Weak Bounded Arithmetic, the Diffie-Hellman Problem and Constable's Class K

Jan Johannsen[*]
Dept. of Mathematics
U.C. San Diego
`johannsn@math.ucsd.edu`

Abstract

The bounded arithmetic theory C_2^0 of [9], which is closely related to the complexity class DLogTime-uniform TC^0, is extended by a function symbol and axioms for integer division, which is not known to be in DLogTime-uniform TC^0. About this extended theory $C_2^0[div]$, two main results are proved:

1. The Σ_1^b-definable functions of $C_2^0[div]$ are exactly Constable's class K, a function algebra whose precise complexity-theoretic nature is yet to be determined. This also yields the new upper bound $K \subseteq$ uniform NC^2.

2. The Δ_1^b-theorems of $C_2^0[div]$ do not have Craig-interpolants of polynomial circuit size, unless the Diffie-Hellman key exchange protocol is insecure.

1 Introduction

The class K was introduced by Constable [7] as a natural feasible analog of the elementary recursive functions. It is defined to be the least class that contains the basic arithmetic operations, and is closed under composition and the formation of short iterated sums and products, i.e. sums and products of logarithmically many terms. It was claimed in [7] without proof that K contains exactly the polynomial time computable functions, but this seems to be unlikely now, as we know from Clote [5] that K is contained in uniform NC. It is also shown in [5] that K contains DLogTime-uniform TC^0. An exact complexity-theoretic characterization of K is yet to be given.

In [9], we defined a bounded arithmetic theory C_2^0 whose Σ_1^b-definable functions (the provably total functions whose graphs can be written as bounded existential (Σ_1^b-) formulas) are exactly those in DLogTime-uniform TC^0. In this paper, we consider an extension $C_2^0[div]$ that results from C_2^0 by adding a function symbol for integer division (which is not known to be in DLogTime-uniform TC^0, but only in P-uniform TC^0) and some quantifier-free axioms that specify its interpretation. We show that the Σ_1^b-definable functions in $C_2^0[div]$ are exactly those in the class K. As a corollary, we obtain a new function algebra that is equivalent to K, viz. the function algebra characterization of DLogTime-uniform TC^0 of Clote and Takeuti [6], extended by integer division as a base function. This characterization implies a better upper bound on the complexity of K, viz. $K \subseteq$ uniform NC^2.

The main technical work goes into showing that the Σ_1^b-definable functions of $C_2^0[div]$ are closed under short iterated products. This is done by formalizing the reduction of ITERATED MULTIPLICATION to DIVISION of Beame et al. [1], and involves formalizing some basic number theory in $C_2^0[div]$ for small numbers, e.g. the Chinese Remainder Theorem and the structure theorem for the multiplicative groups $(\mathbb{Z}/q\mathbb{Z})^*$ for prime powers q.

We then study $C_2^0[div]$ with respect to interpolation. We say that a theory T of Bounded Arithmetic has *feasible Δ_1^b-interpolation*, if for every pair of formulas $A(\vec{x},\vec{y})$ and $B(\vec{x},\vec{z})$ that are Δ_1^b w.r.t. T, and for which $T \vdash A(\vec{x},\vec{y}) \rightarrow B(\vec{x},\vec{z})$, there is a Craig-interpolant $C(\vec{x})$ that — as a predicate of natural numbers — is computable by polynomial size circuits, i.e., it is in $P/poly$.

Somewhat surprisingly, the question whether bounded arithmetic theories have feasible Δ_1^b-interpolation is related to cryptographic hardness assumptions. Krajíček and Pudlák [11] show that if the RSA cryptosystem is secure against attacks in $P/poly$, then the well-known bounded arithmetic theory S_2^1 of Buss [3] does not have feasible Δ_1^b-interpolation. Moreover if $S_2^1 + \Phi$ has feasible Δ_1^b-interpolation, then the

[*] Supported by DFG grant No. Jo 291/1-1

discrete logarithm function is computable in $P/poly$ [11], and also by Buss [4], the Rabin cryptosystem can be broken in $P/poly$. Here Φ is the axiom that every natural number satisfying Pratt's NP definition of primes [13] is actually irreducible.

We show that $C^0_2[div]$ does not have feasible Δ^b_1-interpolation, unless the Diffie-Hellman problem, whose hardness is a common assumption in cryptography, can be solved in $P/poly$, and hence, by a well-known reduction [15, 12], factoring integers that are products of two large primes is possible in $P/poly$. Our result improves upon the above in two ways: we replace $S^1_2 + \Phi$ by the much weaker theory $C^0_2[div]$, and we also weaken the cryptyographic hardness assumption.

