# Propositional Logic for Circuit Classes

Klaus Aehlig[1]* and Arnold Beckmann[2]

[1] Department of Computer Science
University of Toronto
10 King's College Road, Toronto, ON M5S 3G4, Canada
[2] Department of Computer Science
University of Wales Swansea
Singleton Park, Swansea, SA2 8PP, United Kingdom

**Abstract.** By introducing a parallel extension rule that is aware of independence of the introduced extension variables, a calculus for quantified propositional logic is obtained where heights of derivations correspond to heights of appropriate circuits. Adding an uninterpreted predicate on bit-strings (analog to an oracle in relativised complexity classes) this statement can be made precise in the sense that the height of the most shallow proof that a circuit can be evaluated is, up to an additive constant, the height of that circuit.
The main tool for showing lower bounds on proof heights is a variant of an iteration principle studied by Takeuti. This reformulation might be of independent interest, as it allows for polynomial size formulae in the relativised language that require proofs of exponential height.

## 1 Introduction and Related Work

In systems like "extended Frege" there is a rule that allows one to introduce a new variable by a defining clause $p \leftrightarrow A$. If several variables are to be introduced, several instances of this rule have to be used. This holds regardless of the presence or absence of dependencies between these variables.

However, such dependencies are known to make a big difference in the world of computation. Both, uniform $\mathsf{AC}^0$ and polynomial time can be described by families of polynomial size circuits. Nevertheless, $\mathsf{AC}^0$ has much smaller computational power. The reason is that the nodes in $\mathsf{AC}^0$ circuits are constrained to be arranged in a finite number of layers.

Even though various propositional calculi are known for small complexity classes, none reflects correctly the height of circuits. We suggest a calculus that has this property and can serve as a framework for investigating the "circuit strength" of various propositional calculi and small complexity classes; the latter come in via propositional translations of appropriate theories of Bounded Arithmetic [4, 6].

* Supported by DFG grant Ae 102/1-1. Part of the work was carried out while affiliated with University of Wales Swansea and supported by EPSRC grant EP/D03809X/1.
Corresponding author, email klausa@cs.toronto.edu

We consider relativised circuit classes [15]. That is, our circuits will not only contain logical gates but also gates that query an oracle. There are several motivations for doing so. Hardly any separations of unrelativised small complexity classes are known, but separating relativised circuit classes is straightforward. So, in order to precisely state that the calculus adequately reflects the differences between different circuit classes, we need to consider the relativised forms; an absolute separation of the levels of the $\mathrm{AC}^k$ hierarchy seems out of reach at the moment. Moreover, this calculus is intended as a target for propositional translations of theories of Bounded Arithmetic. Following standard proof theoretical practise [11], a better classification of theories can be obtained for the variants relativised to an uninterpreted predicate.

Quantified propositional logic in relation to complexity classes and bounded arithmetic has been studied by Krajíček and Pudlák [9]. They introduced various dag-like ($G_1$, $G_2$,..., $G$) and tree-like systems ($G_1^*$, $G_2^*$,..., $G^*$). Cook and Morioka (in a slightly modified setting) identified [5] $G_0$ and $G_0^*$ which relate to $\mathsf{NC}^1$. One motivation for the study of restricted propositional proof systems is the relation to (weak) theories of bounded arithmetic [4, 6]. For various complexity classes, corresponding proof systems [10, 12] have been identified. However, a unifying framework for the propositional systems still seems to be missing. We suggest a calculus which is flexible enough to allow for embedding of various theories, but is still strict enough that the *height* of proofs is a meaningful measure.

Studying the height of proofs is a standard approach in ordinal informative proof theory [11] and has been adopted to the Bounded Arithmetic setting by Beckmann [1]. It was also implicitly used by Krajíček [8].

Our research presented here investigates a particular form of the iteration principle. An important source for this has been Takeuti's investigations [14] where he obtained separations of some versions of bounded arithmetic theories [3] related to circuit complexity classes. A different form of the iteration principle has been introduced and studied by Buss and Krajíček [2] to obtain separations between bounded arithmetic theories related to (relativised) polynomial time and polynomial local search (PLS).

This article is organized as follows. In Section 2 we define our calculus $\mathsf{AC}^0$-Tait. In Section 3 we consider the formula expressing that a circuit of height $h$ can be evaluated. We note that this formula can be proven by a proof of height $h+\mathcal{O}(1)$. For the other direction we need a few preparations, that are interesting results in their own right. First we study in Section 4 a formula expressing that a function can be iterated $\ell$ times; we show that a proof of this formula requires height at least $\ell$. As the iteration formula is a polynomial size $\Sigma_1^q(\alpha)$-formula and can express exponentially long iterations, this establishes an exponential lower bound for the calculus with cuts on arbitrary quantifier-free formulae. In Section 5 we study a version of the calculus, extended with cuts, and prove cut-elimination. The cut-lemma will allow us to transform a proof that a particular circuit can be evaluated into a proof of the iteration principle without increasing the height by more than a constant. Putting things together in Section 6 shows

that there are circuits of height $h$ where a proof that they can be evaluated requires height at least $h - \mathcal{O}(1)$.

## 2 Quantified Propositional Logic and Definition of the Calculus

In this section we will introduce our calculus. It will be in the style of Tait [13], that is, roughly, one-sided sequent calculus. Following standard simplifications, a sequent is a set of formulae, and negation is an operation on formulae, not a logical symbol.

**Definition 1.** The *atoms of propositional logic* are variables $p, q, r, \ldots$, their negations $\bar{p}, \bar{q}, \bar{r}, \ldots$, as well as the constants T and F for truth and falsity.

The set of all propositional atoms is denoted by $\mathcal{A}$ and we use $\wp$ to range over elements of $\mathcal{A}$.

The set of *quantified propositional formulae* $A, B, C, \ldots$ is built up from the atoms of propositional logic and parameter $\alpha_k \wp_1 \ldots \wp_k$ and negated parameter $\bar{\alpha}_k \wp_1 \ldots \wp_k$ where $\wp_1, \ldots, \wp_k$ are propositional atoms, by conjunctions $\bigwedge_k A_1 \ldots A_k$ and disjunctions $\bigvee_k A_1 \ldots A_k$, and universal $\forall_k p_1 \ldots p_k A$ and existential $\exists_k p_1 \ldots p_k A$ quantification.

Here $k \geq 1$ is a natural number on the meta level. The variables $p_1, \ldots, p_k$ and their negations $\bar{p}_1, \ldots, \bar{p}_k$ are bound in $\forall_k p_1 \ldots p_k A$ and $\exists_k p_1 \ldots p_k A$.

Syntactical equality is denoted by $\equiv$. A quantified propositional formula without any quantifications is called a *propositional formula*. We use the expression *purely propositional formula* for a propositional formula, if we want to emphasise that it is not quantified.

We write $\wedge$ and $\vee$ for $\bigwedge_2$ and $\bigvee_2$, respectively. We use $A \wedge B$ and $A \vee B$ as abbreviations for $\wedge AB$ and $\vee AB$, respectively, if there is no danger of confusion. Also, parentheses may be used to facilitate reading or to disambiguate these abbreviations.

