bound variables will be necessary; hence we don't have to define what this renaming would mean. Note, however, that the details of Definition 11.3 of $\mathrm{BA}^\star$-derivations and Definition 11.5 of substitution become obvious with this intuition on mind.

Definition 11.3 (Inductive definition of $\vec{x}\colon h$)**.** For $\vec{x}$ a finite list of distinct variables and $h = \mathcal{I}h_0 \dots h_{n-1}$ a BA*-quasi-derivation we inductively define the relation $\vec{x}\colon h$ that h is a BA*-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 $\mathrm{FV}(t) \cup \mathrm{FV}(\Gamma(\mathcal{I}h_0)) \subseteq \{\vec{x}\}$ 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 $\mathrm{FV}(A_0 \wedge A_1) \subseteq \{\vec{x}\}$ 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 $\mathrm{FV}(A_0 \vee A_1) \subseteq \{\vec{x}\}$ then $\vec{x}\colon \mathcal{I}h_0$.

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

Proposition 11.4. *If* $\vec{x}\colon h$ *then* $\mathrm{FV}(\Gamma(h)) \subseteq \{\vec{x}\}$. *In particular* $\mathrm{FV}(\Gamma(h)) = \emptyset$ *for closed* h.

Proof. Trivial induction on the inductive definition of $\vec{x}\colon h$. ∎

Definition 11.5. For h a BA*-derivation, y a variable and t a closed term of Bounded Arithmetic we define the substitution $h(t/y)$ inductively by setting $(\mathcal{I}h_0 \dots h_{n-1})(t/y)$ to be $\mathcal{I}(t/y)h_0(t/y)\dots h_{n-1}(t/y)$ if $\mathcal{I}$ is not of the form $\bigwedge^y_{(\forall x)A}$, $\mathrm{IND}^{y,t}_F$, or $\mathrm{IND}^{y,n,i}_F$ with the same variable y, and $\mathcal{I}h_0 \dots h_{n-1}$ otherwise.

Substitution for inference symbols is defined by setting

$$
\begin{array}{lcllcl}
\mathrm{Ax}_\Delta(t/y) & = & \mathrm{Ax}_{\Delta(t/y)} & & & \\
\bigwedge_{A_0\wedge A_1}(t/y) & = & \bigwedge_{(A_0\wedge A_1)(t/y)} & \bigvee^k_{A_0\wedge A_1}(t/y) & = & \bigvee^k_{(A_0\wedge A_1)(t/y)} \\
\bigwedge^z_{(\forall x)A}(t/y) & = & \bigwedge^z_{((\forall x)A)(t/y)} & \bigvee^{t'}_{(\exists x)A}(t/y) & = & \bigvee^{t'(t/y)}_{((\exists x)A)(t/y)} \\
\mathrm{IND}^{z,t'}_F(t/y) & = & \mathrm{IND}^{z,t'(t/y)}_{F(t/y)} & \mathrm{IND}^{z,n,i}_F(t/y) & = & \mathrm{IND}^{z,n,i}_{F(t/y)}
\end{array}
$$

We now show the substitution property for BA*-derivations. The formulation of Lemma 11.6 might look a bit strange with "$\subseteq$" instead of the more familiar equality. The reason is, that a substitution may make formulae equal which are not equal without the substitution.

Recalling however Definition 5.3, we note that derivations h in fact prove every superset of $\Gamma(h)$. Of course, an easy consequence of Lemma 11.6 is that if $\Gamma(h) \subseteq \Delta$ then $\Gamma(h(t/y)) \subseteq \Delta(t/y)$.

Lemma 11.6. *Assume* $\vec{x}\colon h$ *and let* y *be a variable and* t *a closed term, then* $\vec{x} \setminus \{y\}\colon h(t/y)$ *and moreover* $\Gamma(h(t/y)) \subseteq (\Gamma(h))(t/y)$.

Proof. We argue by induction on the build-up of h.

In the cases where no substitution occurs (as $h = \mathcal{I}\ldots$ with $\mathcal{I}$ of the form $\bigwedge^y_{(\forall x)A}$, $\mathrm{IND}^{y,t}_F$, or $\mathrm{IND}^{y,n,i}_F$ with the same variable y) both claims are trivial.

Otherwise, by induction hypothesis, we know that the sub-derivations are $\mathrm{BA}^\star$-derivations with the correct set of free variables; since substitution is also carried out in the inference symbols, the y in the variable conditions for Cut_C and $\bigvee^t_{(\exists x)A}$ will also disappear due to the substitution. The Eigenvariable condition $z \notin \mathrm{FV}(\Gamma(h))$ will follow once we have shown the second claim.

For the second claim let $h = \mathcal{I}h_0 \ldots h_{n-1}$. Then

$$h(t/y) = \mathcal{I}' h_0(t/y) \ldots h_{n-1}(t/y)$$

with $\mathcal{I}' = \mathcal{I}(t/y)$, and by induction hypothesis we have

$$\Gamma(h_i(t/y)) \subseteq \Gamma(h_i)(t/y) \ .$$

Hence

$$\begin{aligned}
\Gamma(h(t/y)) &= \Delta(\mathcal{I}') \cup \bigcup_{i<n} \Big(\Gamma(h_i(t/y)) \setminus {\approx_{\mathbb{N}}}\Delta_i(\mathcal{I}')\Big) \\
&\overset{i.h.}{\subseteq} \Delta(\mathcal{I}') \cup \bigcup_{i<n} \Big(\Gamma(h_i)(t/y) \setminus {\approx_{\mathbb{N}}}\Delta_i(\mathcal{I}')\Big) \\
&\overset{!!!}{\subseteq} \Big(\Delta(\mathcal{I}) \cup \bigcup_{i<n} \big(\Gamma(h_i) \setminus {\approx_{\mathbb{N}}}\Delta_i(\mathcal{I})\big)\Big)(t/y) \\
&= \Gamma(h)(t/y)
\end{aligned}$$

At "!!!" we only have inclusion as substitution may make formulae equal (w.r.t. $\approx_{\mathbb{N}}$) which are not equal without the substitution.

This finishes the proof. ∎

We will now define the ingredients for a notation system $\mathcal{H}_{\mathrm{BA}}$ for BA^∞ according to Definition 7.1. The interpretation $[\![h]\!]$ for $h \in \mathcal{H}_{\mathrm{BA}}$ according to Definition 7.3 formalises a translation of closed $\mathrm{BA}^\star$-derivations into BA^∞, which is called *embedding*.

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 $\mathrm{tp}(h)$ as follows: Let $h = \mathcal{I}h_0 \ldots h_{n-1}$,

$$\mathrm{tp}(h) := \begin{cases}
\mathrm{Ax}_A & \text{if } \mathcal{I} = \mathrm{Ax}_\Delta, \text{ where } A \text{ is the “least” true literal in } \Delta \\
\bigwedge_{A_0 \wedge A_1} & \text{if } \mathcal{I} = \bigwedge_{A_0 \wedge A_1} \\
\bigvee^k_{A_0 \vee A_1} & \text{if } \mathcal{I} = \bigvee^k_{A_0 \vee A_1} \\
\bigwedge_{(\forall x)A} & \text{if } \mathcal{I} = \bigwedge^y_{(\forall x)A} \\
\bigvee^{t^{\mathbb{N}}}_{(\exists x)A} & \text{if } \mathcal{I} = \bigvee^t_{(\exists x)A} \\
\mathrm{Rep} & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_F \\
\mathrm{Rep} & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,0}_F \\
\mathrm{Cut}_{F_y(\underline{n+2^i})} & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i+1}_F \\
\mathrm{Cut}_C & \text{if } \mathcal{I} = \mathrm{Cut}_C
\end{cases}$$

For each $h \in \mathcal{H}_{\mathrm{BA}}$ and $j \in \mathbb{N}$ we define the denoted sub-derivation $h[j]$ as follows: Let $h = \mathcal{I}h_0 \dots h_{n-1}$. If $j \geq |\operatorname{tp}(h)|$ let $h[j] := \mathrm{Ax}_{0=0}$. Otherwise, assume $j < |\operatorname{tp}(h)|$ and define

$$h[j] := \begin{cases} h_{\min(j,1)} & \text{if } \mathcal{I} = \bigwedge_{A_0 \wedge A_1} \\ h_0 & \text{if } \mathcal{I} = \bigvee^k_{A_0 \vee A_1} \\ h_0(\underline{j}/y) & \text{if } \mathcal{I} = \bigwedge^y_{(\forall x)A} \\ h_0 & \text{if } \mathcal{I} = \bigvee^t_{(\exists x)A} \\ \mathrm{IND}^{y,0,|t|^{\mathbb{N}}}_F h_0 & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_F \\ h_0(\underline{n}/y) & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,0}_F \\ \mathrm{IND}^{y,n,i}_F h_0 & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i+1}_F \text{ and } j = 0 \\ \mathrm{IND}^{y,n+2^i,i}_F h_0 & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i+1}_F \text{ and } j = 1 \\ h_j & \text{if } \mathcal{I} = \mathrm{Cut}_C \end{cases}$$

The denoted end-sequent function Γ on $\mathcal{H}_{\mathrm{BA}}$ is given by Γ as defined on $\mathrm{BA}^\star$. The size function $|\cdot|$ on $\mathcal{H}_{\mathrm{BA}}$ is given by $|h| := \operatorname{sz}(h)$ as defined for $\mathrm{BA}^\star$. We define the denoted cut-rank function for $h \in \mathcal{H}_{\mathrm{BA}}$ to be $\mathcal{C}\text{-crk}(h) := \mathcal{C}\text{-gcrk}(h)$. We observe that $\mathcal{C}\text{-crk}(h[\iota]) \leq \mathcal{C}\text{-crk}(h)$ for $\iota < |\operatorname{tp}(h)|$, and that $\mathcal{C}\text{-rk}(C) < \mathcal{C}\text{-crk}(h)$ if $\operatorname{tp}(h) = \mathrm{Cut}_C$ and $C \notin \mathcal{C}$.

