The computational efficacy of finite-field arithmetic

Carl Sturtivant
Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, USA

Gudmund Skovbjerg Frandsen
Department of Computer Science, Aarhus University, 8000 Aarhus C, Denmark

Communicated by A. Schönhage
Received February 1989
Revised November 1991

Abstract

We investigate the computational power of finite-field arithmetic operations as compared to Boolean operations. We pursue this goal in a representation-independent fashion. We define a good representation of the finite fields to be essentially one in which the field arithmetic operations have polynomial-size Boolean circuits. We exhibit a function \( f_p \) on the prime fields with two properties: first, \( f_p \) has a polynomial-size Boolean circuit in any good representation, i.e. \( f_p \) is easy to compute with general operations; second, any function that has polynomial-size Boolean circuits in some good representation also has polynomial-size arithmetic circuits if and only if \( f_p \) has polynomial-size arithmetic circuits. Informally, \( f_p \) is the hardest function to compute with arithmetic that has small Boolean circuits.

We reduce the function \( f_p \) to the pair of functions \( g_p = \sum_{k=1}^{p-1} x^k/k \) on the field \( \mathbb{F}_p \), and \( m_p \) on \( \mathbb{Z}_p \). Here \( m_p \) is the "modulo \( p \)" function defined in the natural way. We show that \( f_p \) has polynomial-size arithmetic circuits if and only if \( g_p \) and \( m_p \) have polynomial-size arithmetic circuits, the latter being arithmetic circuits over the ring \( \mathbb{Z}_p \). Finally, we establish a connection of \( f_p \) and \( m_p \) with the Bernoulli polynomials and determine the coefficients of the unique degree \( p-1 \) polynomial over \( \mathbb{F}_p \) that computes \( f_p \).

1. Introduction

In recent years, finite-field arithmetic has had a growing impact on Boolean circuit complexity; see e.g. [10, 11]. This research has focused on the incompatibility of

Correspondence to: C. Sturtivant, Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, USA.

0304-3975/93/$06.00 © 1993 — Elsevier Science Publishers B.V. All rights reserved
computations modulo different characteristics. These investigations were carried out on arbitrary fan-in circuits of constant depth and polynomial size. In a recent paper [1], it is shown that by allowing the field size to increase with the size of the input, the computational restrictions of a particular characteristic could be overcome, and the whole of TC$^0$ captured.

However, in [1], this result is carried through at some fixed characteristic. In this paper, we investigate the analogous situation where the characteristic is unbounded and may increase with the input size. This seems quite different because it is no longer clear that field arithmetic operations are anywhere near as powerful as Boolean operations even at polynomial size.

We investigate the relationship between finite-field arithmetic circuit size and Boolean circuit size. To relate these two, we must discuss representations of finite fields in order that Boolean operations may be performed on field elements. We deal with representations quite generally, defining a good representation in such a way as to capture anything that might remotely be considered useful.

Another motivation for considering this problem is provided by the discrete logarithm problem on finite fields. The discrete log appears to be hard to compute and to be a one-way function. It is the basis of many cryptographic schemes [9]. This function is defined on all finite fields and it would be natural to investigate its arithmetic complexity, should such complexity prove to be polynomially related to its Boolean complexity.

The paper proceeds as follows. We define a good representation to be essentially one where arithmetic has polynomial-size Boolean circuits. We show that finite-field arithmetic is as powerful as Boolean operations at polynomial size if and only if, in any good representation of field elements as bit strings, there is a polynomial-size arithmetic circuit that computes from a field element its representation and vice versa. (Here we regard the zeros and ones of the representation as the field elements zero and one in every finite field).

We call a representation that has this bit accessibility property a strong representation.

We show that if a strong representation exists then all good representations are strong. Thus, the question of the efficacy of arithmetic as compared to Boolean operations is reduced to the question of whether a standard representation of the finite fields is strong. This is easily reduced to the case of the prime fields $\mathbb{F}_p$ for odd $p$, and we may take a standard representation to be the integers 0 to $p-1$ with modulo-$p$ arithmetic.