By induction on $A$ a formula $\neg A$ is defined according to the de Morgan rules in the obvious way, e.g., $\neg p \equiv \bar{p}$, $\neg \bar{p} \equiv p$, $\neg(\alpha_k \wp_1 \ldots \wp_k) \equiv \bar{\alpha}_k \wp_1 \ldots \wp_k$, $\neg(\bigwedge_k A_1 \ldots A_k) \equiv \bigvee_k (\neg A_1) \ldots (\neg A_k)$, and so on. A simple induction on $A$ shows that $\neg\neg A \equiv A$.

If $A$ is a quantified propositional formula, $\vec{p}$ are pairwise disjoint propositional variables, and $\vec{B}$ are quantified propositional formulae, then by $A[\vec{B}/\vec{p}]$ we denote the simultaneous capture-free substitution of all $p_i$ by $B_i$ and of all $\bar{p}_i$ by $\neg B_i$.

We use the notation $A(\vec{p})$ to distinguish certain variables of $A$, in order to be able to use $A(\vec{B})$ as a shorthand for the substitution $A[\vec{B}/\vec{p}]$. This notation does not imply that these variables actually do occur free and the list $\vec{p}$ does not necessarily exhaust all the free variables of $A$.

We use $\Gamma, \Delta, \ldots$ to denote finite sets of formulae.

**Definition 2.** The *propositional rules* are the following rules.

$$\frac{}{\Gamma, p, \bar{p}} \qquad \frac{}{\Gamma, \mathrm{T}} \qquad \frac{}{\Gamma, \alpha_k(\wp_1, \ldots, \wp_k), \bar{\alpha}_k(\wp_1, \ldots, \wp_k)}$$

$$\frac{\Gamma, A_i}{\Gamma, \bigvee_k A_1 \ldots A_k} \qquad \frac{\ldots \Gamma, A_i \ldots \quad (1 \leq i \leq k)}{\Gamma, \bigwedge_k A_1 \ldots A_k}$$

**Definition 3.** The *rules of parameter extensionality* are the following rules.

$$\frac{\Gamma, \alpha_k(\wp_1, \ldots, \wp_k) \qquad \ldots \Gamma, \wp_i \leftrightarrow \wp_i' \ldots \qquad (1 \leq i \leq k)}{\Gamma, \alpha_k(\wp_1', \ldots, \wp_k')}$$

$$\frac{\Gamma, \bar{\alpha}_k(\wp_1, \ldots, \wp_k) \qquad \ldots \Gamma, \wp_i \leftrightarrow \wp_i' \ldots \qquad (1 \leq i \leq k)}{\Gamma, \bar{\alpha}_k(\wp_1', \ldots, \wp_k')}$$

**Definition 4.** The rules of quantification are the following rules.

$$\frac{\Gamma, A(\vec{a})}{\Gamma, \forall_k \vec{p} A(\vec{p})} \qquad \frac{\Gamma, A(\vec{\wp})}{\Gamma, \exists_k \vec{p} A(\vec{p})}$$

Here $\vec{a}$ have to be pairwise distinct eigenvariables. The $\vec{\wp}$ may be arbitrary propositional atoms.

**Definition 5.** The *cut rule* is the following rule.

$$\frac{\Gamma, A \qquad \Gamma, \neg A}{\Gamma}$$

The formula $A$ in the cut rule is called the "cut formula".

One of the problems that can be solved in $\mathsf{AC}^0$ is the following:

> Given truth values $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$, output $q_i$ if $i$ is the smallest index such that $p_i$ is true.

A similar task in standard calculi of propositional logic would require a sequence of cuts, thus artificially increasing the height. As our investigations are essentially based on differences like constant versus logarithmic height, we cannot afford this increase. We therefore introduce a new rule allowing multiple cuts at once.

The presence of this rule will be essential in Corollary 44 where it is used to obtain a proof of *constant* height.

**Definition 6.** The multi-cut rule is the rule

$$\frac{\ldots \qquad \Gamma, \Delta_i \qquad \ldots}{\Gamma}$$

where the $\Delta_i$ are sets of purely propositional formulae such that from the collection of the $\Delta_i$ the empty sequent can be derived by cuts only. The weight of the multi-cut rule is $\sum_i |\Delta_i|$, where $|\Delta_i|$ is the cardinality of the set $\Delta_i$.

In other words, if from an arbitrary number of sequents, a sequent $\Gamma$ can be derived by cuts on only purely propositional formulae, then this derivation of $\Gamma$ counts as a single application of the multi-cut rule. For the calculus obtained to be a proof system in the sense of Cook and Reckhow [7] we require that the sequence of cuts be annotated in notations for proofs. However, as we are only interested in the number of rules applied we will never deal with notations for proofs.

*Remark 7.* Using the multi-cut rule it is possible to prove purely propositional induction in constant depth. In fact, from proofs of $\Gamma, \neg A_i, A_{i+1}$ for all $i$, we can conclude by a single inference $\Gamma, \neg A_0, A_k$.

Next we will define the comprehension rule. It is motivated by the extension rule of extended Frege calculus. There, a new propositional variable may be introduced by the axiom $p \leftrightarrow \varphi$, if $p$ is new, that is, does not occur anywhere earlier in the derivation. The extension rule says that if $\Gamma$ can be derived from the assumption $\exists p(p \leftrightarrow \varphi)$, then it can also be derived without. Note that $\neg(\exists p(p \leftrightarrow \varphi)) \equiv \forall p \neg(p \leftrightarrow \varphi)$. As usual, the universal quantifier is expressed by the eigenvariable condition. As discussed in the introduction, we allow the introduction of several extension variables at the same time.

**Definition 8.** The *$\mathcal{F}$-comprehension rule of width $k$* is the rule

$$\frac{\Gamma, \neg(p_1 \leftrightarrow \varphi_1), \ldots, \neg(p_k \leftrightarrow \varphi_k)}{\Gamma}$$

where $\varphi_1, \ldots, \varphi_k \in \mathcal{F}$ and $p_1, \ldots, p_k$ are pairwise distinct eigenvariables, that is, variables that do not occur (free) in $\Gamma$ or any of the $\varphi_i$'s.

The variables $p_i$ are also called "extension variables" and the $\varphi_i$ "extension formulae".

The name "$\mathcal{F}$-Comprehension Rule" is justified by the fact, that it allows simple proofs of (propositional translations of) the comprehension axiom for formulae in $\mathcal{F}$. Consider the following derivation (where we omit some side formulae; note that weakening is admissible).

$$\dfrac{\dfrac{\dfrac{\ldots \qquad \dfrac{}{(p_i \leftrightarrow \varphi_i), \neg(p_i \leftrightarrow \varphi_i)} \qquad \ldots}{\bigwedge_k (p_i \leftrightarrow \varphi_i), \neg(p_1 \leftrightarrow \varphi_1), \ldots, \neg(p_k \leftrightarrow \varphi_k)}\,\bigwedge_k}{\exists_k \vec{p} \bigwedge_k (p_i \leftrightarrow \varphi_i), \neg(p_1 \leftrightarrow \varphi_1), \ldots, \neg(p_k \leftrightarrow \varphi_k)}\,\exists_k}{\exists_k \vec{p} \bigwedge_k (p_i \leftrightarrow \varphi_i)}\,\mathcal{F}\text{-comprehension}$$

It should be noted that the height of this derivation only depends on the $\varphi_i$ and is independent of $k$. Proposition 13 will provide the needed proofs of the first sequents and will actually show that the heights depend only on the depths of $\varphi_i$'s.