To define the denoted height function we need some analysis yielding an upper bound to the log of the lengths of inductions which may occur during the embedding (we take the log as this bounds the height of the derivation tree which embeds the application of induction). Let us first assume m is such an upper bound, and let us define the denoted height $\mathrm{o}_m(h)$ of h relative to m: For a $\mathrm{BA}^\star$-derivation $h = \mathcal{I}h_0 \dots h_{n-1}$ we define

$$\mathrm{o}_m(h) := \begin{cases} \mathrm{o}_m(h_0) + i + 1 & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i}_F \\ \mathrm{o}_m(h_0) + m + 1 & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_F \\ 1 + \sup_{i<n} \mathrm{o}_m(h_i) & \text{otherwise} \end{cases}$$

Observe that $\mathrm{o}_m(h) > 0$ (in particular, $\mathrm{o}(\mathrm{Ax}_\Delta) = 1$).

To fill the gap of providing a suitable upper bound function of closed $\mathrm{BA}^\star$-derivations we first need to fix monotone bounding terms for any term in $\mathcal{L}_{\mathrm{BA}}$.

Bounding terms

For a term t we define a term $\operatorname{bd}(t)$ which represents a monotone function with the following property: If $\mathrm{FV}(t) = \{\vec{x}\}$ then

$$(\forall \vec{n}) \qquad t_{\vec{x}}(\underline{\vec{n}})^{\mathbb{N}} \quad \leq \quad \operatorname{bd}(t)_{\vec{x}}(\underline{\vec{n}})^{\mathbb{N}}$$

Let $x_0, x_1, x_2, \dots$ be a fixed list of free variables. We fix for each function symbol f of arity n a *monotone bounding term* T_f with $\mathrm{FV}(T_f) \subseteq \{x_0, \dots, x_{n-1}\}$. E.g., assume that we have fixed for each function symbol f in our language a number $c_f \in \mathbb{N}$ such that $(\forall \vec{n})|f^{\mathbb{N}}(\vec{n})| \leq \max\{2, |\vec{n}|\}^{c_f}$ holds. We then can define

$$T_f \quad := \quad \underbrace{(\max\{2, \vec{x}\}) \# \ldots \# (\max\{2, \vec{x}\})}_{c_f \text{ times}} \ .$$

As the only exception we demand that $T_{|\cdot|} := |x_0|$.

Now, let t be a term. If t is a closed term, let $\mathrm{bd}(t) := \underline{t^{\mathbb{N}}}$. If $t = x$, let $\mathrm{bd}(t) := x$. If $t = ft_1 \ldots t_n$ is not a closed term, let

$$\mathrm{bd}(t) := (T_f)_{\vec{x}}(\mathrm{bd}(t_1), \ldots, \mathrm{bd}(t_n)) \ .$$

Bounding terms for closed $\mathrm{BA}^{\star}$-derivations

For $h \in \mathcal{H}_{\mathrm{BA}}$, the bounding term $\mathrm{bd}(h)$ is intended to bound any variable which occurs during the embedding of h, and the term $|\,\mathrm{ibd}(h)|$ is intended to bound the length of any induction which occurs during the embedding of h.

Let $h = \mathcal{I} h_0 \ldots h_{n-1}$ be in $\mathcal{H}_{\mathrm{BA}}$. Let $\max(n_1, \ldots, n_k)$ denote the maximal value amongst $\{n_1, \ldots, n_k\}$, where we set $\max() = 0$. We define

$$\mathrm{bd}(h) := \begin{cases} \max(\mathrm{bd}(h_0(\underline{\mathrm{bd}(t)}/y)), \mathrm{bd}(t)) & \text{if } \mathcal{I} = \bigwedge^{y}_{(\forall x \leq t)A} \\ \max(\mathrm{bd}(h_0), \mathrm{bd}(t)) & \text{if } \mathcal{I} = \bigvee^{t}_{(\exists x)A} \\ \max(\mathrm{bd}(h_0(\underline{2^{|\,\mathrm{bd}(t)|}}/y)), 2^{|\,\mathrm{bd}(t)|}) & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_{F} \\ \max(\mathrm{bd}(h_0(\underline{n + 2^i}/y)), n + 2^i) & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i}_{F} \\ \max(\mathrm{bd}(h_0), \ldots, \mathrm{bd}(h_{n-1})) & \text{otherwise.} \end{cases}$$

$$\mathrm{ibd}(h) := \begin{cases} \mathrm{ibd}(h_0(\underline{\mathrm{bd}(t)}/y)) & \text{if } \mathcal{I} = \bigwedge^{y}_{(\forall x \leq t)A} \\ \max(\mathrm{ibd}(h_0(\underline{2^{|\,\mathrm{bd}(t)|}}/y)), 2^{|\,\mathrm{bd}(t)|}) & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_{F} \\ \max(\mathrm{ibd}(h_0(\underline{n + 2^i}/y)), 2^i) & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i}_{F} \\ \max(\mathrm{ibd}(h_0), \ldots, \mathrm{ibd}(h_{n-1})) & \text{otherwise.} \end{cases}$$

Now we define for $h \in \mathcal{H}_{\mathrm{BA}}$ the denoted height function $\mathrm{o}(h)$ to be $\mathrm{o}_{|\,\mathrm{ibd}(h)|}(h)$.

Theorem 11.7. *The just defined system $\langle \mathcal{H}_{\mathrm{BA}}, \mathrm{tp}, \cdot[\cdot], \Gamma, \mathrm{crk}, \mathrm{o}(\cdot), |\cdot| \rangle$ forms a notation system for BA^{∞} in the sense of Definition 7.1.*

Proof. We have to show that $\mathcal{H}_{\mathrm{BA}} = \langle \mathcal{H}_{\mathrm{BA}}, \mathrm{tp}, \cdot[\cdot], \Gamma, \mathrm{crk}, \mathrm{o}(\cdot), |\cdot| \rangle$ has the following properties:

1. $\mathcal{C}$-$\mathrm{crk}(h[n]) \leq \mathcal{C}$-$\mathrm{crk}(h)$
2. $\mathrm{tp}(h) = \mathrm{Cut}_C$ implies $\mathcal{C}$-$\mathrm{rk}(C) < \mathcal{C}$-$\mathrm{crk}(h)$ for $C \notin \mathcal{C}$
3. $\mathrm{o}(h[i]) < \mathrm{o}(h)$ for $i < |\,\mathrm{tp}(h)|$
4. $\mathcal{H}_{\mathrm{BA}}$ has the local faithfulness property in the sense of Definition 7.1.

The first two properties are obvious from the definition of crk as given above. The last property is proven in Proposition 11.9.

For 3., we first observe that $\mathrm{o}_{\cdot}(\cdot)$ satisfies the following monotonicity property:

$$m \le m' \quad \Rightarrow \quad \mathrm{o}_m(h) \le \mathrm{o}_{m'}(h) \ . \tag{4}$$

We also observe the following substitution property by inspection:

$$\mathrm{o}_m(h(t/y)) = \mathrm{o}_m(h) \ . \tag{5}$$

Now we can prove the following slightly more general assertion:

$$m \ge |\operatorname{ibd}(h)| \quad \& \quad i < |\operatorname{tp}(h)| \quad \Rightarrow \quad \mathrm{o}_m(h[i]) < \mathrm{o}_m(h) \tag{6}$$

Then the 3. follows using the monotonicity property (4), as $\operatorname{ibd}(h[i]) \le \operatorname{ibd}(h)$.

The proof of (6) is by induction on the build-up of h. Let $h = \mathcal{I}h_0 \dots h_{n-1}$. First assume that $h[i] = h_j(t/y)$. The definition of o_m immediately shows that in this case $\mathrm{o}_m(h) = 1 + \sup_{i<n} \mathrm{o}_m(h_i)$. The substitution property (5) shows that $\mathrm{o}_m(h_j(t/y)) = \mathrm{o}_m(h_j)$. Hence

$$\mathrm{o}_m(h) > \mathrm{o}_m(h_j) = \mathrm{o}_m(h_j(t/y)) = \mathrm{o}_m(h[i]) \ .$$

The remaining cases are the following ones: If $h = \mathrm{IND}_F^{y,t} h_0$, then $h[0] = \mathrm{IND}_F^{y,0,|t|} h_0$. As $|t| \le |\operatorname{bd}(t)| < |\operatorname{ibd}(h)| \le m$ we obtain

$$\mathrm{o}_m(h[0]) = \mathrm{o}_m(h_0) + |t| + 1 < \mathrm{o}_m(h_0) + m + 1 = \mathrm{o}_m(h) \ .$$

If $h = \mathrm{IND}_F^{y,n,k+1} h_0$, then $h[i] = \mathrm{IND}_F^{y,n',k} h_0$ for some n' Hence

$$\mathrm{o}_m(h[i]) = \mathrm{o}_m(h_0) + k + 1 < \mathrm{o}_m(h_0) + k + 2 = \mathrm{o}_m(h) \ .$$

Thus, assertion (6) is proven, which finishes the proof. ∎

Lemma 11.8. *Let $h = \mathcal{I}h_0 \dots h_{n-1}$ be a $\mathrm{BA}^\star$-derivation, and assume $h \vdash_{\approx_\mathbb{N}} \Gamma$.*

1. *$h_i \vdash_{\approx_\mathbb{N}} \Gamma, \Delta_i(\mathcal{I})$ for $i < n$.*
2. *$h(\underline{i}/y) \vdash_{\approx_\mathbb{N}} \Gamma(\underline{i}/y)$ for $i \in \mathbb{N}$ and variables y.*

Proof. 1. is obvious from the definition of $\vdash_{\approx_\mathbb{N}}$. For 2. we compute

$$\Gamma(h(\underline{i}/y)) \overset{\text{Lemma 11.6}}{\subseteq} \Gamma(h)(\underline{i}/y) \overset{h \vdash_{\approx_\mathbb{N}} \Gamma}{\subseteq} (\approx_\mathbb{N}\Gamma)(\underline{i}/y) \overset{\text{definition of } \approx_\mathbb{N}\Gamma}{\subseteq} \approx_\mathbb{N}\big(\Gamma(\underline{i}/y)\big)$$

∎

Proposition 11.9. *$\mathcal{H}_{\mathrm{BA}}$ has the local faithfulness property.*

Proof. Let $h \in \mathcal{H}_{\mathrm{BA}}$ be of the form $h = \mathcal{I}h_0 \dots h_{n-1} \in \mathcal{H}_{\mathrm{BA}}$, and let $\Gamma := \Gamma(h)$. We have to show

$$\Delta(\mathrm{tp}(h)) \subseteq \approx_{\mathbb{N}}\Gamma \quad \text{and} \quad \forall i < |\,\mathrm{tp}(h)|\; h[i] \vdash_{\approx_{\mathbb{N}}} \Gamma, \Delta_i(\mathrm{tp}(h)) \ .$$

We distinguish cases according to the form of $\mathcal{I}$:

Case 1. $\mathcal{I} = \mathrm{Ax}_\Delta$: Then $\Delta \subseteq \Gamma$. Let A be the "least" true literal in Δ. Then $\mathrm{tp}(h) = \mathrm{Ax}_A$ and $A \in \Delta$.