We define the function $f_p: \mathbb{F}_p \rightarrow \mathbb{F}_p$ as follows:

$$f_p(x) = \left(\frac{x - x^p}{p}\right) \mod p,$$

where the arithmetic within parentheses is integer arithmetic. Since $x - x^p \equiv 0 \mod p$, the division by $p$ is well-defined. We show that the standard representation of $\mathbb{F}_p$ is strong if and only if $f_p$ has polynomial-size arithmetic circuits. Clearly, $f_p$ has
polynomial-size Boolean circuits in the standard representation, since we may compute \( x - x^p \) modulo \( p^2 \) in binary and perform the division by \( p \) explicitly. Thus, the efficacy of arithmetic in finite fields has been reduced to the question of the arithmetic complexity of \( f_p \). The function \( f_p \) is not the only one with this property, but it seems canonical in its form and expressive power.

We define the function \( g_p : \mathbb{F}_p \rightarrow \mathbb{F}_p \) as follows:

\[
g_p(x) = \left( \frac{(x - 1)^p + 1 - x^p}{p} \right) \mod p,
\]

where, just as with \( f_p \), the arithmetic within parentheses is integer arithmetic. This function has a simple expansion as a polynomial of degree \( p - 1 \) within the field:

\[
g_p = \sum_{k=1}^{p-1} x^k / k.
\]

We define the function \( m_p : \mathbb{Z}_{p^2} \rightarrow \mathbb{Z}_{p^2} \) by \( m_p(x) = x \mod p \) in the intuitive concrete sense. We show that \( f_p \) has a polynomial-size arithmetic circuit if and only if \( g_p \) has a polynomial-size arithmetic circuit over \( \mathbb{F}_p \) and \( m_p \) has a polynomial-size arithmetic circuit over the ring \( \mathbb{Z}_{p^2} \).

2. Preliminaries

We take an arithmetic circuit over a field \( K \) to be a function and a straight-line program over \( K \) defining it. (We treat Boolean circuits similarly.) The size of the circuit is the length of the straight-line program. Thus, for example, if \( f : K^n \rightarrow K^m \) and \( g : K^r \rightarrow K^n \) are arithmetic circuits, we write \( f \circ g : K^r \rightarrow K^m \) for the composition of \( f \) and \( g \) both as functions and as straight-line programs in the obvious way, the size of \( f \circ g \) being the size of \( f \) plus the size of \( g \).

It is implicit in the rest of this paper that the terms “function” and “circuit” are to denote a family of functions or a family of circuits, respectively, indexed over all finite fields \( \mathbb{F}_q \) (or over all prime fields \( \mathbb{F}_p \)). As usual, the complexity of a function will be taken to mean the size of the smallest circuit computing the function. Throughout, let \( n = \lceil \log q \rceil \) be the “input size”. Thus, all circuit sizes will be expressed as functions of \( n \). As usual, we will use the term “polynomial-size circuit” to mean a circuit whose size is bounded above by some fixed polynomial in \( n \). All references to “polynomial size” will be with respect to \( n \).

3. Good representations of \( \mathbb{F}_q \)

A good representation is intended to be one in which an arithmetic circuit may be efficiently implemented as a Boolean circuit, i.e. one in which arithmetic is efficient. Since we use polynomial-size circuits as our definition of “efficient” here, we call a good representation, in this context, a polynomial representation. The following definition is intended to capture the widest possible class of good representations that are compatible with our definition of “efficient”.

Definition 3.1. A polynomial representation is a surjective function \( \phi_q : S_q \rightarrow \mathbb{F}_q \), where \( S_q \subseteq \{0, 1\}^{t(q)} \) and \( t(q) \) is a polynomially bounded function (i.e. \( t(q) = n^{O(1)} \)), where, furthermore, there exist polynomial-size Boolean circuits
\[
\begin{align*}
\alpha_q : S_q \times S_q & \rightarrow S_q, \\
\mu_q : S_q \times S_q & \rightarrow S_q, \\
\zeta_q : S_q & \rightarrow \{0, 1\},
\end{align*}
\]
simulating field addition, multiplication and a zero test, respectively, i.e. satisfying
\[
\begin{align*}
\phi_q(\alpha_q(x, y)) &= \phi_q(x) + \phi_q(y), \\
\phi_q(\mu_q(x, y)) &= \phi_q(x) \cdot \phi_q(y), \\
\zeta_q(x) &= 1 \text{ if and only if } \phi_q(x) = 0,
\end{align*}
\]
for all \( x, y \in S_q \).