Note that in all the rules we may always assume without loss of generality that the conclusion is already contained in the premise (i.e., is an element of the

context $\Gamma$ already). For example, a typical instance of the or-rule would in fact be

$$\frac{\Gamma, A_0 \vee A_1, A_i}{\Gamma, A_0 \vee A_1}$$

**Definition 9.** The $\mathsf{AC}^0$-Tait calculus is given by the rules considered so far, that is, it is given by the propositional rules, the parameter extensionality rule, the rules of quantification, the cut rule with cut-formulae restricted to purely propositional formulae, the multi-cut rule, and the comprehension rule for purely propositional formulae.

We assume all our proofs to be tree-like. This is not a restriction, as we only look at the height (not the size) of proofs.

Immediately by inspection of the rules, we note that weakening is admissible. This will be used tacitly in the sequel.

**Definition 10.** An $\mathsf{AC}^0$-Tait proof is called $w, c$-*slim*, if all formulae occurring in the proof have size at most $w$, each multi-cut rule has weight at most $c$, and each comprehension rule has at most $c$ extension variables.

We write $\vdash^h_{w,c} \Gamma$ to denote that $\Gamma$ has an $\mathsf{AC}^0$-Tait proof of height $h$ that is $w, c$-slim.

The calculus $\mathsf{AC}^0$-Tait is our analogue to what in usual proof theoretic investigations corresponds to cut-free proofs. So we also consider a variant with proper cuts. In Section 5 we will show how they can be eliminated.

**Definition 11.** If $\mathcal{C}$ is a set of formulae that contains all the purely propositional formulae and is closed under substitution of propositional atoms for propositional atoms we define the calculus "$\mathsf{AC}^0$-Tait with $\mathcal{C}$-cuts" to be $\mathsf{AC}^0$-Tait, but with the cut rule liberalised to formulae in $\mathcal{C}$.

We write $d \vdash^h_{\mathcal{C};w,c} \Gamma$ to denote that $d$ is an $\mathsf{AC}^0$-Tait with $\mathcal{C}$-cuts proof of $\Gamma$ of height $h$ that is $w, c$-slim.

**Definition 12.** The size $\mathsf{sz}(A)$ and depth $\mathsf{dp}(A)$ of a formula $A$ are defined to be the number of occurrences of atoms and connectives in $A$, and the length of a longest path in the syntax tree of $A$, respectively. In particular, $\mathsf{dp}(\mathrm{T}) = \mathsf{dp}(p) = \mathsf{dp}(\alpha_k \vec{\wp}) = 1$, $\mathsf{dp}(\bigvee_k \vec{A}) = 1 + \max\{\mathsf{dp}(A_i) \mid 1 \leq i \leq k\}$, $\mathsf{sz}(\mathrm{T}) = \mathsf{sz}(p) = \mathsf{sz}(\alpha_k \vec{\wp}) = 1$, $\mathsf{sz}(\bigvee_k \vec{A}) = 1 + \sum_{1 \leq i \leq k} \mathsf{sz}(A_i)$.

By a simple induction on $A$ one shows

**Proposition 13.** $\vdash^{\mathcal{O}(\mathsf{dp}(A))}_{\mathsf{sz}(A),0} A, \neg A$

A reader familiar with theories [4] like $V^0$ will note that proofs in $V^0$ translate into families of $\mathsf{AC}^0$-Tait proofs of *constant* height. In fact, $\Delta^B_0$-comprehension in $V^0$ can be translated using the comprehension rule in $\mathsf{AC}^0$-Tait as discussed after Definition 8. The induction implicit in the $|\cdot|$-function can be handled by the multi-cut rule, compare Remark 7. "Wide" conjunction and disjunction and quantifying blocks of propositional variables with their corresponding rules are in one-to-one correspondence with first and second order quantifiers in two-sorted Bounded Arithmetic.

## 3 On Evaluating Circuits

We consider the problem of proving that a circuit, possibly with oracle gates, can be evaluated.

**Definition 14.** Let $C$ be a circuit with nodes $n_1, \ldots, n_k$. Then we define the *evaluation formula associated with* $C$ as the formula $\Psi_C(\vec{p})$ where $p_1, \ldots, p_k$ are propositional variables associated with nodes $n_1, \ldots, n_k$, respectively. $\Psi_C$ is the conjunction of the conditions for each node. If the node $i$ is an $\wedge$-gate, then the associated condition is

$$p_i \leftrightarrow \bigwedge_{\ell} p_{i_1} \ldots p_{i_\ell}$$

where $n_{i_1}, \ldots, n_{i_\ell}$ are the inputs for node $i$; the condition for an $\vee$-gate is similar. In the special cases of an $\wedge$ or $\vee$-gate without inputs, we use the constants T and F, respectively.

For an oracle gate, the condition is

$$p_i \leftrightarrow \alpha_\ell(p_{i_1}, \ldots, p_{i_\ell})$$

where, again, $n_{i_1}, \ldots, n_{i_\ell}$ are, in that order, the inputs to node $i$. Similarly for a negated oracle gate.

It should be noted that $\Psi_C$ is a formula of *constant* depth, irrespectively of the shape of the circuit. However, as we shall see, the height of the proof needed to prove that this circuit can be evaluated depends on the actual structure of the circuit.

**Lemma 15.** *If $C$ is a circuit of height $h$, then there is a proof of height $h+\mathcal{O}(1)$ for $\exists_k \vec{p} \Psi_C(\vec{p})$.*

*Proof.* For $0 \leq \ell < h$ let $p_{i_1}^{(\ell)}, \ldots, p_{i_{k_\ell}}^{(\ell)}$ be the variables associated with the nodes of level $\ell$. So a variable $p_i^{(\ell)}$ depends only on variables $p_j^{(\ell')}$ for some $\ell' < \ell$. We write $C_i^{(\ell)}$ for the condition associated with $p_i^{(\ell)}$. Then the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dots \quad \dfrac{}{p_i^{(\ell)} \leftrightarrow C_i^{(\ell)}, \neg(p_{i_j}^{(\ell)} \leftrightarrow C_{i_j}^{(\ell)})} \quad \dots \quad \begin{matrix}(1 \leq \ell < h)\\(0 \leq j < k_\ell)\end{matrix}}{\Psi_C, \neg(\vec{p}^{\,(h)} \leftrightarrow \vec{C}^{\,(h)}), \ldots, \neg(\vec{p}^{\,(2)} \leftrightarrow \vec{C}^{(2)}), \neg(\vec{p}^{\,(1)} \leftrightarrow \vec{C}^{(1)})}\,\wedge}{\exists_k \vec{p} \Psi_C, \neg(\vec{p}^{\,(h)} \leftrightarrow \vec{C}^{(h)}), \ldots, \neg(\vec{p}^{\,(2)} \leftrightarrow \vec{C}^{(2)}), \neg(\vec{p}^{\,(1)} \leftrightarrow \vec{C}^{\,(1)})}\,(\exists_k)}{\exists_k \vec{p} \Psi_C, \neg(\vec{p}^{\,(h)} \leftrightarrow \vec{C}^{(h)}), \ldots, \neg(\vec{p}^{\,(2)} \leftrightarrow \vec{C}^{(2)})}\,(\text{comp})}{\dfrac{\dots}{\exists_k \vec{p} \Psi_C, \neg(\vec{p}^{\,(h)} \leftrightarrow \vec{C}^{(h)})}\,(\text{comp})}\,(\text{comp})}{\exists_k \vec{p} \Psi_C}\,(\text{comp})$$

is as desired.