Case 2. $\mathcal{I} = \bigwedge_C$ for $C = A_0 \wedge A_1$. Then $C \in \Gamma$ and $h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, A_0$ and $h_1 \vdash_{\approx_{\mathbb{N}}} \Gamma, A_1$. But $\mathrm{tp}(h) = \bigwedge_C$, $h[0] = h_0$ and $h[i] = h_1$ for $i > 0$, and $\Delta_0(\mathrm{tp}(h)) = A_0$ and $\Delta_i(\mathrm{tp}(h)) = A_1$ for $i > 0$.

Case 3. $\mathcal{I} = \bigvee^k_{A_0 \vee A_1}$. This case is similar to Case 2.

Case 4. $\mathcal{I} = \bigwedge^y_{(\forall x)A}$: Then $(\forall x)A \in \Gamma$ and $h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, A_x(y)$. Using the Eigenvariable condition and the previous Lemma we obtain $h_0(\underline{i}/y) \vdash_{\approx_{\mathbb{N}}} \Gamma, A_x(\underline{i})$. But $\mathrm{tp}(h) = \bigwedge_{(\forall x)A}$, $h[i] = h_0(\underline{i}/y)$ and $\Delta_i(\mathrm{tp}(h)) = A_x(\underline{i})$.

Case 5. $\mathcal{I} = \bigvee^t_{(\exists x)A}$: Then $(\exists x)A \in \Gamma$ and $h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, A_x(t)$, hence $h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, A_x(\underline{t^{\mathbb{N}}})$. But $\mathrm{tp}(h) = \bigvee^{t^{\mathbb{N}}}_{(\exists x)A}$, $h[0] = h_0$ and $\Delta_0(\mathrm{tp}(h)) = A_x(\underline{t^{\mathbb{N}}})$.

Case 6. $\mathcal{I} = \mathrm{IND}^{y,t}_F$: Then $\neg F_y(0), F_y(2^{|t|}) \in \Gamma$ and $h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, \neg F, F_y(y+1)$. Hence $\mathrm{IND}^{y,0,|t^{\mathbb{N}}|}_F h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma$ as $F_y(2^{|t|}) \approx_{\mathbb{N}} F_y(\underline{0 + 2^{|t^{\mathbb{N}}|}})$. But $\mathrm{tp}(h) = \mathrm{Rep}$ and $h[0] = \mathrm{IND}^{y,0,|t^{\mathbb{N}}|}_F h_0$.

Case 7. $\mathcal{I} = \mathrm{IND}^{y,n,0}_F$: Then $\neg F_y(\underline{n}), F_y(\underline{n+1}) \in \Gamma$ and $h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, \neg F, F_y(y+1)$. Using the Eigenvariable condition and the previous Lemma we obtain $h_0(\underline{n}/y) \vdash_{\approx_{\mathbb{N}}} \Gamma, \neg F_y(\underline{n}), F_y(\underline{n}+1)$, thus $h_0(\underline{n}/y) \vdash_{\approx_{\mathbb{N}}} \Gamma$ using $F_y(\underline{n+1}) \approx_{\mathbb{N}} F_y(\underline{n}+1)$. But $\mathrm{tp}(h) = \mathrm{Rep}$ and $h[0] = h_0(\underline{n}/y)$.

Case 8. $\mathcal{I} = \mathrm{IND}^{y,n,i+1}_F$: Then $\neg F_y(\underline{n}), F_y(\underline{n + 2^{i+1}}) \in \Gamma$ and we have $h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, \neg F, F_y(y+1)$ which implies

$$h[0] = \mathrm{IND}^{y,n,i}_F h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, \neg F, F_y(\underline{n + 2^i})$$

and

$$h[1] = \mathrm{IND}^{y,n+2^i,i}_F h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, \neg F_y(\underline{n+2^i}), F_y(\underline{n + 2^{i+1}}) \ .$$

Using $\neg F_y(\underline{n}), F_y(\underline{n+2^{i+1}}) \in \Gamma$ this simplifies to

$$h[0] \vdash_{\approx_{\mathbb{N}}} \Gamma, F_y(\underline{n+2^i}) \quad \text{and} \quad h[1] \vdash_{\approx_{\mathbb{N}}} \Gamma, \neg F_y(\underline{n+2^i}) \ .$$

Further, $\mathrm{tp}(h) = \mathrm{Cut}_{F_y(\underline{n+2^i})}$ and thus $\Delta(\mathrm{tp}(h)) = \emptyset$, $\Delta_0(\mathrm{tp}(h)) = \{F_y(\underline{n+2^i})\}$ and $\Delta_1(\mathrm{tp}(h)) = \{\neg F_y(\underline{n+2^i})\}$.

Case 9. $\mathcal{I} = \mathrm{Cut}_C$: Then $h_0 \vdash_{\approx_{\mathbb{N}}} \Gamma, \neg C$ and $h_1 \vdash_{\approx_{\mathbb{N}}} \Gamma, C$. But $h[0] = h_0$, $h[1] = h_1$ and $\mathrm{tp}(h) = \mathrm{Cut}_C$, thus $\Delta(\mathrm{tp}(h)) = \emptyset$, $\Delta_0(\mathrm{tp}(h)) = \{\neg C\}$ and $\Delta_1(\mathrm{tp}(h)) = \{C\}$. ∎

Observation 11.10. *The following relations and functions are polynomial time computable: the functions $h \mapsto \Gamma(h)$, $h \mapsto \mathrm{hgt}(h)$, and $h \mapsto \mathrm{sz}(h)$ denoting the endsequent, the height and the size for an input promised to be a* $\mathrm{BA}^\star$*-quasi derivation h; the bounding term $t \mapsto \mathrm{bd}(t)$ for terms t occurring in $\mathcal{F}_{\mathrm{BA}}$ and the relations* $\mathrm{bd}(h) \leq m$ *and* $\mathrm{ibd}(h) \leq m$ *on inputs in $\mathcal{H}_{\mathrm{BA}} \times \mathbb{N}$; the functions $h \mapsto \mathrm{tp}(h)$, $h, i \mapsto h[i]$, $h \mapsto \Gamma(h)$, $m, h \mapsto \mathrm{o}_m(h)$ and $h \mapsto |h|$ for inputs in $\mathcal{H}_{\mathrm{BA}}$.*

Here we used our implicit assumptions on the restrictions of term-depths and occurrences of function symbols as explained at the end of Section 9.

The relations $\mathrm{BA}^\star$*, "being a* $\mathrm{BA}^\star$*-quasi-derivation", and $\mathcal{H}_{\mathrm{BA}}$ in general are not polynomial time computable (even not computable). Later in the applications they will be restricted to suitable polynomial time computable subsets.*

Proof. For bounding terms we use our assumption that a fixed (finite) number of function symbols and term depth is only allowed, which implies that terms can only denote a fixed finite number of different polynomial time computable functions. The computation of $\mathrm{bd}(h)$ computes a monotone increasing sequence of values by successively applying one of the finitely many polynomial time computable functions. The length of the sequence, i.e. the number of values to be computed, is bounded by the size of h as a term. Thus, the relation $\mathrm{bd}(h) \leq m$ is clearly polynomial time decidable, because once the bound m is exceeded during the process of computing $\mathrm{bd}(h)$ one can already output *NO*. □

As the function $\mathrm{bd}(h)$ in general may not be polynomially bounded, we cannot conclude in general that $\mathrm{o}(h)$ is polynomial time computable. However, the function $m, h \mapsto \mathrm{o}_{\min(|\,\mathrm{ibd}(h)|, m)}(h)$ is polynomial time computable and will be sufficient in our applications.

We finish this section by connecting $\mathrm{BA}^\star/\mathcal{H}_{\mathrm{BA}}$ to the theories of Bounded Arithmetic as defined in Section 9. This step also includes some proof normalisation which we will not give in all details as it is similar to the ones known in the literature, for example free cut-elimination in [Bus86] or partial cut-elimination in [Bec03].