All the mentioned results, including ours, remain true if $P/poly$ is replaced by any other complexity class in the definition of feasible Δ^b_1-interpolation as well as in the hardness assumption. The reason that $P/poly$ is used, instead of e.g. the more natural P, is the relation of feasible Δ^b_1-interpolation to the notion of *feasible interpolation* for propositional proof systems.

A proof system Σ for propositional logic is said to have *feasible interpolation*, if for every tautology $A(\vec{p},\vec{q}) \to B(\vec{p},\vec{r})$ there is a Craig-interpolant $C(\vec{p})$ whose circuit size is bounded by a polynomial in the size of a shortest proof of $A(\vec{p},\vec{q}) \to B(\vec{p},\vec{r})$ in Σ.

By a translation of bounded arithmetic formulas into propositional logic, the Δ^b_1-consequences of some bounded arithmetic theory T have polynomial size proofs in an associated proof system Σ_T. In this case, feasible interpolation for Σ_T implies feasible Δ^b_1-interpolation for the theory T. In this way, the first mentioned result of [11] implies that Extended Frege systems do not enjoy feasible interpolation, unless RSA is insecure.

This was improved upon by Bonet et al. [2], who showed that the TC^0-Frege system, in which every line is represented by a constant-depth threshold circuit, does not have feasible interpolation if the Diffie-Hellman problem is hard. Our result can be seen as the uniform version of this result, which was posed as a problem in [2], and the idea of our proof is taken from that paper.

The natural candidate for a bounded arithmetic theory corresponding to TC^0-Frege proofs is of course C^0_2. Although the simulation of the Δ^b_1-consequences of C^0_2 by polynomial size TC^0-Frege proofs has never been worked out in detail, it is straightforward though tedious. But the TC^0-circuits occurring in the proofs obtained by the simulation are all DLogTime-uniform, whereas the proofs of the Diffie-Hellman tautologies in [2] make essential use of the P-uniform, but probably not DLogTime-uniform circuits of [1]. Thus it is highly unlikely that C^0_2 itself could prove the arithmetic form of these tautologies, and thus could be shown not to enjoy feasible Δ^b_1-interpolation by this method.

On the other hand, all the functions used in the proofs in [2] are in the class K, so our first Main Theorem shows that the theory $C^0_2[div]$ has just the right amount of (non-)uniformity to prove the inverse image under translations of the Diffie-Hellman tautologies.

The structure of the paper is as follows. In Section 2, we recall the function algebra characterization of DLogTime-uniform TC^0 due to Clote and Takeuti [6] and the definition of Constable's class K. In Section 3, we define the theory $C^0_2[div]$ and prove that its Σ^b_1-definable functions are the class K, by showing that they are closed under short iterated sums and products. The upper bound on the complexity of the functions in K follows as a corollary. In Section 4, we define the Diffie-Hellman formula, and show that it is provable in $C^0_2[div]$. Then we observe that a Craig-interpolant for this formula solves the Diffie-Hellman problem, which allows us to conclude that the assumption that this problem is hard implies that $C^0_2[div]$ does not have feasible Δ^b_1-interpolation.

In this extended abstract we give only sketches of the proofs, the details will be presented in a forthcoming full version of this paper.

2 Function Algebras

We say that a function f is defined by *concatenation recursion on notation* (CRN) from g and h_0, h_1 if

$$\begin{aligned} f(0,\vec{y}) &= g(\vec{y}) \\ f(s_0(x),\vec{y}) &= 2\cdot f(x,\vec{y}) + h_0(x,\vec{y}) \qquad \text{for } x>0 \\ f(s_1(x),\vec{y}) &= 2\cdot f(x,\vec{y}) + h_1(x,\vec{y}) \end{aligned}$$

provided that $h_i(x,\vec{y}) \le 1$ for all $x,\vec{y}$ and $i = 0,1$.

Let $i^n_k(x_1,\dots x_n) := x_k$, $s_0(x) := 2x$, $s_1(x) = 2x+1$, $|x| := \lceil \log_2(x+1)\rceil$, $x\#y := 2^{|x|\cdot|y|}$ and $Bit(x,i) := \lfloor \frac{x}{2^i} \rfloor \bmod 2$. The following characterization of the number-theoretic functions in DLogTime-uniform TC^0 was given by Clote and Takeuti [6]:

Definition 1. *The class T is the smallest class of functions that contains 0, i^n_k, s_0, s_1, multiplication $\cdot$, $\#$, $|x|$, Bit and which is closed under composition and CRN.*

Proposition 1. $T =$ *DLogTime-uniform* TC^0.