## 4 Sequential Iteration in Quantified Propositional Logic

For $n$ a natural number we write $[n]$ for the set $\{0, 1, \ldots, n-1\}$. That is, in set theoretic terms, $[n] = n$. If $A$ and $B$ are sets we denote by $f\colon A \rightharpoonup B$ that $f$ is a *partial function from $A$ to $B$*. In other words, $f$ is a function, its domain $\mathsf{dom}(f)$ is a subset of $A$ and its range $\mathsf{rng}(f)$ is a subset of $B$.

By abuse of notation we identify a list $\langle \wp_0, \ldots, \wp_{n-1} \rangle \in \{\mathrm{T}, \mathrm{F}\}^n$ of $n$ boolean values with an element of $[2^n]$ in the following way, assuming the $n$ is understood from the context. $\langle \wp_0, \ldots, \wp_{n-1} \rangle = \sum_{i=0}^{n-1} \chi_{\wp_i} \cdot 2^i$, where we set $\chi_{\mathrm{T}} = 1$ and $\chi_{\mathrm{F}} = 0$.

For the rest of this section we assume that $n$ is big enough, so that $n + \log(n)$ and $2n$ are different. Note that this is the case if $n \geq 1$. The intended meaning of $\alpha_{n+\log n}$ and $\alpha_{2n}$ is that they fix the values of a function $f\colon [2^n] \to [2^n]$ in the following way: $\alpha_{n+\log n}(i, x)$ is true iff the $i$-th bit of $f(x)$ is 1, and $\alpha_{2n}(i, x)$ is true iff $f^i(0) = x$, where $f^i(0)$ is the result of computing the $i$th iterative of $f$ on 0. Storing $f$ by its bitgraph $\alpha_{n+\log n}$ automatically guarantees that a total function of $[n]$ is described, a property which would otherwise require adding more complex quantification to our principle.

**Definition 16.** We write "$f(p_1, \ldots, p_n) = q_1, \ldots, q_n$" for $\bigwedge_{i<n}(q_i \leftrightarrow \alpha_{\tilde{n}}(i, \vec{p}))$ where $\tilde{n} = n + \log(n)$. We write "$\vec{p} = \vec{q}$" for $\bigwedge_{i<n}(p_i \leftrightarrow q_i)$

**Definition 17.** We write "$f^{p_1,\ldots,p_n}(0) = q_1, \ldots, q_n$" for $\alpha_{2n}(\vec{p}, \vec{q})$.

It should be noted that "$f(0) = \vec{q}$" and "$f^1(0) = \vec{q}$" are not only different formulae, but are not even logically equivalent.

**Definition 18.** We write "$p_0, \ldots, p_{n-1} = q_0, \ldots, q_{n-1} + 1$" for the obvious $\mathsf{AC}^0$-formulation of the successor relation, that is, for

$$\bigvee_i \Big( \bigwedge_{j<i} p_j \wedge \neg p_i \wedge \bigwedge_{j<i} \neg q_j \wedge q_i \wedge \bigwedge_{j>i} (p_j \leftrightarrow q_j) \Big) \,.$$

Fix $\ell \leq n$. Our iteration principle will express that $\alpha_{2n}$ stores the graph of $i \mapsto f^i(0)$ for $i = 0, \ldots, \ell$. Using the common idea that $\exists x. f^i(0) = x$ expresses that $f^i(0)$ can be computed, we can argue as follows. If $f^0(0)$ can be computed but $f^\ell(0)$ cannot, then there must be some $i$ such that $f^i(0)$ can be computed but $f^{i+1}(0)$ cannot. The crux is now that this can be expressed using existential quantifiers only, which makes use of the trick that we are storing $f$ by it's bitgraph. If $f^0(0) = 0$ and no $m$ exists with $f^\ell(0) = m$, then there are $m, m', i, i'$ with $i' = i + 1$ and $f^i(0) = m$ and $f(m) = m'$ and not $f^{i'}(0) = m'$. Prenexing this description and identifying the two independent occurrences of $m$ gives us the following iteration formula and principle.

**Definition 19.** *The $n, \ell$-iteration formula* $\Phi_{n,\ell}$ is the following purely propositional formula

$$\begin{aligned}
&\Phi_{n,\ell}(\vec{p}, \vec{p}\,', \vec{q}, \vec{q}\,') \;\equiv \\
&\quad \text{“}f^\ell(0) = \vec{p}\text{”} \vee \neg \text{“}f^0(0) = 0\text{”} \\
&\vee \big(\text{“}\vec{q}\,' = \vec{q} + 1\text{”} \;\wedge\; \text{“}f^{\vec{q}}(0) = \vec{p}\text{”} \;\wedge\; \text{“}f(\vec{p}) = \vec{p}\,'\text{”} \;\wedge\; \neg \text{“}f^{\vec{q}\,'}(0) = \vec{p}\,'\text{”}\big)
\end{aligned}$$

The $n, \ell$-*iteration principle* is the formula

$$\exists_{4n}\vec{p}\vec{p}'\vec{q}\vec{q}'.\, \Phi_{n,\ell}(\vec{p},\vec{p}',\vec{q},\vec{q}')\ .$$

**Definition 20.** A *partial propositional assignment* is a finite partial mapping from the propositional variables to $\{\mathrm{T},\mathrm{F}\}$.

A *partial parameter assignment* is any partial mapping (not necessarily finite) from atomic parameters $\alpha_k(\vec{\wp})$, with $\wp_i \in \{\mathrm{T},\mathrm{F}\}$, to $\{\mathrm{T},\mathrm{F}\}$.

In the context of propositional logic, we use "valuation" as another word for partial (propositional or parameter) assignment. We use $\eta$ to range over valuations. In accordance with set theoretic notions we write the empty valuation as $\emptyset$.

**Definition 21.** A quantified propositional formula is $\alpha$*-free*, if it does not contain any propositional parameter $\alpha_n$, for any $n$. It is called *closed*, if it does not contain any free propositional variables.

Note that any closed, $\alpha$-free quantified propositional formula has a standard truth value T of F in the obvious way.