We include a zero test in the definition of a polynomial representation for the following reason: in a polynomial representation, a given field element may be represented by an exponential number of distinct bit strings. The existence of an efficient zero test at least makes the computation of predicates possible. Without this facility, it may not be possible to recognize the output of a computation at all, in which case such computations cannot be regarded as meaningful.

Note that if we restrict our attention to polynomial representations that are polynomially-onto (i.e. for which there are at most \( n^{O(1)} \) representatives of any field element), then an efficient zero test always exists, and the last part of the definition is unnecessary. In particular, this is the case for injective polynomial representations.

Lemma 3.2. There exists a polynomial representation of the finite fields.

Proof. A standard representation of \( \mathbb{F}_p \) as binary numbers less than \( p \) (with modular arithmetic obviously having polynomial-size circuits), together with a standard representation of \( \mathbb{F}_{p^k} \) as polynomials over \( \mathbb{F}_p \) of degree less than \( k \) (with arithmetic modulo a fixed irreducible polynomial obviously having polynomial-size circuits), as described in [7], clearly defines a polynomial representation. \( \square \)

The proof of the following lemma is trivial from the definition of a polynomial representation.

Lemma 3.3 (Implementation lemma). If \( \Gamma_q^{m,k} : \mathbb{F}_q^m \rightarrow \mathbb{F}_q^k \) is an arithmetic circuit of size \( C = C(q, m, k) \), and \( \phi_q \) is a polynomial representation with domain \( S_q \) of size \( t(q) \), then there exists a corresponding Boolean circuit \( \gamma_q^{m,k} : (\{0, 1\}^{t(q)})^m \rightarrow (\{0, 1\}^{t(q)})^k \) satisfying \( \gamma_q^{m,k}(S_q^m) \subseteq S_q^k \) of size \( (C + n)^{O(1)} \) satisfying also, for \( 1 \leq i \leq k \), \( \phi_q(\gamma_q^{m,k}(x_1, \ldots, x_m)) = \Gamma_q^{m,k}(\phi_q(x_1), \ldots, \phi_q(x_m)) \), for all \( x_1, \ldots, x_m \in S_q \).
Informally, this lemma asserts that in a polynomial representation, a polynomial-size arithmetic circuit always has a corresponding polynomial-size Boolean circuit. (In fact, it is slightly less trivial to show that this is also true in the case when arbitrary fan-in arithmetic circuits are the model of arithmetic computation, where size is defined to be the number of $\Sigma$ and $\Pi$ gates.)

4. Strong representations of $\mathbb{F}_q$

A strong representation is intended to be one in which, if a Boolean circuit computes some finite-field function, then there is an equally efficient arithmetic circuit computing the same function. (In our context, “equally efficient” will mean “of polynomially related size”.) Thus, strong is the dual concept of good.

Just as the existence of a good representation (polynomial representation) shows that what may be computed efficiently with arithmetic may be computed efficiently by Boolean operations, so will the existence of a strong representation imply that what may be computed efficiently by Boolean operations may be computed efficiently by arithmetic operations. Thus, we will have formalized the question of the efficacy of finite-field arithmetic as the question of whether a strong representation exists. (Note that, if a strong representation does not exist, we need to show that finite-field arithmetic is not as effective as Boolean operations in any polynomial representation to complete this formalization. We do this in Theorem 7.3.)

Just as the definition of a good representation involves the existence of certain efficient Boolean circuits (performing arithmetic), so does the definition of strong involve the existence of certain efficient arithmetic circuits. The purpose of these circuits will become apparent in the following definition, in which we regard $\{0, 1\}$ as a subset of $\mathbb{F}_q$ for all $q$ in the natural way.