Definition 2. *The class $T[div]$ is defined exactly as T, but with integer division $\lfloor \frac{\cdot}{\cdot} \rfloor$ among the base functions.*

A function $f(x,\vec{y})$ is a *weak sum*, if it is defined from $g(x,\vec{y})$ by

$$f(x,\vec{y}) = \sum_{i=0}^{|x|} g(i,\vec{y}) \ ,$$

and it is a *weak product*, if it is defined by

$$f(x,\vec{y}) = \prod_{i=0}^{|x|} g(i,\vec{y}) \ .$$

Definition 3. *Constable's class K is the smallest class of functions that contains 0, i^n_k, s_0, s_1, $+$, $\dot{-}$, $\cdot$ and division $\lfloor\dot{\div}\rfloor$, and is closed under composition and weak sums and products.*

Proposition 2. *$T[div] \subseteq K$.*

This follows by the proof in [5] that $T \subseteq K$, which shows that the base functions of T are in K and that K is closed under CRN. We will see below that actually $K = T[div]$. Since integer division is known to be in uniform NC^2 by Reif [14], this implies $K \subseteq$ uniform NC^2.

3 Bounded Arithmetic

The language of Bounded Arithmetic comprises the usual signature of arithmetic $0, S, +, \dot{-}, \cdot, \leq$, together with function symbols for $\lfloor \frac{1}{2}x \rfloor$, $MSP(x,i) := \lfloor x/2^i \rfloor$, $|x|$ and $\#$.

A quantifier of the form $\forall x \leq t$, $\exists x \leq t$ with x not occurring in t is called a *bounded quantifier.* Furthermore, the quantifier is called *sharply bounded* if the bounding term t is of the form $|s|$ for some term s. A formula is called (sharply) bounded if all quantifiers in it are (sharply) bounded.

We denote the class of quantifier-free formulas by *open*. The class of sharply bounded formulas is denoted Σ^b_0 or Π^b_0. For $i \in \mathbb{N}$, Σ^b_{i+1} (resp. Π^b_{i+1}) is the least class containing Π^b_i (resp. Σ^b_i) and closed under conjunction, disjunction, sharply bounded quantification and bounded existential (resp. universal) quantification.

$BASIC$ denotes a set of quantifier-free axioms specifying the interpretations of the function symbols of the language, see Buss [3] and Takeuti [16].

For more background concerning Bounded Arithmetic see Krajíček [10] or [3]. We define some terms that will be used frequently below:

$$\begin{aligned} 2^{|x|} &:= 1\#x \\ mod2(x) &:= x \dot{-} 2 \cdot \lfloor \frac{1}{2}x \rfloor \\ Bit(x,i) &:= mod2(MSP(x,i)) \\ 2^{\min(x,|y|)} &:= MSP(2^{|y|}, |y| \dot{-} x) \\ LSP(x,i) &:= x \dot{-} 2^{\min(i,|x|)} \cdot MSP(x,i) \\ \beta_a(w,i) &:= MSP(LSP(w, Si \cdot |a|), i \cdot |a|) \end{aligned}$$

so that $LSP(x,i)$ returns the number consisting of the last i bits of x, and $\beta_a(w,x)$ projects the xth block of bits of length $|a|$ out of w, which is used for sequence coding.

The theory C^0_2 is axiomatized by the quantifier-free $BASIC$ axioms, the scheme *open-LIND*

$$A(0) \wedge \forall x\, (A(x) \to A(Sx)) \to \forall x\, A(|x|)$$

for all quantifier-free formulas $A(x)$, and the replacement scheme $BB\Sigma^b_0$

$$\begin{aligned} &\forall x \leq |s|\ \exists y \leq t(x)\ A(x,y) \to \\ &\qquad \exists w < 2(t^*\#2s)\ \forall x \leq |s|\ \beta_{t^*}(w,x) \leq t(x) \\ &\qquad\qquad \wedge A(x, \beta_{t*}(w,x)) \end{aligned}$$

for every Σ^b_0-formula $A(x,y)$. Here $t^* := t^M(|s|)$ for a monotone term t^M that majorizes t.