**Definition 22.** If $A$ is a quantified propositional formula and $\eta$ a partial propositional assignment, we define $A\eta$ by induction on $A$. For $p$ a propositional variable with $p \in \mathsf{dom}(\eta)$ we set $p\eta \equiv \eta(p)$ and $\bar{p}\eta \equiv \neg\eta(p)$. For $p \notin \mathsf{dom}(\eta)$ we set $p\eta \equiv p$ and $\bar{p}\eta \equiv \bar{p}$. The remaining cases are defined homomorphically, e.g., $(\bigwedge_k \vec{A})\eta \equiv \bigwedge_k \vec{A\eta}$. In particular $\alpha_k(\wp_1,\ldots,\wp_k)\eta \equiv \alpha_k(\wp_1\eta,\ldots,\wp_k\eta)$.

If $A$ is a closed purely propositional formula and $\eta$ a partial parameter assignment, we define $A\eta$ by induction on $A$. For $\alpha_k(\vec{\wp})$ with $\alpha_k(\vec{\wp}) \in \mathsf{dom}(\eta)$ we set $(\alpha_k\vec{\wp})\eta \equiv \eta(\alpha_k(\vec{\wp}))$ and $(\bar{\alpha}_k\vec{\wp})\eta \equiv \neg\eta(\alpha_k(\vec{\wp}))$. Otherwise we set $(\alpha_k\vec{\wp})\eta \equiv \alpha_k(\vec{\wp})$ and $(\bar{\alpha}_k\vec{\wp})\eta \equiv \bar{\alpha}_k(\vec{\wp})$. The remaining cases are defined homomorphically.

If $\Gamma = \{A_1,\ldots,A_k\}$ is a set of formulae we write $\Gamma\eta$ for $\{A_1\eta,\ldots,A_k\eta\}$.

**Lemma 23.** *If $\eta \subset \eta'$ are partial propositional assignments and $A$ is a quantified propositional formula such that $A\eta$ is closed, then $A\eta \equiv A\eta'$.*

*If $\eta \subset \eta'$ are partial parameter assignments and $A$ is a closed purely propositional formula such that $A\eta$ is $\alpha$-free, then $A\eta \equiv A\eta'$.*

Definitions 24 and 26 encode the crucial idea of our proof of the boundedness theorem (Theorem 32). Eventually we will be working upwards through a single path of a given proof, and partially define a function $f\colon [2^n] \rightharpoonup [2^n]$ in order to falsify all quantifier free formulae on this path. We want to do this in such a way, that, at level $h$, only $0, f(0),\ldots,f^{h-1}(0)$ are defined. But, to assign a truth value to a quantifier free formula, we not only have to set the parameter bits that encode the relation "$f(x) = y$", but also those that encode the iterations of $f$ of the form "$f^k(0) = y$".

The idea is to assign them values consistent with what we have so far and also consistent with our strategy on how we plan to extend $f$. As we want to keep $f^h(0)$ undefined, all the values in $\mathsf{dom}(f)$ are "forbidden" anyway for the next extension of $f$. Note that, if $f^i(0)$ is defined and $f^i(0) = f^j(0)$ for some $i < j$, then all the values $f^k(0)$ are already defined.

**Definition 24.** A partial function $f\colon [2^n] \rightharpoonup [2^n]$ is called *$\ell$-sequential* if for some $k \leq \ell$ it is the case that $0, f(0), f^2(0), \ldots, f^k(0)$ are all defined, but $f^k(0) \notin \mathsf{dom}(f)$.

*Example 25.* The empty function is $\ell$-sequential for any $\ell \in \mathbb{N}$. If $f$ is a partial function with $f(0) = 0$ then $f$ is *not* $\ell$-sequential for any $\ell$.

**Definition 26.** If $n \in \mathbb{N}$ is a natural number and $f\colon [2^n] \rightharpoonup [2^n]$ a partial function, we associate to $f$, or actually to the pair $n, f$, a partial parameter assignment $\eta_f$ as follows.

For $j \in [n]$, $x \in [2^n]$ with $f(x)$ defined, say $f(x) = \langle \vec{r} \rangle \in [2^n]$, we set $\eta_f(\alpha_{n+\log(n)}(j, x)) = r_j$. Otherwise $\eta_f(\alpha_{n+\log(n)}(j, x))$ is undefined.

For $x, \ell \in [2^n]$ we set $\alpha_{2n}(\ell, x) = \mathrm{T}$ if $f^\ell(0)$ is defined and equal to $x$; otherwise we set $\alpha_{2n}(\ell, x) = \mathrm{F}$ if $x \in \mathsf{dom}(f)$; otherwise $\alpha_{2n}(\ell, x)$ is undefined.

For $k \notin \{2n, n + \log(n)\}$ we set $\eta_f(\alpha_k(\vec{\wp}))$ arbitrarily, say F. Also, if $\vec{p} \in \{\mathrm{T}, \mathrm{F}\}^{\log n} \setminus [n]$, we set $\alpha_{n+\log n}(\vec{p}, \vec{q})$ arbitrarily, say F.

"Good extensions" of partial functions are those that comply with the above idea, that is, those that do not assign new values that are already in the domain.

**Definition 27.** If $f, f'\colon [2^n] \rightharpoonup [2^n]$ are partial functions, and $f \subset f'$ then $f'$ is called a *good extension* of $f$, if $\forall x \in \mathsf{dom}(f')(x \in \mathsf{dom}(f) \vee f'(x) \notin \mathsf{dom}(f))$.

*Remark 28.* If $f \subset f'$ and $f' \subset f''$ are good extensions, then so is $f \subset f''$.

**Proposition 29.** *If $f \subset f'$ is a good extension, then $\eta_f \subset \eta_{f'}$.*

**Lemma 30.** *Let $n \in \mathbb{N}$ and $f\colon [2^n] \rightharpoonup [2^n]$ be an $\ell$-sequential partial function. Moreover, let $M \subset [2^n]$ such that $|\mathsf{dom}(f) \cup M| < 2^n$. Then there is an $(\ell + 1)$-sequential good extension $f'$ of $f$ with $\mathsf{dom}(f') = \mathsf{dom}(f) \cup M$.*

*Proof.* Let $a \in [2^n] \setminus (M \cup \mathsf{dom}(f))$. Such an $a$ exists by our assumption on the cardinality of $M \cup \mathsf{dom}(f)$. Let $f'$ be $f$ extended by setting $f'(x) = a$ for all $x \in M \setminus \mathsf{dom}(f)$. This $f'$ is as desired.

Indeed, assume that $0, f'(0), \ldots, f'^{\ell+1}(0), f'^{\ell+2}(0)$ are all defined. Then, since $a \notin \mathsf{dom}(f')$, all the $0, f'(0), \ldots, f'^{\ell+1}(0)$ have to be different from $a$. Hence these values have already been defined in $f$. But this contradicts the assumption that $f$ was $\ell$-sequential. □

**Lemma 31.** *For every closed, purely propositional, formula $A$ of size $\ell$ there is a set $M \subset [2^n]$ such that $|M| \leq \ell$ and for every function $f$ with $M \subset \mathsf{dom}(f)$ it holds that $A\eta_f$ is $\alpha$-free.*

*Proof.* Let $M$ be the set of all $x \in [2^n]$ such that an atom of the form $\alpha_{n+\log(n)}(j, x)$ or $\alpha_{2n}(k, x)$ occurs in $A$.