Theorem 11.11 (Partial Cut-elimination)**.** *Assume* Σ^{b}_j-$\mathrm{L}^k\mathrm{IND} \vdash \varphi$ *with* $k \geq 1$, $\varphi \in \mathrm{BFOR}$ *and* $\mathrm{FV}(\varphi) \subseteq \{x\}$*. Then, there is some* $\mathrm{BA}^\star$*-derivation h such that* $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{\varphi\}$, Σ^{b}_j-$\mathrm{gcrk}(h) = 0$ *and* $\mathrm{o}(h(\underline{a}/x)) = O(|a|_{k+1})$*.*

Proof. Assume Σ^{b}_j-$\mathrm{L}^k\mathrm{IND} \vdash \varphi$ with $k \geq 1$, $\varphi \in \mathrm{BFOR}$ and $\mathrm{FV}(\varphi) \subseteq \{x\}$. Induction in Σ^{b}_j-$\mathrm{L}^k\mathrm{IND}$ is given as universal closures of axioms of the form $\mathrm{Ind}(A, z, 2^{||t|_k|})$ with $A \in \Sigma^{\mathrm{b}}_j$ and $\mathcal{L}_{\mathrm{BA}}$-terms t. They can be derived in $\mathrm{BA}^\star$ in the following form: there is some $\mathrm{BA}^\star$-derivation h_A which satisfies that $\Gamma(h_A) = \{\mathrm{Ind}(A, z, 2^{||t|_k|})\}$, and that h_A contains one occurrence of an induction inference symbol which is of the form $(\mathrm{IND}^{y,|t|_k}_{B(z)})$ for $B(z)$ of the form $A_z(\min(z, 2^{||t|_k|}))$. Thus, there is some $\mathrm{BA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{\varphi\}$ and such that all occurring induction inference symbols are of the form $(\mathrm{IND}_C^{z,|t|_k})$ for some $C \in \Sigma_j^{\mathrm{b}}$ and some $\mathcal{L}_{\mathrm{BA}}$-term t. By partial cut-elimination we can ensure that Σ_j^{b}-gcrk$(h) = 0$. To compute $\mathrm{o}(h(\underline{a}/x))$, we first compute $\mathrm{bd}(h(\underline{a}/x)) = 2^{|a|^{O(1)}}$ and $\mathrm{ibd}(h(\underline{a}/x)) = |a|_k{}^{O(1)}$. By definition of $\mathrm{o}_m(h)$ we observe, as all occurring induction inference symbols are of the form $(\mathrm{IND}_C^{z,|t|_k})$, that $\mathrm{o}_m(h) \leq \mathrm{sz}(h) \cdot (m+1)$. Thus

$$\begin{aligned}\mathrm{o}(h(\underline{a}/x)) &= \mathrm{o}_{|\,\mathrm{ibd}(h(\underline{a}/x))|}(h(\underline{a}/x)) \\ &\leq \mathrm{sz}(h(\underline{a}/x)) \cdot (|\,\mathrm{ibd}(h(\underline{a}/x))| + 1) = O(|a|_{k+1}) \ .\end{aligned}$$

□

12 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 multifunctions.

Let us start by describing the idea for computing witnesses using proof trees. Assume we have a Bounded Arithmetic proof of an existential formula $(\exists y)\varphi(y)$ and we want to compute a k such that $\varphi(k)$ is true—in case we are interested in definable multifunctions, such a situation is obtained from a proof of $(\forall x)(\exists y)\varphi(x,y)$ by inverting the universal quantifier to some $n \in \mathbb{N}$. Assume further, we have applied some proof theoretical transformations to obtain a BA^∞ derivation d_0 of $(\exists y)\varphi(y)$ with $\mathcal{C}$-crk$(d_0) \leq \mathcal{C}$-rk(φ) for some set of formulae $\mathcal{C}$ (the choice of $\mathcal{C}$ depends on the level of definability we are interested in). Then we can define a path through d_0, represented by sub-derivations $d_1, d_2, d_3 \ldots$, such that

- $d_{j+1} = d_j(\iota)$ for some $\iota \in |\,\mathrm{last}(d_j)|$
- $\Gamma(d_j) = (\exists y)\varphi(y), \Gamma_j$ where all formulae $A \in \Gamma_j$ are false and satisfy $\mathcal{C}$-rk$(A) \leq \mathcal{C}$-rk(φ).

Such a path must be finite as $\mathrm{hgt}(d_j)$ is strictly decreasing. Say it ends with some d_ℓ. In this situation we must have that $\mathrm{last}(d_\ell) = \bigvee^k_{(\exists y)\varphi(y)}$ and that $\varphi(k)$ is true. Hence we found our witness.

The path which we have just described can be viewed as the canonical path through a related local search problem. Before explaining this, let us fix the notion of a local search problem.

Definition 12.1. An instance of a *local search problem* consists of a set F of possible solutions, an initial value $d \in F$, a cost function $c\colon F \to \mathbb{N}$, and a neighbourhood function $N\colon F \to F$ which satisfy that $c(N(d)) < c(d)$ if $N(d) \neq d$. A solution to a local search problem, called a *local optimum*, is any $d \in F$ such that $N(d) = d$.

Observe that the ingredients of a local search problem guarantee the existence of a local optimum, by starting with the initial value and iterating the neighbourhood function (this defines the *canonical path through the search problem.*)

Now we define a local search problem whose canonical path is the one described above. The set F of possible solutions is defined as the set of all BA^{∞}-derivations d which have the property that all formulae $A \in \Gamma(d) \setminus \{(\exists y)\varphi(y)\}$ are false and satisfy $\mathcal{C}\text{-rk}(A) \leq \mathcal{C}\text{-rk}(\varphi)$. The cost of a possible solution $d \in F$ is given by the height $\mathrm{hgt}(d)$ of the proof tree d. Assume we have fixed some initial value $d_0 \in F$. The neighbourhood function $N \colon \mathrm{BA}^{\infty} \to \mathrm{BA}^{\infty}$ is defined by case distinction on the shape of $\mathrm{last}(d)$ for $d \in F$:

- $\mathrm{last}(d) = \mathrm{Ax}_A$ cannot occur as all atomic formulae in $\Gamma(d)$ are false by definition of F.
- $\mathrm{last}(d) = \bigwedge_{A_0 \wedge A_1}$, then $A_0 \wedge A_1$ must be false, hence some of A_0, A_1 must be false. Let $N(d) := d(0)$ if A_0 is false, and $d(1)$ otherwise.
- $\mathrm{last}(d) = \bigvee^k_{A_0 \vee A_1}$, then $A_0 \vee A_1$ must be false, hence both A_0, A_1 must be false. Let $N(d) := d(0)$.
- $\mathrm{last}(d) = \bigwedge_{(\forall x)A(x)}$. As $(\forall x)A(x)$ is false there is some i such that $A(i)$ is false. Let $N(d) := d(i)$.
- $\mathrm{last}(d) = \bigvee^k_{(\exists x)A(x)}$. If $(\exists x)A(x)$ is different from $(\exists y)\varphi(y)$ then $(\exists x)A(x)$ must be false; let $N(d) := d(0)$. Otherwise, if $\varphi(k)$ is false let $N(d) = d(0)$, and if it is true let $N(d) = d$. Observe that in the very last case we found our witness.
- $\mathrm{last}(d) = \mathrm{Cut}_C$. If C is false let $N(d) := d(0)$, otherwise let $N(d) := d(1)$.

Obviously, this defines a local search problem according to Definition 12.1. As remarked above, a local optimal solution to the search problem allows us to determine a witness.

In the following we will use proof notations from $\mathcal{CH}_{\mathrm{BA}}$ to define local search problems similar to the one we have just described. Utilising the results from previous sections, we will then obtain characterisations of the definable multifunctions of Bounded Arithmetic theories.

As explained above, a first step will be applying proof theoretic transformations to formal proofs in Bounded Arithmetic theories of some formula expressing the totality of a function whose complexity we want to characterise. Here, the level of proof theoretic reduction will be adjusted in such a way that occurring formulae which have to be decided fall exactly in the computational class under consideration. So our main concern in order for this strategy to be meaningful is to find feasible upper bounds for the length of such reduction sequences and for the complexity of derivation notations occurring in them.

12.1 Complexity Notions for $\mathrm{BA}^{\star}$

In order to describe the set of possible solutions for the search problems we are going to define later, we need some notions describing key complexity properties of $\mathrm{BA}^{\star}$ proof notations which we will provide first.

Although $\mathrm{tp}(A) = \bigwedge$ for any A starting with a $\forall$, and thus we can denote infinitely many direct sub-formulae by $A[n]$ for all $n \in \mathbb{N}$, only finitely many carry non-trivial information, because all quantifiers in A (and in particular this outermost $\forall$) are bounded. The next definition makes this formal by assigning first to each closed formula in $\mathcal{F}_{\mathrm{BA}}$, then to each inference symbol in BA^∞, and finally to each proof notation in $\mathcal{CH}_{\mathrm{BA}}$, its range.

Definition 12.2. Let A be a formula in $\mathcal{F}_{\mathrm{BA}}$. We define *the range of A,* denoted $\mathrm{rng}(A)$, by

$$\mathrm{rng}(A) := \begin{cases} 0 & \text{if } A \text{ a literal } , \\ 2 & \text{if } A = B \wedge C \text{ or } A = B \vee C \ , \\ t^{\mathbb{N}} + 1 & \text{if } A = (\forall x \leq t)B \text{ or } A = (\exists x \leq t)B \ . \end{cases}$$

Let $\mathcal{I}$ be an inference symbol of BA^∞. We define *the range of $\mathcal{I}$,* denoted $\mathrm{rng}(\mathcal{I})$, by

$$\mathrm{rng}(\mathcal{I}) := \begin{cases} 0 & \text{if } \mathcal{I} = \mathrm{Ax}_A \ , \\ 1 & \text{if } \mathcal{I} = \bigvee_C^k \text{ or } \mathcal{I} = \mathrm{Rep} \ , \\ \mathrm{rng}(C) & \text{if } \mathcal{I} = \bigwedge_C \ , \\ 2 & \text{if } \mathcal{I} = \mathrm{Cut}_C \ . \end{cases}$$

For $h \in \mathcal{CH}_{\mathrm{BA}}$ we define

$$\mathrm{rng}(h) := \mathrm{rng}(\mathrm{tp}(h)) \ .$$

Definition 12.3. We extend the definition of bounding terms $\mathrm{bd}(h)$ and $\mathrm{ibd}(h)$ from $\mathcal{H}_{\mathrm{BA}}$ to $\mathcal{CH}_{\mathrm{BA}}$ in the following way by induction on the build-up of $h \in \mathcal{CH}_{\mathrm{BA}}$:

- If $h \in \mathcal{H}_{\mathrm{BA}}$ then the definition of $\mathrm{bd}(h)$ and $\mathrm{ibd}(h)$ are inherited from the definition of $\mathrm{bd}(h)$ resp. $\mathrm{ibd}(h)$ on $\mathcal{H}_{\mathrm{BA}}$.
- $\mathrm{bd}(\mathsf{I}_C^k h_0) := \mathrm{bd}(h_0)$, and $\mathrm{ibd}(\mathsf{I}_C^k h_0) := \mathrm{ibd}(h_0)$.
- $\mathrm{bd}(\mathsf{R}_C h_0 h_1) := \max\{\mathrm{bd}(h_0), \mathrm{bd}(h_1)\}$, and $\mathrm{ibd}(\mathsf{R}_C h_0 h_1) := \max\{\mathrm{ibd}(h_0), \mathrm{ibd}(h_1)\}$.
- $\mathrm{bd}(\mathsf{E} h_0) := \mathrm{bd}(h_0)$, and $\mathrm{ibd}(\mathsf{E} h_0) := \mathrm{ibd}(h_0)$.