We say that a function $f(\vec{x})$ is Σ^b_1-definable in a theory T if there is a Σ^b_1-formula $A(\vec{x},y)$ and a term $t(\vec{x})$ such that

$$\begin{aligned} &\mathbb{N} \models \forall \vec{x}\, A(\vec{x}, f(\vec{x})) \\ &T \vdash \forall \vec{x}\, \exists y \leq t(\vec{x})\ A(\vec{x},y) \\ &T \vdash \forall \vec{x}, y, z\ A(\vec{x},y) \wedge A(\vec{x},z) \to y = z \ . \end{aligned}$$

In [9, 8], we showed that the Σ^b_1-definable functions of C^0_2 are exactly those in DLogTime-uniform TC^0.

Definition 4. *$C^0_2[div]$ is the theory defined like C^0_2, but with an added function symbol $\lfloor\dot{\div}\rfloor$ and the following axioms for it:*

$$\lfloor \frac{x}{0} \rfloor = 0$$

$$y > 0 \ \to\ y \cdot \lfloor \frac{x}{y} \rfloor \ \leq\ x \ <\ y \cdot \lfloor \frac{x}{y} \rfloor + y$$

We say that a formula $A(x)$ is Δ^b_1 in a theory T, if it is provably in T equivalent to a Σ^b_1- and a Π^b_1-formula. For a class of formulas Γ, the bit-comprehension axiom scheme Γ-$COMP$ is

$$\exists y < 2^{|a|}\ \forall i < |a|\ \big(Bit(y,i) = 1 \ \leftrightarrow\ A(i)\big)$$

for every formula $A(i) \in \Gamma$.

Proposition 3. *$C^0_2[div]$ proves Δ^b_1-$COMP$ and Δ^b_1-$LIND$.*

The proofs of these schemes in C^0_2 [9] apply to $C^0_2[div]$ as well.

Main Theorem 4. *The Σ^b_1-definable functions in $C^0_2[div]$ are exactly those in K.*

Proof. The base functions of K are all terms in the language of $C^0_2[div]$, and the closure under composition is trivial. In Theorems 7 and 17 below, we will show closure under weak sums and products. Hence every function in K is Σ^b_1-definable in $C^0_2[div]$.

On the other hand, the witnessing argument in [8] is easily modified to show that every Σ^b_1-definable function in $C^0_2[div]$ is in $T[div]$, hence in K by Prop. 2. □

Corollary 5. *$K = T[div]$.*

By Reif [14], integer division is in uniform NC^2, and it is well-known that uniform NC^2 is closed under CRN, hence we obtain:

Corollary 6. *$K \subseteq$ uniform NC^2.*

3.1 Weak Sums

Theorem 7. *The Σ^b_1-definable functions of $C^0_2[div]$ (and those of C^0_2) are closed under weak sums.*

Let $g(x)$ be Σ^b_1-defined with bounding term $t(x)$. W.l.o.g. we assume that t is monotone. Then we have

$$\Big|\sum_{i=0}^{|x|} g(i)\Big| \le |t(|x|)| + |x| \le |(2x+1)t(|x|)| \;.$$

We reduce the weak sum to multiplication by the following trick: For $b \ge |(2x+1)t(|x|)|$, we set

$$A_g(x,b) := \sum_{i=0}^{|x|} g(i)\cdot 2^{ib}$$
$$B(x,b) := \sum_{i=0}^{|x|} 2^{ib}$$

These functions are easily definable by Δ^b_1-$COMP$. Hence we can define $AuxSum_g(x,b)$ as

$$LSP(MSP(A_g(x,b)\cdot B(x,b), |x||b|), |b|)$$

and finally

$$\sum_{i=0}^{|x|} g(i) := AuxSum_g(x,(2x+1)t(|x|)) \;.$$

The following proposition justifies this definition by showing that the inductive definition of iterated sums is provable in $C^0_2[div]$.

Proposition 8. *C^0_2 proves the following equations:*

$$\sum_{i=0}^{|0|} g(i) = g(0) \tag{1}$$
$$\sum_{i=0}^{|x|} g(i) = \Big(\sum_{i=0}^{|\lfloor\frac12 x\rfloor|} g(i)\Big) + g(|x|) \tag{2}$$

For the proof we use the following decomposition, which follows easily from distributivity and the definitions

$$\begin{aligned}&A_g(x,b)\cdot B(x,b)\\ &\quad= A_g(\lfloor\tfrac12 x\rfloor,b)\cdot B(\lfloor\tfrac12 x\rfloor,b) + A_g(\lfloor\tfrac12 x\rfloor,b)\cdot 2^{|x||b|}\\ &\qquad\qquad + B(x,b)\cdot g(|x|)\cdot 2^{|x||b|}\;.\end{aligned}$$