Note that $x \in \mathsf{dom}(f)$ forces $\eta_f(\alpha_{2n}(k, x))$ to have a definite value (F unless $f^k(0) = x$, in which case it would be T). □

**Theorem 32.** *Let $k, n, w, c$ be natural numbers with $c \cdot w \geq 2$. Assume $\vdash^h_{w,c} \Gamma$ with $\Gamma = \Delta, \exists_{4n} \vec{r} \Phi_{n,\ell}(\vec{r})$, where $\Phi_{n,\ell}$ is the $n, \ell$-iteration formula. Let $\eta$ be a partial propositional assignment and $f \colon [2^n] \rightharpoonup [2^n]$ be $k$-sequential. Assume $|\mathsf{dom}(f)| + cwh < 2^n$. If $\Delta\eta\eta_f$ is purely propositional, closed, $\alpha$-free, and false then $\ell \leq k + h$.*

The special case $\Delta = \emptyset$, $\eta = \emptyset$, $f = \emptyset$ and $k = 0$ yields

**Corollary 33.** *If $\vdash^h_{w,c} \exists_{4n} \vec{r} \Phi_{n,\ell}(\vec{r})$ and $cwh < 2^n$ for some $c, w$ with $cw \geq 2$ then $h \geq \ell$.*

*Proof (of the theorem).* We argue by induction on $h$ with case distinction according to the last rule of the proof.

The last rule cannot be a propositional axiom, as axioms cannot have $\exists_{4n} \vec{r} \Phi_{n,\ell}(\vec{r})$ as a main formula; however, all the formulae in $\Delta\eta\eta_f$ are false so $\Delta$ cannot be a tautology, as it would have to be, as the calculus is sound. In the case of an $\bigvee_k$-inference apply the induction hypothesis, in the case of an $\bigwedge_k$-inference, the induction hypothesis is applicable to at least one of the subderivations. The last rule cannot be an $\forall_j$-rule as this would require a quantified formula in $\Delta$.

If the last rule is a multi-cut rule

$$\frac{\ldots \Gamma, \Delta_i \ldots}{\Gamma}$$

we know, since the proof is $w, c$-slim, that $\bigcup_i \Delta_i$ contains at most $c$ formulae of size at most $w$. Let $\eta' \supset \eta$ such that all $\Delta_i \eta'$ are closed. Let $M$ be the union of the sets asserted by Lemma 31 for the formulae in $\bigcup_i \Delta_i \eta'$. Then $|M| \leq c \cdot w$. We extend $f$ in a good way to some $(k+1)$-sequential $f'$ with $\mathsf{dom}(f') = \mathsf{dom}(f) \cup M$. Noting that all the $\Delta_i \eta' \eta_{f'}$ are sets of $\alpha$-free, closed, purely propositional formulae we can assign them truth values. Since, by cuts we can derive the empty sequent from the sets $\Delta_i$, and hence also from the sets $\Delta_i \eta' \eta_{f'}$, one of them has to contain only false formulae. Apply the induction hypothesis to this subderivation.

The case of a cut rule is similar, but easier.

Assume that the last rule was a parameter extensionality rule as follows.

$$\frac{\Gamma, \alpha_k(\wp_1, \ldots, \wp_j) \qquad \ldots \Gamma, \wp_i \leftrightarrow \wp'_i \ldots \qquad (1 \leq i \leq j)}{\Gamma, (\alpha_k(\wp'_1, \ldots, \wp'_k))}$$

Extend $\eta$ to some $\eta'$ assigning values to all the $\vec{\wp}$. If for some $1 \leq i \leq j$ we have $\wp\eta' \neq \wp'\eta'$ we can apply the induction hypothesis to the corresponding subderivation. Otherwise $(\alpha_k(\vec{\wp}))\eta'\eta_f \equiv (\alpha_k(\vec{\wp}'))\eta'\eta_f$ and we can apply the induction hypothesis to the first subderivation.

Assume that the last inference rule was an $\exists_j$-rule.

$$\frac{\Gamma, \Phi_{n,\ell}(\vec{\wp}, \vec{\wp}', \vec{\wp}'', \vec{\wp}''')}{\Gamma} \exists_{4n}$$

We can extend $\eta$ to $\eta'$ such that there are natural numbers $m, m', i, i'$ such that $\vec{\varphi}\eta' = m$, $\vec{\varphi}'\eta' = m'$, $\vec{\varphi}''\eta' = i$ and $\vec{\varphi}'''\eta' = i'$. If $\ell \leq k$ there is nothing to show. Otherwise, we will argue as follows that $\Phi_{n,\ell}(m, m', i, i')\eta_{f'}$ can be falsified by choosing an appropriate $(k{+}1)$-sequential good extension $f'$ of $f$. Since $\ell > k$, for every good $(k{+}1)$-sequential extension $f'$ of $f$ we have $f'^{(\ell+1)}(0)$ undefined. Hence for any such $f'$ with $m \in \mathsf{dom}(f')$ we know that $f'^{(\ell)}$ is either undefined or different from $m$ (for otherwise $f'^{(\ell+1)}(0)$ would be defined). In either case $\eta_{f'}(\alpha_{2n}(\ell, m)) = \mathrm{F}$. Recall that adding a value $m$ to the domain of $f'$ ensures that $\eta_{f'}(\alpha_{2n}(\ell, m))$ has a definite value. The second disjunct $\neg$"$f^0(0) = 0$" is falsified by $\eta_{f'}$ for any $f'$. For the last disjunct "$i' = i + 1$" $\wedge$ "$f^i(0) = m$" $\wedge$ "$f(m) = m'$" $\wedge \neg$"$f^{i'}(0) = m'$", we may assume that $i' = i + 1$, for otherwise it is falsified anyway. For any $f'$ with $m, m' \in \mathsf{dom}(f')$ we know that $\eta_{f'}$ assigns definite truth values to "$f^i(0) = m$", "$f(m) = m'$", and "$f^{i'}(0) = m'$". If the first two conjuncts are assigned T, than this can only be if $f'^i(0) = m$ and $f'(m) = m'$. But in this case $f'^{i+1}(0) = m'$, so $\neg$"$f^{i+1}(0) = m'$" is assigned F. Altogether we can take any $(k{+}1)$-sequential good extension $f'$ of $f$ with $\mathsf{dom}(f') = \mathsf{dom}(f) \cup \{m, m'\}$. Then $\Phi_{n,\ell}(\vec{\varphi}, \ldots)\eta'\eta_{f'}$ is $\alpha$-free, closed, purely propositional and false and we can apply the induction hypothesis (recalling that we assumed $wc \geq 2$).