**Definition 4.1.** A strong polynomial representation $\phi_q : S_q \rightarrow \mathbb{F}_q$, where $S_q \subseteq \{0, 1\}^{t(q)}$, is a polynomial representation with the property that there are polynomial-size arithmetic circuits

$$i_q : \mathbb{F}_q \rightarrow \mathbb{F}_q^{t(q)} \quad \text{and} \quad o_q : \mathbb{F}_q^{t(q)} \rightarrow \mathbb{F}_q,$$

where the image of $i_q$ is contained in $S_q$, satisfying

$$\phi_q(i_q(z)) = z \quad \text{and} \quad o_q(x) = \phi_q(x)$$

for all $z \in \mathbb{F}_q$ and for all $x \in S_q$, i.e. $i_q \subseteq \phi_q^{-1}$ and $o_q$ is an extension of $\phi_q$ with respect to the natural embedding $S_q \subseteq \mathbb{F}_q^{t(q)}$.

Intuitively, a strong polynomial representation is one in which entry to or exit from the representation can be accomplished efficiently with arithmetic. We now show that this property does indeed imply the truth of the claim at the beginning of the section that any Boolean circuit may be “abstracted” to a correspondingly efficient arithmetic circuit.
Lemma 4.2 (Abstraction lemma). If $\gamma_{q}^{m,k} : \{0,1\}^{\tau(q)} \rightarrow \{0,1\}^{\tau(q)}$ is a Boolean circuit of size $C = C(q, m, k)$, with $\gamma_{q}^{m,k}(S_{q}^{m}) \subseteq S_{q}^{m}$, and $\phi_{q} : S_{q} \rightarrow F_{q}$ is a strong polynomial representation where $S_{q} \subseteq \{0,1\}^{\tau(q)}$, then there exists a corresponding arithmetic circuit $\Gamma_{q}^{m,k} : \mathbb{F}_{q}^{m} \rightarrow \mathbb{F}_{q}^{k}$ of size $(C + n)^{(1)}$ satisfying, for $1 \leq i \leq k$,

$$\phi_{q}(\gamma_{q}^{m,k}(x_{1}, \ldots, x_{m})) = \Gamma_{q}^{m,k}(\phi_{q}(x_{1}), \ldots, \phi_{q}(x_{m})), \text{ for all } x_{1}, \ldots, x_{m} \in S_{q}.$$  

Proof. We give the proof in the case where $m = k = 1$, as the full proof is a trivial generalization of this case. Then, because $\phi_{q}$ is a strong polynomial representation, there exist polynomial-size circuits $i_{q}$ and $o_{q}$ for efficiently entering and exiting the representation. Consequently, we may take the Boolean circuit $\gamma_{q}^{m,k} : \{0,1\}^{\tau(q)} \rightarrow \{0,1\}^{\tau(q)}$ and modify it into an arithmetic circuit $\gamma_{q}^{m,k} : \mathbb{F}_{q}^{m} \rightarrow \mathbb{F}_{q}^{k}$ by replacing a negation gate $\neg g$ by $1 - g$, an and gate $g_{1} \land g_{2}$ by $g_{1} \cdot g_{2}$, etc., so that $\gamma_{q}^{m,k}$ on zero-one inputs computes the same function as $\gamma_{q}$. Now the arithmetic circuit $\Gamma_{q}^{m,k} : \mathbb{F}_{q}^{m} \rightarrow \mathbb{F}_{q}^{k}$ defined by $\Gamma_{q}^{m,k}(x) = o_{q}(\gamma_{q}^{m,k}(i_{q}(x)))$ for all $x \in \mathbb{F}_{q}$ obviously satisfies the statement of the lemma. \qed

5. Equivalence of representations

Two polynomial representations $\phi_{q}$ and $\phi'_{q}$ are effectively the same for computational purposes if they can be translated into each other efficiently. We have not shown the existence of a strong polynomial representation, but we now show that, if one exists, then all polynomial representations are equivalent in this sense, and, thus, they are all strong.

In what follows, let $\phi_{q} : S_{q} \rightarrow \mathbb{F}_{q}$ and $\phi'_{q} : S'_{q} \rightarrow \mathbb{F}_{q}$ be any polynomial representations with $S_{q} \subseteq \{0,1\}^{\tau(q)}$ and $S'_{q} \subseteq \{0,1\}^{\tau(q)}$.