By induction, it is proved that

$$\begin{aligned}&LSP(MSP(A_g(\lfloor\tfrac12 x\rfloor,b)\cdot B(\lfloor\tfrac12 x\rfloor,b),|x|),|b|)\\ &\qquad\qquad = AuxSum_g(\lfloor\tfrac12 x\rfloor,b) - g(0)\end{aligned}$$

and then (2) follows by the decomposition above and the observation that

$$LSP(MSP(A_g(\lfloor\tfrac12 x\rfloor,b)\cdot 2^{|x||b|},|x||b|),|b|) = g(0)$$
$$LSP(MSP(B(x,b)\cdot g(|x|)\cdot 2^{|x||b|},|x||b|),|b|) = g(|x|)\;.$$

3.2 Exponentiation

We now define exponentiation from division using a geometric series expansion, as in [1]. Let $A(b) := 2^{2|b|^3+2|b|^2}$ and $B(b) := 2^{2|b|^2}$, then we define

$$E(x,b) := \left\lfloor \frac{A(b)}{B(b) \dot- x}\right\rfloor$$
$$exp(i,x,b) := LSP\big(MSP\big(E(x,b), 2|b|^2(|b| \dot- i)\big), 2|b|^2\big)$$

Note that $exp(i,x,b)$ is a term in the language of $C^0_2[div]$. The following proposition justifies that we write x^i for $exp(i,x,b)$ if we know $i, |x| \le |b|$.

Proposition 9. *$C^0_2[div]$ proves: if $|x| \le |b|$, then*

$$exp(0,x,b) = 1 \tag{3}$$
$$exp(i+1,x,b) = exp(i,x,b)\cdot x \quad \textit{for } i < |b|\;. \tag{4}$$

The proof is a formalization of the usual proof of the basic property of geometric series, we derive the following equation

$$E(x,b) = MSP(x\cdot E(x,b) + A(b), 2|b|^2)$$

from which the proposition follows by use of the obvious decomposition

$$E(x,b) = \sum_{i=0}^{|b|} exp(|b| \dot{-} i, x, b) 2^{2i|b|^2} \ .$$

Proposition 10. *The following properties of exponentiation are proved using LIND on j and Prop. 9.*

$$|x| \leq |b| \wedge i+j \leq |b| \ \rightarrow \ x^i \cdot x^j = x^{i+j} \tag{5}$$
$$|xy| \leq |b| \wedge j \leq |b| \ \rightarrow \ (x \cdot y)^j = x^j \cdot y^j \tag{6}$$
$$|x^i| \leq |b| \wedge i \cdot j \leq |b| \ \rightarrow \ (x^i)^j = x^{i \cdot j} \tag{7}$$

3.3 Weak Products

In order to show that the Σ_1^b-definable functions of $C_2^0[div]$ are closed under weak products, we will formalize the reduction of ITERATED MULTIPLICATION to DIVISION of [1] in $C_2^0[div]$.

First, we show that $C_2^0[div]$ can determine the structure of the multiplicative groups $(\mathbb{Z}/q\mathbb{Z})^*$ for small prime powers q:

Lemma 11. $C_2^0[div]$ *proves: if* $q \leq |b|$ *is a prime power, then*

- $(\mathbb{Z}/q\mathbb{Z})^*$ *is cyclic for* q *odd or* $q = 2, 4$*, and*
- $(\mathbb{Z}/q\mathbb{Z})^*$ *is generated by* -1 *and* 5*, for* $q \geq 8$ *a power of* 2*.*

Lemma 11 is used to define, for every function $g(x)$ and small prime power q, a function $\tilde{Prod}_g(x,q)$ that is equal to $\prod_{i=0}^{|x|} g(i) \bmod q$, provided that $p \nmid g(i)$ for every $i \leq |x|$. This is done by weak summation of the indices of the values $g(i) \bmod q$ w.r.t. a generator of $(\mathbb{Z}/q\mathbb{Z})^*$, which is possible by Lemma 11. The following lemma is proved using the properties of weak sums and exponentiation, and shows that the definition is correct.

Lemma 12. $C_2^0[div]$ *proves*

$$p \nmid g(0) \ \rightarrow \ \tilde{Prod}_g(0,q) \equiv g(0) \pmod{q}$$

and

$$\forall i \leq |x| \, p \nmid g(i)$$
$$\rightarrow \ \tilde{Prod}_g(x,q) \equiv \tilde{Prod}_g(\lfloor \tfrac{1}{2} x \rfloor, q) \cdot g(|x|) \pmod{q}$$

Now for the general case, we define

$$e(i) := \mu e \!<\! |q| \, p^{e+1} \nmid g(i)$$
$$g'(i) := \left\lfloor \frac{g(i)}{p^{e(i)}} \right\rfloor$$
$$\bar{e}(x) := \sum_{i=0}^{|x|} e(i)$$
$$Prod_g(x,q) := p^{\bar{e}(x)} \cdot \tilde{Prod}_{g'}(x,q)$$

then the following lemma is easily proved by $PIND$ on x, using Lemma 8, (5) and Lemma 12.