The last remaining case is that the last rule was a comprehension rule

$$\frac{\Gamma, \neg(p_1 \leftrightarrow \varphi_1), \ldots, \neg(p_j \leftrightarrow \varphi_j)}{\Gamma}$$

where the $\varphi_i$ are purely propositional, the $\vec{p}$ are eigenvariables, and, since the proof is $w, c$-slim, $j \leq c$. Let $\eta'' \supset \eta$ be such that all $\varphi_i\eta''$ are closed. Let $M_i$ be the set asserted by Lemma 31 for $\varphi_i\eta''$. Extend in a good way $f$ to a $(k{+}1)$-sequential $f'$ with $\mathsf{dom}(f') = \mathsf{dom}(f) \cup \bigcup_i M_i$. Due to the eigenvariable condition we can assume without loss of generality that $\vec{p} \notin \mathsf{dom}(\eta'')$. Extend $\eta''$ to $\eta'$ by setting $p_i$ to the truth value of $\varphi_i\eta''\eta_{f'}$. We then can apply the induction hypothesis.

This finishes the proof. □

As a proof complexity consequence of the above theorem we can make the following observation.

**Corollary 34.** *There is a family of polynomial size $\Sigma^q_1(\alpha)$-formulae, i.e., formulae of the shape of existentially quantified purely propositional formulae, such that every $\mathsf{AC}^0$-Tait proof with polynomially branching rules and polynomial size formulae requires exponential height.*

*Any proof of this family requires exponential size.*

*Proof.* As Corollary 33 shows, the family $(\exists_{4n}\vec{r}\Phi_{n,2^n-1}(\vec{r}))_{n\in\mathbb{N}}$ is as desired. It should be noted that these formuale indeed only grow polynomially, as, of course, the number $2^n - 1$ can be represented by $n$ bits. □

## 5 Cut-Elimination

We now show how cuts on more complicated formulae can be reduced to quantifier-free cuts. We will obtain the typical increase in height occurring in proof-theoretic cut-elimination.

**Definition 35.** A substitution $\sigma$ is called an *atomic substitution*, if for every propositional variable $p$ we have $\sigma(p) \in \mathcal{A}$. In other words, a substitution is atomic, if the range only contains propositional atoms.

**Lemma 36.** *If* $\vdash^h_{\mathcal{C};w,c} \Gamma$ *then* $\vdash^h_{\mathcal{C};w,c} \Gamma\sigma$ *for every atomic substitution* $\sigma$*.*

**Lemma 37.** *If* $\vdash^h_{\mathcal{C};w,c} \Gamma, \forall_k \vec{p} A(\vec{p})$*, then* $\vdash^h_{\mathcal{C};w,c} \Gamma, A(\vec{\wp})$ *for arbitrary propositional atoms* $\vec{\wp}$*.*

*Proof.* Induction on the derivation. We can identically reproduce any rule that does not have $\forall_k \vec{p} A(\vec{p})$ as main formula.

In case $\Delta, \forall_k \vec{p} A(\vec{p})$ was concluded from $\Delta, \forall_k \vec{p} A(\vec{p}), A(\vec{a})$ with pairwise distinct eigenvariables $\vec{a}$, we may, by Lemma 36, assume without loss of generality that the $\vec{a}$ are disjoint from $\vec{\wp}$. First apply the induction hypothesis to the premise, obtaining $\Delta, A(\vec{\wp}), A(\vec{a})$ and then apply Lemma 36 to obtain $\Delta, A(\vec{\wp})$. Note that the eigenvariable property ensures that $\Delta$ is not affected by this substitution. □

**Lemma 38.** *Assume* $A \in \mathcal{C}$*. If* $\vdash^h_{\mathcal{C};w,c} \Gamma, \forall_k \vec{p} A(\vec{p})$ *and* $\vdash^{h'}_{\mathcal{C};w,c} \Gamma, \exists_k \vec{p} \neg A(\vec{p})$ *then* $\vdash^{h+h'}_{\mathcal{C};w,c} \Gamma$*.*

*Proof.* Let $\vec{q}$ be new and pairwise distinct variables. By Lemma 37 we get $\vdash^h_{\mathcal{C};w,c} \Gamma, A(\vec{q})$. Now argue by Induction on the second derivation, or, equivalently, by induction on $h'$. Every rule of the second derivation can be reproduced identically, except for an $\exists_k$-introduction with conclusion $\exists_k \vec{p} \neg A(\vec{p})$.

So assume that $\vdash^{h''+1}_{\mathcal{C};w,c} \Gamma, \exists_k \vec{p} \neg A(\vec{p})$ was concluded from $\vdash^{h''}_{\mathcal{C};w,c} \Gamma, \exists_k \vec{p} \neg A(\vec{p}), \neg A(\vec{\wp})$ with the $\vec{\wp}$ necessarily propositional atoms, by the restriction of the quantification rules (Definition 4). First apply the induction hypothesis to $\vdash^{h''}_{\mathcal{C};w,c} \Gamma, \exists_k \vec{p} \neg A(\vec{p}), \neg A(\vec{\wp})$ and obtain $\vdash^{h''+h}_{\mathcal{C};w,c} \Gamma, \neg A(\vec{\wp})$. Also, apply Lemma 36 to $\vdash^h_{\mathcal{C};w,c} \Gamma, A(\vec{q})$ and obtain $\vdash^h_{\mathcal{C};w,c} \Gamma, A(\vec{\wp})$ using that the $\vec{q}$ are fresh, i.e., in particular not free in $\Gamma$. Then conclude $\vdash^{h''+h+1}_{\mathcal{C};w,c} \Gamma$ by a cut on $A(\vec{\wp})$ which is allowed as $A \in \mathcal{C}$ and $\mathcal{C}$ is closed under atomic substitutions. □

**Corollary 39.** *If* $\vdash^h_{\exists\mathcal{C};w,c} \Gamma$ *then* $\vdash^{2^h}_{\mathcal{C};w,c} \Gamma$ *where* $\exists\mathcal{C} = \mathcal{C} \cup \{\exists_k \vec{p} A(\vec{p}) \mid A(\vec{p}) \in \mathcal{C}\}$*.*

## 6 Circuit Evaluation and Iteration

We now have all the preparations needed to show the following lower bound. Consider a proof that a circuit can be evaluated. If the circuit has height $h$, then the proof has to have height at least $h - \mathcal{O}(1)$.

Obviously, a circuit of height $h$ can compute the $h$'th iterate of the function given by $\alpha$. From the fact that this circuit can be evaluated, we can conclude that the $h$-iteration principle holds.