Definition 5.1. We say that $\phi_{q}$ $p$-translates into $\phi'_{q}$ (written as $\phi_{q} \leq_{p} \phi'_{q}$) if and only if there exists a polynomial-size Boolean circuit $T_{q} : \{0,1\}^{\tau(q)} \rightarrow \{0,1\}^{\tau(q)}$ satisfying

$$\phi_{q}(x) = \phi'_{q}(T_{q}(x))$$

for all $x \in S_{q}$. Furthermore, we say that $\phi_{q}$ and $\phi'_{q}$ are polynomially equivalent if and only if $\phi_{q} \leq_{p} \phi'_{q}$ and $\phi'_{q} \leq_{p} \phi_{q}$.
Theorem 5.2. If \( \phi_q \) is a strong polynomial representation and \( \phi'_q \) is any polynomial representation, then \( \phi_q \) is polynomially equivalent to \( \phi'_q \).

Proof. First we show that \( \phi_q \leq_p \phi'_q \).

Since \( \phi_q \) is strong, we have the polynomial-size arithmetic circuit \( o_q : \mathbb{F}_q^{t(q)} \to \mathbb{F}_q \) satisfying \( o_q(x) = \phi_q(x) \) for \( x \in S_q \). Now, implement this circuit as a polynomial-size Boolean circuit using the polynomial representation \( \phi'_q \), following Lemma 3.3; call the resulting circuit \( \gamma_q : \mathbb{F}_q^{t(q)} \to \mathbb{F}_q \).

By definition, the new circuit satisfies the following condition: if \( y_1, \ldots, y_{t(q)} \in S'_q \), where \( \phi'_q(y_i) = x_i \in \{0, 1\} \) for \( 1 \leq i \leq t(q) \) and \( x = (x_1, \ldots, x_{t(q)}) \in S_q \), then \( \phi'_q(z) = \phi_q(x) \). (This simply asserts that \( \gamma_q \) takes a \( \phi'_q \) representation of a field element, and computes a \( \phi_q \) representation of that field element.) Now let \( b_q : \{0, 1\} \to \{0, 1\}^{t(q)} \) be a polynomial-size Boolean circuit satisfying \( \phi'_q(b_q(k)) = k \) for \( k \in \{0, 1\} \). (This is just a circuit computing, from a bit, a \( \phi'_q \) representation of that bit.)

Thus, \( \phi'_q(\gamma_q(b_q(x_1), \ldots, b_q(x_{t(q)}))) = \phi_q(x) \) for \( x = (x_1, \ldots, x_{t(q)}) \in S_q \), and \( \gamma_q \), with \( t(q) \) copies of \( b_q \), gives a polynomial-size circuit \( T_q \) translating \( \phi_q \) into \( \phi'_q \) (Definition 5.1).

Now we show that \( \phi'_q \leq_p \phi_q \).

Since \( \phi_q \) is strong, we have the polynomial-size arithmetic circuit \( i_q : \mathbb{F}_q \to S'_q \) satisfying \( i_q(z) = z \) for \( z \in \mathbb{F}_q \) (where the image of \( i_q \) lies in \( \{0, 1\}^{t(q)} \)). Now implement this circuit as a polynomial-size Boolean circuit using the polynomial representation \( \phi'_q \), following Lemma 3.3; call the resulting circuit \( \gamma_q : \mathbb{F}_q^{t(q)} \to \{0, 1\}^{t(q)} \). By definition, the new circuit satisfies the following condition: if \( \gamma_q(z) = (y_1, \ldots, y_{t(q)}) \) for \( y_1, y_z \in S'_q \) (for \( 1 \leq i \leq t(q) \)) then \( x_i = \phi'_q(y_i) \in \{0, 1\} \), \( x = (x_1, \ldots, x_{t(q)}) \in S_q \) and \( \phi_q(x) = \phi'_q(z) \). (This simply asserts that \( \gamma_q \) takes a \( \phi'_q \) representation of a field element, and computes \( \phi_q \) representations of the zeros and ones of the \( \phi_q \) representation of that field element.)

Now let \( b_q : \{0, 1\} \to \{0, 1\} \) be a polynomial-size Boolean circuit (constructed using \( \gamma_q \), the zero-test circuit for \( \phi'_q \)) such that, for \( k \in \mathbb{F}_q \) with \( \phi'_q(k) \in \{0, 1\} \), we have \( b_q(k) = \phi'_q(k) \). (This is just a circuit computing a bit from any \( \phi'_q \) representation of that bit.)