Lemma 13. $C_2^0[div]$ *proves*

$$Prod_g(0,q) \equiv g(0) \pmod{q}$$
$$Prod_g(x,q) \equiv Prod_g(\lfloor \tfrac{1}{2} x \rfloor, q) \cdot g(|x|) \pmod{q}$$

Finally, the binomial theorem is used to define binomial coefficients from exponentiation.

$$\binom{m}{k} := LSP(MSP((2^{|b|}+1)^m, k|b|), |b|)$$

They provably satisfy the basic properties of binomial coefficients.

Lemma 14. $C_2^0[div]$ *proves the following properties of binomial coefficients for* $m, k < |b|$*:*

$$\binom{m}{0} = \binom{m}{m} = 1 \tag{8}$$
$$\binom{m}{k} = \binom{m-1}{k} + \binom{m-1}{k-1} \text{ for } 1 \leq k < m \tag{9}$$
$$\binom{m}{k} = \binom{m}{m-k} \tag{10}$$
$$k\binom{m}{k} = m\binom{m-1}{k-1} \tag{11}$$

Proof. The recursive equations (8) and (9) are proved by induction using the decomposition

$$(2^{|b|}+1)^m = 2^{|b|}(2^{|b|}+1)^{m-1} + (2^{|b|}+1)^{m-1}$$

and the observation

$$\begin{aligned} &LSP(MSP(2^{|b|} \cdot (2^{|b|}+1)^{m-1}, k|b|), |b|) \\ &\quad = LSP(MSP((2^{|b|}+1)^{m-1}, (k-1)|b|), |b|) \ . \end{aligned}$$

Note that the binomial coefficient $\binom{m}{k}$ is only defined for m and k small. Therefore we can prove (10) and (11) by induction from (8) and (9). □

We need the following further lemma about binomial coefficients:

Lemma 15. $C_2^0[div]$ *proves: If* q *is a prime power dividing* $\binom{2n}{n}$*, then* $q \leq 2n$*.*

This is proved by induction, using (10) and (11) from Lemma 14.

Lemma 16. *$C^0_2[div]$ proves the Chinese Remainder Theorem for small prime power moduli.*

The usual proof can be formalized directly. Now we are ready to prove the main theorem of this section.

Theorem 17. *The Σ^b_1-definable functions of $C^0_2[div]$ are closed under weak products.*

We sketch the main steps of the definition: By Lemma 15, the binomial coefficients $M(n) := \binom{2n^2}{n^2}$ provably in $C^0_2[div]$ form a *good modulus sequence* in the sense of [1]: all the prime powers in the factorization of $M(n)$ are small, but $M(n)$ is sufficiently large s.t. for all values $|x| \leq n$ and functions g with $|g(i)| \leq n$ for all $i \leq |x|$, we have $\prod_{i=0}^{|x|} g(i) \leq M(n)$.

We show that $C^0_2[div]$ can determine the factorization of $M(n)$ into prime powers, by using the bound from Lemma 15 and $BB\Sigma^b_0$. Finally we use Lemma 16 to compute the product $\prod_{i=0}^{|x|} g(i)$ by chinese remaindering from the values $Prod_g(x, q)$, for the prime powers q in the factorization of $M(n)$. The value obtained is congruent to $\prod_{i=0}^{|x|} g(i)$ modulo $M(n)$, and since $M(n)$ is sufficiently large, it is the actual product.

The so defined product provably in $C^0_2[div]$ satisfies the inductive definition of an iterated product.

Proposition 18. *$C^0_2[div]$ proves the following equations:*

$$\prod_{i=0}^{|0|} g(i) = g(0) \tag{12}$$

$$\prod_{i=0}^{|x|} g(i) = \Big(\prod_{i=0}^{|\lfloor \frac{1}{2}x \rfloor|} g(i) \Big) \cdot g(|x|) \tag{13}$$

This follows from Lemma 13 and the congruence property of the chinese remainder construction.