Lemma 12.4. *Let $h \in \mathcal{CH}_{\mathrm{BA}}$.*

1. $\mathrm{bd}(h[j]) \leq \mathrm{bd}(h)$ *and* $\mathrm{ibd}(h[j]) \leq \mathrm{ibd}(h)$ *for all j.*
2. *If* $\mathrm{tp}(h) = \bigvee_C^k$ *then* $k \leq \mathrm{bd}(h)$.

Proof by induction on the build-up of h. ∎

Definition 12.5. For h a $\mathrm{BA}^\star$-derivation or $h \in \mathcal{CH}_{\mathrm{BA}}$, we define *the set of decorations of* h, $\mathrm{deco}(h)$, by induction on the build-up of h. $\mathrm{deco}(h)$ will be a finite set of $\mathcal{L}_{\mathrm{BA}}$-terms and formulae in BFOR. Let $h = \mathcal{I}h_0 \dots h_{n-1}$, where $\mathcal{I}$ ranges over $\mathrm{BA}^\star \cup \{\mathsf{I}_C^k, \mathsf{R}_C, \mathsf{E}\}$ (see also Remark 7.6). We define

$$\mathrm{deco}(h) := \mathrm{deco}(\mathcal{I}) \cup \bigcup_{i<n} \mathrm{deco}(h_i)$$

where

$$\begin{aligned}
\mathrm{deco}(\mathcal{I}) &:= \Delta(\mathcal{I}) \text{ for } \mathcal{I} = \mathrm{Ax}_\Delta, \textstyle\bigwedge_{A_0\wedge A_1}, \bigvee^k_{A_0\vee A_1} \\
\mathrm{deco}(\textstyle\bigwedge^y_{(\forall x)A}) &:= \{(\forall x)A, y\} \\
\mathrm{deco}(\textstyle\bigvee^t_{(\exists x)A}) &:= \{(\exists x)A, t\} \\
\mathrm{deco}(\mathrm{IND}_F^{y,t}) &:= \{F, \neg F_y(0), F_y(2^{|t|}), y, t\} \\
\mathrm{deco}(\mathrm{IND}_F^{y,n,i}) &:= \{F, \neg F_y(\underline{n}), F_y(\underline{n+2^i}), y, c_n\} \\
\mathrm{deco}(\mathrm{Cut}_C) &:= \{C\} \\
\mathrm{deco}(\mathsf{I}_C^k) &:= \{C, C[k], c_k\} \\
\mathrm{deco}(\mathsf{R}_C) &:= \{C\} \\
\mathrm{deco}(\mathsf{E}) &:= \emptyset \ .
\end{aligned}$$

If we would drop F from the definition of $\mathrm{deco}(\mathcal{I})$ in case $\mathcal{I} = \mathrm{IND}_F^{y,t}$ for example, only $F(0/y)$ and $F(t/y)$ would be included in the set of decorations, giving us no access to the intermediate steps needed to "compute" the induction. On the other hand, adding F to the set of decoration (F still contains the free variable y) gives us a generic way to access those intermediate steps.

Observation 12.6. *We have* $\Gamma(h) \subseteq \mathrm{deco}(h)$.

Definition 12.7. Let Φ be a set of $\mathcal{L}_{\mathrm{BA}}$-terms and formulae in BFOR, and let $K \in \mathbb{N}$ be a size parameter. With Φ_K we denote the set obtained by enlarging Φ by the set $\{c_i : 0 \le i \le K\}$ and the set of formulae and terms which result from formulae and terms in Φ by substituting constants from $\{c_i : 0 \le i \le K\}$ for some (possibly none, possibly all) of the free variables.

Lemma 12.8. *Let Φ be a set of $\mathcal{L}_{\mathrm{BA}}$-terms and formulae in* BFOR*, such that $\Phi \cap \mathrm{BFOR}$ is closed under negation and taking sub-formulae. Let $j, K \in \mathbb{N}$ and y be a variable.*

1. *If $j \le K$ and $C \in \Phi \cap \mathrm{BFOR}$, then $C[j] \in \Phi_K$.*
2. *If $h \in \mathrm{BA}^\star$ with $\mathrm{deco}(h) \subseteq \Phi$, and $j \le K$, then $\mathrm{deco}(h(\underline{j}/y)) \subseteq \Phi_K$.*
3. *$\Delta(\mathrm{tp}(h)) \subseteq \mathrm{deco}(h)_{\mathrm{bd}(h)}$ with the subscript understood in the sense of Definition 12.7 (it is needed, e.g., for $\mathrm{IND}_F^{y,n,i+1}$).*
4. *If $h \in \mathcal{CH}_{\mathrm{BA}}$ with $\mathrm{deco}(h) \subseteq \Phi$, and $j \le \mathrm{bd}(h)$, then $\mathrm{deco}(h[j]) \subseteq \Phi_{\mathrm{bd}(h)}$.*

Proof. For 4., consider the case that $h = \mathsf{R}_C h_0 h_1$, $\mathrm{tp}(h_1) = \bigvee^k_{\neg C}$ and $j = 0$, i.e. $h[0] = \mathsf{I}^k_C h_0$. By 3. we have $\neg C \in \Phi_{\mathrm{bd}(h_1)}$, hence $C \in \Phi_{\mathrm{bd}(h)}$. Also $k \leq \mathrm{bd}(h_1)$ by Lemma 12.4, 2. Hence, $C[k] \in \Phi_{\mathrm{bd}(h)}$ by 1. Now we compute

$$\mathrm{deco}(h[0]) = \{C, C[k], c_k\} \cup \mathrm{deco}(h_0) \subseteq \Phi_{\mathrm{bd}(h)} \cup \Phi = \Phi_{\mathrm{bd}(h)} \ .$$

□

Lemma 12.9. *For $h \in \mathcal{CH}_{\mathrm{BA}}$ we have that the cardinality of $\Gamma(h)$ is bounded above by $2 \cdot \mathrm{sz}(h)$.*

Proof. Let the cardinality of a set S be denoted by $\mathrm{card}(S)$. We observe that $\mathrm{card}(\Delta(\mathcal{I})) \leq 2$ for any $\mathcal{I} \in \mathrm{BA}^\infty$. Thus we can compute for $h = \mathcal{I} h_0 \dots h_{n-1} \in \mathcal{CH}_{\mathrm{BA}}$ by induction

$$\mathrm{card}(\Gamma(h)) \leq \mathrm{card}(\Delta(\mathcal{I})) + \sum_{i<n} \mathrm{card}(\Gamma(h_i)) \leq 2 + \sum_{i<n} 2 \cdot \mathrm{sz}(h_i) = 2 \cdot \mathrm{sz}(h) \ .$$

□

12.2 Search Problems Defined by Proof Notations

We identify the notation system $\mathcal{H}_{\mathrm{BA}}$ for BA^∞ with the abstract system of proof notations associated with it according to Observation 8.2. This way, the relation "$\rightarrow$" of "reduction to sub-derivation" is declared also on $\mathcal{H}_{\mathrm{BA}}$. For $s \in \mathbb{N}$ a size parameter and h a $\mathrm{BA}^\star$-derivation we define

$$\mathcal{H}^s_h := \{h' \in \mathcal{H}_{\mathrm{BA}} : \quad \begin{array}{l} |h'| \leq s \text{ and the axioms occurring in } h' \text{ as a} \\ \mathrm{BA}^\star\text{-derivation are substitution instances of} \\ \text{those occurring in } h \ .\} \end{array}$$

Then $\mathcal{H}^s_h$ is an s-bounded, abstract system of proof notations, because we observe that $h \in \mathcal{H}_{\mathrm{BA}}$ and $h \rightarrow h'$ implies $|h'| \leq |h|$. Furthermore, the relation $\mathcal{H}^s_h$ (for fixed h) is polynomial time computable, because deciding whether an inference symbol occurring in the notation $h' \in \mathcal{H}^s_h$ is a $\mathrm{BA}^\star$ axiom can be decided by matching it with one of the finitely many $\mathrm{BA}^\star$-axioms occurring in 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 $\mathcal{H}_{\mathrm{BA}}$ (see Definition 8.4 and Observation 8.5).

Definition 12.10. For $h \in \mathcal{CH}_{\mathrm{BA}}$ we define $\vartheta(h)(s) := \vartheta(\overline{h})(s)$.