In the following let $C_h$ be the circuit canonically iterating the function given by the oracle. We also assume the size parameter $n$ to be understood; we set $\tilde{n} = n + \log n$. Immediately from the definition we get

**Proposition 40.** *$\Psi_{C_h}(\vec{w})$ is the conjunction of the clauses $w_j^{(0)} \leftrightarrow \mathrm{F}$ and the conjuncts "$f(\vec{w}^{(\ell)}) = \vec{w}^{(\ell+1)}$". The latter is built of the formulae $w_j^{(\ell+1)} \leftrightarrow \alpha_{\tilde{n}}(j, \vec{w}^{(\ell)})$ of $0 \leq \ell < h-1$ and $0 \leq j < n$.*

**Proposition 41.** *$\mathsf{sz}(\Psi_{C_h}) \in \mathcal{O}(n \cdot h)$ and $\mathsf{dp}(\Psi_{C_h}) \in \mathcal{O}(1)$.*

For $1 \leq \ell < n$ we set $\Delta_\ell \equiv \neg$"$f^{\ell-1}(0) = \vec{w}^{(\ell-1)}$", "$f^{\ell}(0) = \vec{w}^{(\ell)}$". Recall that "$f^{\vec{p}}(0) = \vec{q}$" is a shorthand for $\alpha_{2n}(\vec{p}, \vec{q})$. So, by resolution the $\Delta_\ell$ imply $\neg$"$f^0(0) = \vec{w}^{(0)}$", "$f^h(0) = \vec{w}^{(h)}$".

**Proposition 42.** *If $i \in [n-1]$ and $\vec{p} = i$ and $\vec{q} = i+1$ then $\vdash^{\mathcal{O}(1)}_{\mathcal{O}(\log n),1}$ "$\vec{q} = \vec{p} + 1$".*

**Lemma 43.** *$\vdash^{\mathcal{O}(1)}_{\mathcal{O}(nh),1} \Delta_\ell, \exists_{4n}\vec{u}\Phi_{n,h}(\vec{u}), \forall_{h\cdot n}\vec{w}\neg\Psi_{C_h}(\vec{w})$ with $\Phi_{n,h}(\vec{u})$ the $n,h$-iteration Formula, as defined in Definition 19.*

*Proof.* First note, that there are constant height proofs of the following sequents.

- "$f(\vec{w}^{(\ell-1)}) = \vec{w}^{(\ell)}$", $\neg\Psi_{C_h}(\vec{w})$
- "$f^{\ell-1}(0) = \vec{w}^{(\ell-1)}$", $\neg$"$f^{\ell-1}(0) = \vec{w}^{(\ell-1)}$"
- "$f^{\ell}(0) = \vec{w}^{(\ell)}$", $\neg$"$f^{\ell}(0) = \vec{w}^{(\ell)}$"
- "$(\ell+1) = \ell + 1$"

Therefore applications of an $\bigwedge_4$-rule followed by an $\vee$-rule and an $\exists_{4n}$-rule gives us $\exists_{4n}\vec{u}\Phi_{n,h}(\vec{u}), \neg\Psi_{C_h}(\vec{w}), \Delta_\ell$ from where we get the desired derivation by an $\forall_{nh}$-rule.

**Corollary 44.** *$\vdash^{\mathcal{O}(1)}_{\mathcal{O}(nh),\mathcal{O}(h)} \exists_{4n}\vec{u}\Phi_{n,h}(\vec{u}), \forall_{h\cdot n}\vec{w}\neg\Psi_{C_h}$*

*Proof.* Apply a mutli-cut rule to the derivations of Lemma 43 to obtain $\neg$"$f^0(0) = \vec{w}^{(0)}$", "$f^h(0) = \vec{w}^{(h)}$", $\exists_{4n}\vec{u}\Phi_{n,h}(\vec{u}), \forall_{h\cdot n}\vec{w}\neg\Psi_{C_h}(\vec{w})$. Two $\vee$-rules and an $\exists_{4n}$-rule finish the proof.

**Theorem 45.** *There are natural numbers $c, C$ such that forall sufficiently large $n, h$ whenever $c^2 \cdot h^2 n < 2^n$ and $\vdash^{h'}_{c\cdot nh,ch} \exists_{nh}\vec{w}\Psi_{C_h}(\vec{w})$ then $h' \geq h - C$.*

*Proof.* Assume $\vdash^{h'}_{\tilde{w},\tilde{c}} \exists_{nh}\vec{w}\Psi_{C_h}$. By Corollary 44 we have (for sufficiently large $n$) a derivation $\vdash^{c_2}_{c_1 nh, c_1 h} \exists_{4n}\vec{u}\Phi_{n,h}(\vec{u}), \forall_{hn}\vec{w}\neg\Psi_{C_h}$. Therefore, by Lemma 38, we get $\vdash^{h'+c_2}_{\max\{w,c_1 nh\},\max\{\tilde{c},c_1 h\}} \exists_{4n}\vec{u}\Phi_{n,h}(\vec{u})$. So, by Corollary 33, we get $h' + c_2 \geq h$, provided $\max\{\tilde{w}, c_1 nh\} \cdot \max\{\tilde{c}, c_1 h\} < 2^n$.

An immediate consequence of Theorem 45 is that a proof of $\Psi_{C_h}$ requires height $h - \mathcal{O}(1)$, for $h$ growing sub-exponentially with $n$.

## References


1. Arnold Beckmann. Dynamic ordinal analysis. *Archive for Mathematical Logic*, 42:303–334, 2003.
2. Samuel R. Buss and Jan Krajíček. An application of boolean complexity to separation problems in bounded arithmetic. *Proceedings of the London Mathematical Society*, 69(3):1–27, 1994.
3. Peter Clote and Gaisi Takeuti. First order bounded arithmetic and small Boolean circuit complexity classes. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, volume 13 of *Progr. Comput. Sci. Appl. Logic*, pages 154–218. Birkhäuser, Boston, MA, 1995.
4. Stephen A. Cook. Theories for complexity classes and their propositional translations. In Jan Krajíček, editor, *Complexity of computations and proofs*, Quaderni die Matematica, pages 175–227. Dipartimento di Matematica, Seconda Universitá degli Studi di Napoli, 2003.
5. Stephen A. Cook and Tsuyoshi Morioka. Quantified propositional calculus and a second-order theory for $NC^1$. *Arch. Math. Logic*, 44(6):711–749, 2005.
6. Stephen A. Cook and Phuong Nguyen. Foundations of proof complexity: Bounded arithmetic and propositional translations. draft of a book, available at `http://www.cs.toronto.edu/~sacook/csc2429h/book/`.
7. Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1), March 1979.
8. Jan Krajíček. Lower bounds to the size of constant-depth propositional proofs. *The Journal of Symbolic Logic*, 59(1):73–86, 1994.
9. Jan Krajíček and Pavel Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 36:29–46, 1990.
10. Steven Perron. A propositional proof system for log space. In C.-H. Luke Ong, editor, *Proceedings of the 19th international Workshop on Computer Science Logic (CSL '05)*, volume 3634 of *Lecture Notes in Computer Science*, pages 509–524. Springer Verlag, August 2005.
11. Wolfram Pohlers. *Proof theory*, volume 1407 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin, 1989. An introduction.
12. Alan Skelley. Propositional PSPACE reasoning with Boolean programs versus quantified Boolean formulas. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, volume 3142 of *Lecture Notes in Computer Science*, pages 1163–1175. Springer Verlag, 2004.
13. 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 Verlag, 1968.
14. Gaisi Takeuti. Separations of theories in weak bounded arithmetic. *Annals of Pure and Applied Logic*, 71:47–67, 1995.
15. Christopher B. Wilson. A measure of relativized space which is faithful with respect to depth. *Journal of Computer and System Sciences*, 36(3):303–312, June 1988.