Proposition 19. *The following properties of weak products are proved using $PIND$ on x and Prop. 18.*

$$\prod_{i=0}^{|x|} g(i) \cdot h(i) = \prod_{i=0}^{|x|} g(i) \cdot \prod_{i=0}^{|x|} h(i) \tag{14}$$

$$\prod_{j=0}^{|y|} \prod_{i=0}^{|x|} g(i,j) = \prod_{i=0}^{|x|} \prod_{j=0}^{|y|} g(i,j) \tag{15}$$

$$\Big(\prod_{i=0}^{|x|} g(i) \Big)^k = \prod_{i=0}^{|x|} g(i)^k \tag{16}$$

4 Interpolation and the Diffie-Hellman Problem

The Diffie-Hellman problem is the following number-theoretic problem: given a large integer q and an element $g \in (\mathbb{Z}/q\mathbb{Z})^*$ of large order, compute $g^{ab} \bmod q$ from inputs $g^a \bmod q$ and $g^b \bmod q$. The hardness of this problem is a common assumption in public-key cryptography and is the basis for the security of the Diffie-Hellman key exchange protocol.

Obviously, the Diffie-Hellman problem is at most as hard as the discrete logarithm problem: given $g^a \bmod q$, compute a. It is also known that for integers $q = p_1 p_2$, where p_1, p_2 are prime, the Diffie-Hellman problem is at least as hard as factoring q [15, 12].

The formula $DH(u, v, q, G, X, Y, k)$ is given as the conjunction of modulo formulas expressing the following properties, where we write n for $|q|$:

- G codes a sequence $\langle g_0, \ldots, g_{2n-1} \rangle$ of length $2n$. Let $g := g_0$.
- $\forall i < 2n - 1 \;\; g_{i+1} = g_i^2 \bmod q$, which means $g_i = g^{2^i} \bmod q$.
- X codes a sequence $\langle x_0, \ldots, x_{n-1} \rangle$ of length n, and $\forall i < n - 1 \;\; x_{i+1} = x_i^2 \bmod q$.
- Likewise, Y codes a sequence $\langle y_0, \ldots, y_{n-1} \rangle$ of length n, and $\forall i < n - 1 \;\; y_{i+1} = y_i^2 \bmod q$.
- $x_0 = g^u \bmod q$ is expressed by
$$x_0 = \prod_{i<n} g_i^{Bit(u,i)} \bmod q \; .$$
- Similarly, $y_0 = g^v \bmod q$ is given by
$$y_0 = \prod_{i<n} g_i^{Bit(v,i)} \bmod q \; .$$
- Finally , $Bit(g^{uv} \bmod q, k) = 1$ is expressed as
$$Bit\Big(\prod_{j<n} \prod_{i<n} g_{i+j}^{Bit(u,i) \cdot Bit(v,j)} \bmod q, k \Big) = 1 \; .$$

By Thm. 17, the formula $DH(u, v, q, G, X, Y, k)$ is Δ^b_1 in $C^0_2[div]$.

Theorem 20. *$C^0_2[div]$ proves the implication*

$$DH(a, b, q, G, X, Y, k) \rightarrow DH(c, d, q, G, X, Y, k)$$

Proof. It obviously suffices to show that $C_2^0[div]$ can prove the congruence

$$\prod_{i,j<n} g_{i+j}^{Bit(a,i)\cdot Bit(b,j)} \equiv \prod_{i,j<n} g_{i+j}^{Bit(c,i)\cdot Bit(d,j)} \bmod q\,. \tag{17}$$

First, by induction on i, we can prove

$$\forall j<n\; z_j \equiv \prod_{i<n} g_{i+j}^{Bit(w,i)} \bmod q \tag{18}$$

where either z stands for x and w stands for a and c, or z stands for y and w stands for b and d. Then (17) follows by the following chain of congruences:

$$\begin{aligned}
&\prod_{j<n}\prod_{i<n} g_{i+j}^{Bit(a,i)\cdot Bit(b,j)} \\
&\quad\equiv \prod_{j<n}\Big(\prod_{i<n} g_{i+j}^{Bit(a,i)}\Big)^{Bit(b,j)} && \text{by (7,16)}\\
&\quad\equiv \prod_{j<n} x_j^{Bit(b,j)} && \text{by (18) for } x{,}a\\
&\quad\equiv \prod_{j<n}\Big(\prod_{i<n} g_{i+j}^{Bit(c,i)}\Big)^{Bit(b,j)} && \text{by (18) for } x{,}c\\
&\quad\equiv \prod_{j<n}\prod_{i<n} g_{i+j}^{Bit(c,i)\cdot Bit(b,j)} && \text{by (7,16)}\\
&\quad\equiv \prod_{i<n}\Big(\prod_{j<n} g_{i+j}^{Bit(b,i)}\Big)^{Bit(c,j)} && \text{by (15,7,16)}\\
&\quad\equiv \prod_{i<n} y_i^{Bit(c,i)} && \text{by (18) for } y{,}b\\
&\quad\equiv \prod_{i<n}\Big(\prod_{j<n} g_{i+j}^{Bit(d,j)}\Big)^{Bit(c,i)} && \text{by (18) for } y{,}d\\
&\quad\equiv \prod_{j<n}\prod_{i<n} g_{i+j}^{Bit(c,i)\cdot Bit(d,j)} \pmod q && \text{by (15,7,16).}
\end{aligned}$$