Thus, if \( b_q(\gamma_q(z_i)) = x_i \) for \( 1 \leq i \leq t(q) \), \( z \in S'_q \), then \( x = (x_1, \ldots, x_{t(q)}) \in S_q \) has \( \phi_q(x) = \phi'_q(z) \). \( \gamma_q \), with \( t(q) \) copies of \( b_q \), gives a polynomial-size circuit translating from \( \phi'_q \) to \( \phi_q \). \( \square \)

Note that the above proof does not, in fact, assume that \( \phi_q \) is good, only that \( t(q) \) is polynomially bounded.

Corollary 5.3. If a strong polynomial representation exists then all polynomial representations are polynomially equivalent and strong.

Proof. The first part follows immediately from Theorem 5.2. The second part follows from the first, and from the assertion that if a polynomial representation \( \phi_q \) is polynomially equivalent to a strong polynomial representation \( \phi_q \), then \( \phi'_q \) is also strong, which can be seen as follows. Since \( \phi_q \) is strong, polynomial-size circuits \( i_q \) and
o_q exist, according to the definition. We show the existence of corresponding polynomial-size circuits for \( \phi_q \), namely, \( i_q \) and \( o_q \).

Since \( \phi_q \leq \phi_q^{\prime} \), we have a translation circuit \( T_q \) of polynomial size following the definition, and we may take \( i_q = T_q \cdot i_q \); the circuit \( o_q = o_q \cdot T_q \) is constructed similarly, using \( \phi_q^{\prime} \leq \phi_q \). The correctness of these circuits is easily verified; they are polynomial-size and, so, \( \phi_q^{\prime} \) is strong.

6. Standard representations of \( \mathbb{F}_q \)

Corollary 5.3 shows that we may ask if a strong polynomial representation exists by asking if a standard representation is a strong polynomial representation. (By “a standard representation”, we mean one of the well-known representations described in the proof of Lemma 3.2.) We now investigate this question. First, we give a reduction to the prime fields \( \mathbb{F}_p \), where \( p \) is prime.

**Lemma 6.1.** Any standard representation is a strong polynomial representation if and only if the standard representation of the prime fields is strong.

**Proof.** If the standard representation of the prime fields is not strong, then, clearly, no standard representation of all the fields is strong, since such includes the standard representation of the prime fields. Conversely, given arithmetic in \( \mathbb{F}_p \), it is easy to simulate arithmetic in \( \mathbb{F}_p^{\prime} \), by using circuits for polynomial arithmetic over \( \mathbb{F}_p \) in the obvious way, modulo an irreducible polynomial \( h(x) \) of degree \( k \).

If \( \theta \in \mathbb{F}_p^{\prime} \) is a fixed root of \( h(x) \), then a standard representation of \( u \in \mathbb{F}_p^{\prime} \) over \( \mathbb{F}_p \) is just the tuple \( (u_0, \ldots, u_{k-1}) \in \mathbb{F}_p^k \), where \( u = \sum_{j=0}^{k-1} u_j \theta^j \). (See [7, p. 34]. These expressions add and multiply like polynomials modulo the relation \( h(\theta) = 0 \).)

Given \( (u_0, \ldots, u_{k-1}) \), there is obviously a polynomial-size arithmetic circuit to compute \( u = \sum_{j=0}^{k-1} u_j \theta^j \). Equally, given \( u \), there is a polynomial-size arithmetic circuit computing \( (u_0, \ldots, u_{k-1}) \). This follows because the conjugate linear relations \( u' = \sum_{j=0}^{k-1} u_j \theta^{j+p} \) are independent [7, p. 62] and, so, each of the \( u_j \) is a fixed linear combination of the \( u_{j'} \) for \( 0 \leq j' \leq k \). The latter may be computed efficiently by repeated squaring.

Consequently, if \( \mathbb{F}_p^{\prime} \)'s standard representation as binary numbers is strong, then we may efficiently find the bit representation of \( u \) with arithmetic, by first finding \( (u_0, \ldots, u_{k-1}) \) efficiently as above, and then finding the bit representations of the \( u_j \) in \( \mathbb{F}_p^{\prime} \)'s standard representation. This gives the polynomial-size circuit \( i_{\rho} \) for a standard representation of \( \mathbb{F}_p^{\prime} \). The circuit \( o_{\rho} \) is similarly constructed.