For the reader's convenience let us mention that $\mathcal{CH}^s_h$ stands for $\mathcal{C}(\mathcal{H}^s_h)$, the notation system for cut-elimination on $\mathcal{H}^s_h$ according to Definition 7.5. Theorem 8.11 now reads as follows:

Corollary 12.11. *If $h' \in \mathcal{CH}^s_h$ and $h' \rightarrow h''$, then $\vartheta(h')(s) \geq \vartheta(h'')(s)$.*

Definition 12.12. We will define a local search problem L which is parameterised by

- a finite set Φ of $\mathcal{L}_{\mathrm{BA}}$-terms and formulae in BFOR, for which $\Phi \cap \mathrm{BFOR}$ is closed under negation and taking sub-formulae,
- a *"complexity class"* $\mathcal{C}$ given as a polynomial time computable set of $\mathcal{L}_{\mathrm{BA}}$-formulae which is assumed to be closed under taking subformulae and intensional equal formulae (usually $\mathcal{C} = \Sigma^{\mathrm{b}}_i$ for some i),
- a *size parameter* $s \in \mathbb{N}$,
- a $\mathrm{BA}^{\star}$-derivation h and a reduction level j, which together define an *initial value function* $h_{\bullet} \colon \mathbb{N} \to \mathcal{CH}^s_h$, $a \mapsto h_a := \mathsf{E}^j h(\underline{a}/x) := \underbrace{\mathsf{E} \ldots \mathsf{E}}_{j \text{ times}} h(\underline{a}/x)$,
- a formula $(\exists y)\varphi(x,y) \in \Phi$ with $\neg\varphi \in \mathcal{C}$,

such that, for $a \in \mathbb{N}$,

- $\Gamma(h_a) = \{(\exists y)\varphi(\underline{a}, y)\}$,
- $\mathcal{C}\text{-crk}(h_a) = 0$,
- $\mathrm{o}(h_a) = 2^{|a|^{O(1)}}$,
- $\vartheta(h_a)(s) = |a|^{O(1)}$,
- $\mathrm{deco}(h_a) \subseteq \Phi_a$.

We denote such a parametrisation by $L = \langle \Phi, \mathcal{C}, s, h, j, (\exists y)\varphi(x,y) \rangle$.

An instance of L is given by $a \in \mathbb{N}$ in the following way:

- The set of possible solutions $F(a) \in \mathfrak{P}_{\mathrm{fin}}(\mathcal{CH}^s_h)$ is given as the set of those $h' \in \mathcal{CH}^s_h$ which satisfy:
  - i) $\Gamma(h') \subseteq \{(\exists y)\varphi(\underline{a}, y)\} \cup \Delta$ for some $\Delta \subseteq \mathcal{C} \cup \neg\mathcal{C}$ such that all $A \in \Delta$ are closed and false,
  - ii) $\mathcal{C}\text{-crk}(h') = 0$,
  - iii) $\mathrm{o}(h') \leq \mathrm{o}(h_a)$,
  - iv) $\vartheta(h')(s) \leq \vartheta(h_a)(s)$,
  - v) $\mathrm{bd}(h') \leq \mathrm{bd}(h_a)$ and $\mathrm{ibd}(h') \leq \mathrm{ibd}(h_a)$,
  - vi) $\mathrm{deco}(h') \subseteq \Phi_{\max(a, \mathrm{bd}(h_a))}$;
- The initial value is given by $i(a) := h_a$;
- the cost function is defined as $c(a, h') := \mathrm{o}(h')$; and

- the neighbourhood function is given by

$$N(a,h') := \begin{cases} h'[j] & \text{if } \mathrm{tp}(h') = \bigwedge_C, j < \mathrm{rng}(C) \text{ and } C[j] \text{ false} \ , \\ h'[0] & \text{if } \mathrm{tp}(h') = \bigvee_C^i \text{ and } C \neq (\exists y)\varphi(\underline{a},y) \\ & \quad \text{or } \mathrm{tp}(h') = \bigvee^i_{(\exists y)\varphi(\underline{a},y)} \text{ and } \varphi(\underline{a},\underline{i}) \text{ false} \ , \\ h'[0] & \text{if } \mathrm{tp}(h') = \mathrm{Cut}_C \text{ and } C \text{ false} \ , \\ h'[1] & \text{if } \mathrm{tp}(h') = \mathrm{Cut}_C \text{ and } C \text{ true} \ , \\ h'[0] & \text{if } \mathrm{tp}(h') = \mathrm{Rep} \ , \\ h' & \text{otherwise} \ . \end{cases}$$

(Observe that the just defined neighbourhood function is a multifunction due to case $\bigwedge_C$.)

Lemma 12.13. *For each $a \in \mathbb{N}$ the above defined $\langle F(a), i(a), c(a,\cdot), N(a,\cdot)\rangle$ is indeed an instance of a local search problem, i.e. we have:*

1. *$h_a \in F(a)$,*
2. *$h \in F(a)$ and $N(a,h) \neq h$ implies $N(a,h) \in F(a)$ and $\mathrm{o}(N(a,h)) < \mathrm{o}(h)$.*

Proof. The first claim is obvious by definition. For the second claim let $h \in F(a)$ with $h' := N(a,h) \neq h$. Then we have to show $h' \in F(a)$ and $\mathrm{o}(h') < \mathrm{o}(h)$. As $h' = h[j]$ for some $j < \mathrm{rng}(h)$ by construction, we obviously have $\mathrm{o}(h') < \mathrm{o}(h)$ and $h \to h'$.

To show $h' \in F(a)$, we consider i)–vi) of the definition of $h' \in F(a)$: ii) is clear; iii) is obvious; for iv) observe that $h \to h'$ implies $\vartheta(h')(s) \leq \vartheta(h)(s)$ by Corollary 12.11; for v) observe $\mathrm{bd}(h') \leq \mathrm{bd}(h)$ and $\mathrm{ibd}(h') \leq \mathrm{ibd}(h)$ by Lemma 12.4; for vi) observe that $j < \mathrm{rng}(h)$ implies $\mathrm{deco}(h') \subseteq (\Phi_{\max(a,\mathrm{bd}(h_a))})_{\mathrm{bd}(h)} = \Phi_{\max(a,\mathrm{bd}(h_a))}$ by Lemma 12.8, 4., because $\mathrm{rng}(h) \leq \mathrm{bd}(h)$ and $\mathrm{bd}(h) \leq \mathrm{bd}(h_a)$. And finally for i), we first observe that the first condition, which says $\Gamma(h')\setminus\{(\exists y)\varphi(\underline{a},y)\}$ is a subset of $\mathcal{C}\cup\neg\mathcal{C}$ consisting only of closed formulae, is obviously satisfied as $\mathcal{C}\text{-crk}(h') = 0$. For the second condition of i) let $\mathcal{I} := \mathrm{tp}(h)$. By Proposition 7.2 we have that

$$\Gamma(h[j]) \subseteq \approx_{\mathbb{N}}\Big(\Gamma(h) \cup \Delta_j(\mathcal{I})\Big) \ .$$

Thus it is enough to show that $\bigvee\Delta_j(\mathcal{I})$ is false.

- $\mathcal{I} = \bigwedge_C$: $\Delta_j(\mathcal{I}) = \{C[j]\}$ and $C[j]$ is false by construction.
- $\mathcal{I} = \bigvee^i_C$: then $j = 0$. If $C \neq (\exists y)\varphi(\underline{a},y)$, then $\Delta_0(\mathcal{I}) = \{C[i]\}$. Now C is false by i) of $h \in F(a)$, hence $C[i]$ must be false as well. Otherwise, $\Delta_0(\mathcal{I}) = \{\varphi(\underline{a},\underline{i})\}$, and $\varphi(\underline{a},\underline{i})$ is false by construction.
- $\mathcal{I} = \mathrm{Cut}_C$: If $j = 0$, then $\Delta_0(\mathcal{I}) = \{C\}$ and C is false by construction. Otherwise, $j = 1$, then $\Delta_1(\mathcal{I}) = \{\neg C\}$ and $\neg C$ is false by construction.

- $\mathcal{I} = \mathrm{Rep}$: then $j = 0$ and $\Delta_0(\mathcal{I}) = \emptyset$ and nothing is to be shown.

$\square$

In the following, P denotes the class of predicates which can be decided in polynomial time, and FP the class of functions which can be computed in polynomial time. We denote the relativisations of P and FP to some oracle from a given class $\mathcal{C}$ by $\mathrm{P}^{\mathcal{C}}$ respectively $\mathrm{FP}^{\mathcal{C}}$. Furthermore, $\mathrm{FP}^{\mathcal{C}}[\mathrm{wit}, g(n)]$ denotes the class of functions which can be computed by a Turing machine in polynomial time, where the Turing Machine is allowed to ask witness queries to some oracle in $\mathcal{C}$ such that the number of witness queries asked in computations on inputs of length n is bounded by $g(n)$. An overview and more in depth discussion of these definitions can be found in Krajíček's encyclopedical book [Kra95].

Proposition 12.14 (Complexity of L)**.** *Let $L = \langle \Phi, \mathcal{C}, s, h, j, (\exists y)\varphi(x, y)\rangle$ be a local search problem as defined in Definition 12.12, with associated set of possible solutions F, initial value function i, cost function c and neighbourhood function N.*

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

Proof. First observe that the functions $a \mapsto i(a) = h_a$, $a \mapsto \mathrm{bd}(h_a)$, $a \mapsto \mathrm{ibd}(h_a)$, $a \mapsto \mathrm{o}(h_a)$, $a \mapsto \vartheta(h_a)$, and $a \mapsto \mathrm{deco}(h_a)$ are polynomial time computable.

Furthermore, the relations $\mathcal{CH}^s_h$, $\mathcal{C}$-$\mathrm{crk}(h') = 0$, $\mathrm{bd}(h') \leq m$, $\mathrm{ibd}(h') \leq m$ and $\mathrm{deco}(h') \subseteq \Phi_m$ are polynomial time computable. Thus, also $\mathrm{o}(h')$ for $h' \in \mathcal{CH}^s_h$ with $\mathrm{ibd}(h') \leq \mathrm{ibd}(h_a)$ is polynomial time computable using Observation 11.10 and the remark following the Observation. Hence $c \in \mathrm{FP}$.

Also, the functions $\mathrm{tp}(h')$ and $h'[i]$ are polynomial time computable on $\mathcal{CH}^s_h$, which shows $N \in \mathrm{FP}^{\mathcal{C}}[\mathrm{wit}, 1]$. N is only a multifunction because of the $\bigwedge_C$-case.