□

Main Theorem 21. *If $C_2^0[div]$ has feasible Δ_1^b-interpolation, then the Diffie-Hellman problem is solvable in P/poly.*

Proof. Using an interpolant $C(q,G,X,Y,k)$ of the Δ_1^b-theorem of Thm. 20, $g^{ab} \bmod q$ can be computed from g, q, $g^a \bmod q$ and $g^b \bmod q$ as follows:

First compute the sequences $G = \langle g^{2^i} \bmod q\rangle_{i<2n}$, $X = \langle g^{a2^i} \bmod q\rangle_{i<n}$ and $Y = \langle g^{b2^i} \bmod q\rangle_{i<n}$ by repeated squaring. Then for every $k < n$, the kth bit of $g^{ab} \bmod q$ is 1 if and only if $C(q,G,X,Y,k)$ is true, by the properties of an interpolant. Thus n parallel copies of $C(q,G,X,Y,k)$ for $0 \leq k < n$ form a circuit computing $g^{ab} \bmod q$, which is of size $n \cdot \|C(q,G,X,Y,k)\| + n^{O(1)}$. □

References

[1] P. W. Beame, S. A. Cook, and H. J. Hoover. Log depth circuits for division and related problems. *SIAM Journal of Computing*, 15:994–1003, 1986.

[2] M. L. Bonet, T. Pitassi, and R. Raz. No feasible interpolation for TC^0-Frege proofs. In *Proc. 38th Symposium on Foundations of Computer Science*, pages 254–263, 1997.

[3] S. R. Buss. *Bounded Arithmetic.* Bibliopolis, Napoli, 1986.

[4] S. R. Buss. Bounded arithmetic, cryptography and complexity. To appear in *Theoria*, 1998.

[5] P. Clote. A note on the relation between polynomial time functionals and Constable's class K. In H. Kleine Büning, editor, *Computer Science Logic*, volume 1092 of *Lecture Notes in Computer Science*, pages 145–160. Springer, 1996.

[6] P. Clote and G. Takeuti. First order bounded arithmetic and small boolean circuit complexity classes. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 154–218. Birkhäuser, Boston, 1995.

[7] R. Constable. Type 2 computational complexity. In *Proc. 5th ACM Symposium on Theory of Computing*, pages 108–121, 1973.

[8] J. Johannsen. A bounded arithmetic theory for constant depth threshold circuits. In P. Hájek, editor, *GÖDEL '96*, pages 224–234, 1996. Springer Lecture Notes in Logic 6.

[9] J. Johannsen and C. Pollett. On proofs about threshold circuits and counting hierarchies (extended abstract). In *Proc. 13th IEEE Symposium on Logic in Computer Science*, pages 444–452, 1998.

[10] J. Krajíček. *Bounded Arithmetic, Propositional Logic and Complexity Theory.* Cambridge University Press, 1995.

[11] J. Krajíček and P. Pudlák. Some consequences of cryptographical conjectures for S_2^1 and EF. *Information and Computation*, 140:82–94, 1998. Preliminary version in D. Leivant, ed., *LCC '94*, Springer LNCS 960, 1995.

[12] K. McCurely. A key distribution system equivalent to factoring. *Journal of Cryptology*, 1:95–105, 1988.

[13] V. R. Pratt. Every prime has a succinct certificate. *SIAM Journal of Computing*, 4:214–220, 1975.

[14] J. H. Reif. Logarithmic depth circuits for algebraic functions. *SIAM Journal of Computing*, 15:231–242, 1986.

[15] Z. Shmuely. Composite Diffie-Hellman public-key generating systems are hard to break. Technical Report 356, Computer Science Department, Technion, Israel, 1985.

[16] G. Takeuti. *RSUV* isomorphisms. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, volume 23 of *Oxford Logic Guides*, pages 364–386. Clarendon Press, Oxford, 1993.