7. The standard representation of the prime fields

We now investigate whether the standard representation of the prime fields \( \mathbb{F}_p \) is strong. Suppose \( u \in \mathbb{F}_p \) and \( u = \sum_{i} u_i 2^i \) is the binary expansion of \( u \) in the standard...
Computational efficacy of finite-field arithmetic

representation; so, \( u_i \in \{0, 1\} \). Clearly, the sum \( \sum u_i 2^i \) is also correct and meaningful within the field (i.e., \( 2 = 1 + 1 = 1 \pmod{2} \); employ field arithmetic throughout). Thus, the circuit \( o_p \) is obviously easy to construct for this representation.

(If \( p = 2 \) then a field element is equal to its representation, we assume \( p > 2 \) throughout.) It remains to consider whether the polynomial-size circuit \( i_p \) exists. Suppose now we define \( l_p : \mathbb{F}_p \rightarrow \{0, 1\} \), the last bit problem for the prime fields, as follows.

**Definition 7.1.** For \( y \in \mathbb{F}_p \), let \( l_p(y) \) be the least significant bit of the standard representation of \( y \); i.e., if we identify \( \mathbb{F}_p \) with \( \{0, 1, \ldots, p-1\} \) then \( l_p \) is zero on even numbers and one on odd numbers.

**Theorem 7.2.** The following are equivalent:

1. Finite-field arithmetic is as effective as Boolean operations; i.e., in any polynomial representation, any function with a polynomial-size Boolean circuit has a polynomial-size arithmetic circuit.

2. There is a polynomial-size arithmetic circuit for \( l_p \).

3. There exists a strong polynomial representation.

4. All polynomial representations are strong.

**Proof.** (1)\( \Rightarrow \) (2): Using the standard representation, \( l_p \) has a trivial polynomial-size Boolean circuit and, so, by (1), \( l_p \) has a polynomial-size arithmetic circuit.

(2)\( \Rightarrow \) (3): A standard representation of \( \mathbb{F}_p \) is strong if \( l_p \) has a polynomial-size circuit.

By Lemma 6.1, we only need consider the prime fields. The circuit \( o_p \) is easily constructed to compute \( \sum u_i 2^i \) in polynomial size, and \( i_p \) is constructed using about \( n \) copies of the circuit for \( l_p \) (peel off one bit at a time, subtract, divide by 2).

(3)\( \Rightarrow \) (4): This is Corollary 5.3.

(4)\( \Rightarrow \) (1): This is Lemma 4.2.

**Theorem 7.3.** If \( l_p \) does not have a polynomial-size arithmetic circuit, then in any polynomial representation \( \phi_p : S_p \rightarrow \mathbb{F}_p \), where \( S_p \subseteq \{0, 1\}^{(p)} \), arithmetic is ineffective compared to Boolean operations in the following sense: there exists a finite-field function with a polynomial-size Boolean circuit in the representation \( \phi_p \) that does not have a polynomial-size arithmetic circuit.

**Proof.** By Theorem 7.2, \( \phi_p \) is not strong; therefore, any arithmetic circuit to compute the functions that would have been computed by one of \( i_p \) and \( o_p \), if \( \phi_p \) were strong, cannot be polynomial-size. On the other hand, a pair of these functions have polynomial-size Boolean circuits. The \( i_p \) circuit must simply map each bit \( b \) in the representation of a field element into some bit string \( x \in \phi_p^{-1}(b) \), and the \( o_p \) circuit must do the opposite. The latter requires a judicious use of \( \zeta_p \), the zero-test circuit for \( \phi_p \), as in the proof of Theorem 5.2.