For $F \in \mathrm{P}^{\mathcal{C}}$ observe that $\Gamma(h') \subseteq \mathrm{deco}(h') \subseteq \Phi_{\max(a, \mathrm{bd}(h_a))}$, hence the first condition of $h' \in F(a)$ that all formulae in $\Gamma(h') \setminus \{(\exists y)\varphi(\underline{a}, y)\}$ are in $\mathcal{C} \cup \neg\mathcal{C}$ and false, is a property in $\mathrm{P}^{\mathcal{C}}$. $\square$

A *polynomial local search (*PLS*) problem* [JPY88] is a local search problem which in addition satisfies that the initial value, cost and neighbourhood functions are polynomial time computable, and that the set of possible solutions is polynomial time computable and polynomially bounded, which means that for any instance of the search problem of length n, the length of any possible solution is bounded polynomially in n. The relativisation of a PLS problem to a class $\mathcal{C}$ is given by relativising its set of possible solutions, and initial value, cost and neighbourhood function, to $\mathcal{C}$. The class of such relativisations is denoted by $\mathrm{PLS}^{\mathcal{C}}$.

Proposition 12.15 (Properties of L)**.** *Let $L = \langle \Phi, \mathcal{C}, s, h, j, (\exists y)\varphi(x, y)\rangle$ be a local search problem as defined in Definition 12.12, with associated set of possible solutions F, initial value function i, cost function c and neighbourhood function N. Let f be the multifunction defined by φ: $f(x) = y$ iff $\varphi(x, y)$.*

1. *If $N(a,h) = h$ then $\mathrm{tp}(h)$ is of the form $\bigvee^{i}_{(\exists y)\varphi(\underline{a},y)}$ such that $\varphi(\underline{a},\underline{i})$ is true. Obviously, i can be extracted from $\bigvee^{i}_{(\exists y)\varphi(\underline{a},y)}$ by a polynomial time computable operation. We say that f* is computed by *L.*

2. *The search problem L in general defines a search problem in $\mathrm{PLS}^{\mathcal{C}}$, assuming that we turn the neighbourhood multifunction into a function, which can easily be achieved by using an intermediate $\mathrm{PLS}^{\mathcal{C}}$ search problem which searches for the smallest witness in 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 by iterating the neighbourhood function, is of polynomial length with terms of polynomial size, thus $f \in \mathrm{FP}^{\mathcal{C}}[wit, \mathrm{o}(h_a)]$.*

□

Observe that it is common for the treatment of multifunctions that we do not require that all possible values of a multifunction appear as outputs of some computations (cf. [Kra95, Section 6.3]).

Proposition 12.16. *Assume that $(\exists y)\varphi(x,y) \in \mathrm{BFOR}$, $\varphi(x,y) \in \Pi^{\mathrm{b}}_i$, h is a $\mathrm{BA}^{\star}$-derivation with $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) \subseteq \{(\exists y)\varphi(x,y)\}$, and Σ^{b}_i-$\mathrm{gcrk}(h) \leq j$.*

Let Φ be $\mathrm{deco}(h)$ together with $\mathrm{deco}(h) \cap \mathrm{BFOR}$ closed under negation and taking sub-formulae, $\mathcal{C} := \Sigma^{\mathrm{b}}_i$, $s := |h|$, and $h_a := \mathsf{E}^j h(\underline{a}/x)$. Then the following holds:

1. *$(\exists y)\varphi(x,y) \in \Phi$, $\neg\varphi \in \mathcal{C}$, and $h_a \in \mathcal{C}\mathcal{H}^s_h$.*
2. *$\Gamma(h_a) = \{(\exists y)\varphi(\underline{a},y)\}$.*
3. *$\mathcal{C}$-$\mathrm{crk}(h_a) = 0$*
4. *$\mathrm{o}(h_a) \leq 2_j(\mathrm{o}(h(\underline{a}/x)))$*
5. *$\vartheta(h_a)(s) \leq 2_{j-1}((s+2)\cdot \mathrm{o}(h(\underline{a}/x)))$ if $j \geq 1$*
6. *$\mathrm{deco}(h_a) \subseteq \Phi_a$*
7. *If either $j \leq 2$ and $\mathrm{o}(h(\underline{a}/x)) = O(|a|_j)$, or $\mathrm{o}(h(\underline{a}/x)) = O(|a|_{1+j})$, then $L = \langle \Phi, \mathcal{C}, s, h, j, (\exists y)\varphi(x,y)\rangle$ defines a local search problem according to Definition 12.12.*

Proof. 1., 3. and 4. are obvious by definition. 6. follows from Lemma 12.8.2. For 2. we compute $\Gamma(h_a) = \Gamma(h(\underline{a}/x)) = \Gamma(h)(\underline{a}/x) = \{(\exists y)\varphi(\underline{a},y)\}$, using that $\Gamma(\mathsf{E}h(\underline{a}/x)) = \Gamma(h(\underline{a}/x))$, as well as Lemma 11.6.

For 5. we prove

$$\vartheta(\mathsf{E}^{k+1}h(\underline{a}/x))(s) \leq 2_k((s+2)\cdot \mathrm{o}(h(\underline{a}/x))) \tag{7}$$

by induction on k. For the case $k=0$ we compute

$$\vartheta(\mathsf{E}h(\underline{a}/x))(s) = \mathrm{o}(h(\underline{a}/x))\cdot(\vartheta(h(\underline{a}/x))(s)+2) = \mathrm{o}(h(\underline{a}/x))\cdot(s+2)$$

For the induction step we compute

$$\begin{aligned}
\vartheta(\mathsf{E}^{k+2}h(\underline{a}/x))(s) &= \mathrm{o}(\mathsf{E}^{k+1}h(\underline{a}/x)) \cdot (\vartheta(\mathsf{E}^{k+1}h(\underline{a}/x)) + 2) \\
&\leq 2_{k+1}(\mathrm{o}(h(\underline{a}/x))) \cdot (\vartheta(\mathsf{E}^{k+1}h(\underline{a}/x)) + 2) \\
&\overset{i.h.}{\leq} 2_{k+1}(\mathrm{o}(h(\underline{a}/x))) \cdot (2_k((s+2) \cdot \mathrm{o}(h(\underline{a}/x))) + 2) \\
&\leq 2_{k+1}(\mathrm{o}(h(\underline{a}/x))) \cdot 2_{k+1}((s+1) \cdot \mathrm{o}(h(\underline{a}/x))) \\
&\leq 2_{k+1}((s+2) \cdot \mathrm{o}(h(\underline{a}/x))) \ .
\end{aligned}$$

The last but one inequality holds because $2_k((s+2) \cdot u + 2) \leq 2_{k+1}((s+1) \cdot u)$ for $s \geq 2$, which is satisfied for $s = |h|$.

With 1., 2., 3. and 6. in place for 7. we are left to show $\mathrm{o}(h_a) = 2^{|a|^{O(1)}}$ and $\vartheta(h_a)(s) = |a|^{O(1)}$. If $j \leq 2$ and $\mathrm{o}(h(\underline{a}/x)) = O(|a|_j)$, then we have

$$\mathrm{o}(h_a) \overset{4.}{\leq} 2_j(\mathrm{o}(h(\underline{a}/x))) = 2_j(O(|a|_j)) = 2_2(O(|a|_2)) = 2^{|a|^{O(1)}}$$

using for last equation that $2^{O(|n|)} = n^{O(1)}$. Furthermore, for $j = 0$ we have $\vartheta(h_a)(s) = s = |a|^{O(1)}$, and, for $j \geq 1$,

$$\vartheta(h_a)(s) \overset{5.}{\leq} 2_{j-1}((s+2) \cdot \mathrm{o}(h(\underline{a}/x))) = 2_{j-1}(O(|a|_j)) = 2_1(O(|a|_2)) = |a|^{O(1)} \ .$$

If $\mathrm{o}(h(\underline{a}/x)) = O(|a|_{1+j})$ then

$$\mathrm{o}(h_a) \overset{4.}{\leq} 2_j(\mathrm{o}(h(\underline{a}/x))) = 2_j(O(|a|_{1+j})) = O(a)$$

using for last equation that $c|n| \leq n$ for $n \geq 4c^2$, and

$$\vartheta(h_a)(s) \overset{5.}{\leq} 2_{j-1}((s+2) \cdot \mathrm{o}(h(\underline{a}/x))) = 2_{j-1}(O(|a|_{1+j})) = O(|a|) \ .$$

$\square$

We will now give new proofs for known characterisations of definable multifunctions of Bounded Arithmetic theories. Let f be a Σ^{b}_j-definable multifunction of S^i_2, $j > 0$. Then f is defined by some $\psi \in \Sigma^{\mathrm{b}}_j$, that is $f(x) = y$ iff $\psi(x, y)$, such that $\mathrm{S}^i_2 \vdash (\forall x)(\exists y)\psi(x, y)$. By Parikh's Theorem (cf. [Bus86, Section 4.7]) there is some term $t(x)$ such that $\mathrm{S}^i_2 \vdash (\forall x)(\exists y \leq t(x))\psi(x, y)$. Now, $\psi(x, y)$ is of the form $(\exists z \leq s(x, y))\chi(x, y, z)$ with $\chi \in \Pi^{\mathrm{b}}_{j-1}$. Consider $\varphi(x, y) \equiv \chi(x, (y)_0, (y)_1)$, then $\varphi \in \Pi^{\mathrm{b}}_{j-1}$, $\mathrm{S}^i_2 \vdash (\forall x)(\exists y)\varphi(x, y)$, and the values of f are exactly projections of values of the multifunction g defined by φ. As projections are polynomial time computable, it will be enough to characterise g in order to characterise f.

Therefore, in the following proofs we will just consider $\mathrm{S}^i_2 \vdash (\forall x)(\exists y)\varphi(x, y)$ with $\varphi \in \Pi^{\mathrm{b}}_{j-1}$ and $(\exists y)\varphi(x, y) \in \mathrm{BFOR}$ to characterise the Σ^{b}_j-definable multifunctions of S^i_2.

Theorem 12.17 (Krajíček [Kra93])**.** *Let $i \geq 2$. The Σ^{b}_i-definable multifunctions of S^{i-1}_2 are in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}[wit, O(\log n)]$.*

Proof. Let $i \geq 2$ and assume that $\mathrm{S}_2^{i-1} \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in$ BFOR, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By Theorem 11.11 we obtain some $\mathrm{BA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$, $\Sigma^{\mathrm{b}}_{i-1}$-gcrk$(h) = 0$, and $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$.

Let Φ be deco(h) together with deco$(h) \cap$ BFOR closed under negation and taking sub-formulae, $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$, $s := |h|$, and $h_a := h(\underline{a}/x)$. Proposition 12.16.7 shows that in this case $L = \langle \Phi, \mathcal{C}, s, h, 0, (\exists y)\varphi(x,y)\rangle$ defines a local search problem according to Definition 12.12, as $||a||$ is the same as $|a|_2$.

As $\mathrm{o}(h_a) = O(||a||)$, Proposition 12.15.3 shows that the multifunction defined by φ is in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}[\mathrm{wit}, O(\log n)]$, using the common notation in computational complexity that n denotes the size of the input a, i.e. $\mathrm{o}(h_a) = O(\log n)$. ∎

Theorem 12.18 (Buss [Bus86])**.** *Let $i > 0$. The Σ^{b}_i-definable functions of S_2^i are in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}$.*

Proof. Let $i > 0$ and assume that $\mathrm{S}_2^{i} \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in$ BFOR, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By Theorem 11.11 we obtain some $\mathrm{BA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$, Σ^{b}_{i}-gcrk$(h) = 0$, and $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$.

Let Φ be deco(h) together with deco$(h) \cap$ BFOR closed under negation and taking sub-formulae, $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$, $s := |h|$, and $h_a := \mathsf{E}h(\underline{a}/x)$. Proposition 12.16.7 shows that in this case $L = \langle \Phi, \mathcal{C}, s, h, 1, (\exists y)\varphi(x,y)\rangle$ defines a local search problem according to Definition 12.12, because $\mathcal{C}$-crk$(h_a) = \mathcal{C}$-gcrk$(h) \leq \Sigma^{\mathrm{b}}_{i}$-gcrk$(h) + 1 = 1$. Observe that $\mathcal{C}$-crk(h_a) is the denoted cut-rank on $\mathcal{H}_{\mathrm{BA}}$, where $\mathcal{C}$-gcrk(h) is the generalised cut-rank on $\mathrm{BA}^\star$.

As $\mathrm{o}(h_a) = |a|^{O(1)}$, Proposition 12.15.3 shows that the multifunction defined by φ 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_2^i 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. ∎

Theorem 12.19 (Buss and Krajíček [BK94])**.** *Let $i > 0$. The Σ^{b}_i-definable multifunctions of S_2^{i+1} are projections of solutions to problems in $\mathrm{PLS}^{\Sigma^{\mathrm{b}}_{i-1}}$.*

Proof. Let $i > 0$ and assume that $\mathrm{S}_2^{i+1} \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in$ BFOR, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By Theorem 11.11 we obtain some $\mathrm{BA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$, $\Sigma^{\mathrm{b}}_{i+1}$-gcrk$(h) = 0$, and $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$.

Let Φ be deco(h) together with deco$(h) \cap$ BFOR closed under negation and taking sub-formulae, $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$, $s := |h|$, and $h_a := \mathsf{E}\mathsf{E}h(\underline{a}/x)$. Proposition 12.16.7 shows that in this case $L = \langle \Phi, \mathcal{C}, s, h, 2, (\exists y)\varphi(x,y)\rangle$ defines a local search problem according to Definition 12.12, because $\mathcal{C}$-crk$(h_a) = \mathcal{C}$-gcrk$(h) \leq \Sigma^{\mathrm{b}}_{i+1}$-gcrk$(h) + 2 = 2$.

Proposition 12.15.2 shows that this defines a search problem in $\mathrm{PLS}^{\Sigma^{\mathrm{b}}_{i-1}}$. ∎

Theorem 12.20 (Pollett [Pol99])**.** *Let $i \geq 1$, $j \geq 0$, $k \geq 1$. The $\Sigma^{\mathrm{b}}_{i+1}$-definable multifunctions of $\Sigma^{\mathrm{b}}_{i+j}$-$\mathrm{L}^{k+j}$IND are in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_i}[wit, 2_j(O(\log^{k+j} n))]$.*

Proof. Let $i \geq 1$, $j \geq 0$, $k \geq 1$ and assume that $\Sigma^{\mathrm{b}}_{i+j}$-$\mathrm{L}^{k+j}\mathrm{IND} \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in \mathrm{BFOR}$, $\varphi \in \Pi^{\mathrm{b}}_i$. By Theorem 11.11 we obtain some $\mathrm{BA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$, $\Sigma^{\mathrm{b}}_{i+j}$-gcrk$(h) = 0$, and $\mathrm{o}(h(\underline{a}/x)) = O(|a|_{1+k+j})$.

Let Φ be deco(h) together with deco$(h) \cap \mathrm{BFOR}$ closed under negation and taking sub-formulae, $\mathcal{C} := \Sigma^{\mathrm{b}}_i$, $s := |h|$, and $h_a := \mathsf{E}^j h(\underline{a}/x)$. Proposition 12.16.7 shows that in this case $L = \langle \Phi, \mathcal{C}, s, h, j, (\exists y)\varphi(x,y)\rangle$ defines a local search problem according to Definition 12.12, because $\mathcal{C}$-crk$(h_a) = \mathcal{C}$-gcrk$(h) \leq \Sigma^{\mathrm{b}}_{i+j}$-gcrk$(h) + j = j$.

As $\mathrm{o}(h_a) = O(||a||)$, Proposition 12.15.3 shows that the multifunction defined by φ is in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_i}[\mathrm{wit}, 2_j(O(\log^{k+j} n))]$. ◻

Finally, we draw another conclusion which has not been covered by the literature so far.

Theorem 12.21. *Let $i \geq 1$, $j \geq 0$, $k \geq 1$. The Σ^{b}_i-definable multifunctions of $\Sigma^{\mathrm{b}}_{i+j}$-$\mathrm{L}^{k+j}\mathrm{IND}$ are projections of solutions to problems in $\mathrm{PLS}^{\Sigma^{\mathrm{b}}_{i-1}}$ where the cost-function is bounded by $2_{j+1}(O(\log^{k+j} n))$, n being the size of the input.*

Proof. Let $i \geq 1$, $j \geq 0$, $k \geq 1$ and assume that $\Sigma^{\mathrm{b}}_{i+j}$-$\mathrm{L}^{k+j}\mathrm{IND} \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in \mathrm{BFOR}$, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By Theorem 11.11 we obtain some $\mathrm{BA}^\star$-derivation h such that $\mathrm{FV}(h) \subseteq \{x\}$, $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$, $\Sigma^{\mathrm{b}}_{i+j}$-gcrk$(h) = 0$, and $\mathrm{o}(h(\underline{a}/x)) = O(|a|_{1+k+j})$.

Let Φ be deco(h) together with deco$(h) \cap \mathrm{BFOR}$ closed under negation and taking sub-formulae, $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$, $s := |h|$, and $h_a := \mathsf{E}^{j+1} h(\underline{a}/x)$. Proposition 12.16.7 shows that in this case $L = \langle \Phi, \mathcal{C}, s, h, j+1, (\exists y)\varphi(x,y)\rangle$ defines a local search problem according to Definition 12.12, because $\mathcal{C}$-crk$(h_a) = \mathcal{C}$-gcrk$(h) \leq \Sigma^{\mathrm{b}}_{i+j}$-gcrk$(h) + j + 1 = j + 1$.

As in Proposition 12.15.2 we obtain that L is a $\mathrm{PLS}^{\Sigma^{\mathrm{b}}_{i-1}}$-problem, but now the cost function is bounded by $\mathrm{o}(h_a) \leq 2_{j+1}(\mathrm{o}(h(\underline{a}/x))) = 2_{j+1}(O(|a|_{1+k+j}))$. ◻

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 hitherto uncharacterised classes.

Acknowledgements

The second author is grateful to Mohammad Javad A. Larijani, Ali Enayat, Iraj Kalantari, Morteza Moniri and Massoud Pourmahdian for organising *IPM Logic*

Conference 2007. Their great hospitality during his stay in Tehran added to the overall wonderful experience. Some of the results in this article were presented at the conference by the second author in one of his two invited lectures.

The authors would like to thank the anonymous referees for their valuable comments which helped improving the article considerably.

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

References

[AJ05] Klaus Aehlig and Felix Joachimski. Continuous normalization for the lambda-calculus and Gödel's T. *Annals of Pure and Applied Logic*, 133(1–3):39–71, May 2005.

[AS00] Klaus Aehlig and Helmut 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.

[Bec01] Arnold Beckmann. Exact bounds for lengths of reductions in typed λ-calculus. *Journal of Symbolic Logic*, 66(3):1277–1285, 2001.

[Bec03] Arnold Beckmann. Dynamic ordinal analysis. *Arch. Math. Logic*, 42:303–334, 2003.

[Bec06] Arnold Beckmann. Generalised dynamic ordinals—universal measures for implicit computational complexity. In *Logic Colloquium '02*, volume 27 of *Lect. Notes Log.*, pages 48–74. Assoc. Symbol. Logic, La Jolla, CA, 2006.

[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.

[Bus04] Samuel 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.

[Gen35a] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[Gen35b] Gerhard Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39:405–431, 1935.

[Göd58] Kurt Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunkts. *Dialectica*, 12:280–287, 1958.

[JPY88] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988.

[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.

[Kra95] Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge University Press, Heidelberg/New York, 1995.

[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.

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

[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.

[Sch51] Kurt Schütte. Die unendliche Induktion in der Zahlentheorie. *Mathematische Annalen*, 122:369–389, 1951.

[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.