We have now established that the efficacy of finite-field arithmetic depends upon whether the last bit problem \( l_p \) has polynomial-size arithmetic circuits. A number of
other functions have just this same property, apparently because of their dependency
on the standard representation: for example, \( \min \) or \( \exp \) defined in the obvious way, or
the computation of arithmetic operations modulo composite numbers (suitably de-
defined). To prove such a statement, it suffices to show two things: first, that the function
in question has a polynomial-size Boolean circuit in the standard representation;
second, that if the function in question has a polynomial-size arithmetic circuit, then \( l_p \)
has a polynomial-size arithmetic circuit (i.e. a reduction of \( l_p \) to the function in
question). Consequently, such functions form a completeness class, and we will call
such a function \textit{prime-field-complete}. (If a function is only known to satisfy the second
criterion, then we suggest using the term \textit{prime field hard} instead.)

For example, let \( g \in \mathbb{F}_p \) be a primitive element and let \( \exp_p : \mathbb{F}_p \rightarrow \mathbb{F}_p \) be defined by
\( \exp_p(x) = g^x \), where the field element in the exponent is regarded as its standard
representation as a number less than \( p \). Clearly, \( \exp_p \) has a polynomial-size Boolean
circuit in the standard representation, since this may be computed by modular
repeated squaring. Also, if \( \exp_p \) has a polynomial-size arithmetic circuit, then the
identity \( l_p(x) = \frac{1}{2}(1 - \exp_p(x)(p^{1/2}) \) gives a polynomial-size arithmetic circuit for \( l_p \).
Consequently, \( \exp_p \) is prime-field-complete.

In the next section we try to find a canonical prime-field-complete problem, and we
investigate its arithmetic complexity.

8. \( \mathbb{Z}_p^2 \) arithmetic and Witt vectors

Let \( \mathbb{Z}_p^2 \) be the ring of integers modulo \( p^2 \). We now consider representations of
\( \mathbb{Z}_p^2 \) over \( \mathbb{F}_p \). Since \( \mathbb{Z}_p^2 \) has characteristic \( p^2 \) which is distinct from that of \( \mathbb{F}_p \), some
prime-field-complete functions emerge when we consider implementing \( \mathbb{Z}_p^2 \) arithmetic
with \( \mathbb{F}_p \) arithmetic in some representation of \( \mathbb{Z}_p^2 \). The cardinality of \( \mathbb{Z}_p^2 \) is nice for such
representations because it is equal to the cardinality of \( \mathbb{F}_p \times \mathbb{F}_p \). Thus, it is natural to
consider, for our purposes, a representation of \( \mathbb{Z}_p^2 \) to be a bijection \( \psi_p : \mathbb{F}_p \times \mathbb{F}_p \rightarrow \mathbb{Z}_p^2 \),
i.e. each pair of field elements represents a distinct ring element. One familiar such
representation is as two-digit numbers in the base \( p \).

In the remainder of this paper, we regard \( \mathbb{F}_p \) as a subset of \( \mathbb{Z}_p^2 \), by regarding \( \mathbb{F}_p \) as
\( \{0, 1, \ldots, p-1\} \) and \( \mathbb{Z}_p \) as \( \{0, 1, \ldots, p^2-1\} \) in the natural way, and we take
\( h_p : \mathbb{Z}_p^2 \rightarrow \mathbb{F}_p \) to be the standard epimorphism, i.e. \( h_p(x) = x \mod p \) in the concrete sense.

**Definition 8.1.** The standard (base \( p \)) representation of \( \mathbb{Z}_p^2 \) over \( \mathbb{F}_p \) is the bijection
\( \psi_p : \mathbb{F}_p \times \mathbb{F}_p \rightarrow \mathbb{Z}_p^2 \) given by \( \psi_p(x_0, x_1) = x_0 + x_1 \mathbb{I} p \) (arithmetic in \( \mathbb{Z}_p^2 \)); furthermore, we
write \( \psi_p(x_0, x_1) \) as \( \langle x_0, x_1 \rangle \) to indicate that \( (x_0, x_1) \in \mathbb{F}_p \times \mathbb{F}_p \) represents \( \langle x_0, x_1 \rangle \in \mathbb{Z}_p^2 \)
in the base-\( p \) representation. Clearly, we then have \( h_p(\langle x_0, x_1 \rangle) = x_0 \).

We now consider arithmetic in the standard representation of \( \mathbb{Z}_p^2 \). Suppose that we have
\[ \langle x_0, x_1 \rangle + \langle y_0, y_1 \rangle - \langle z_0, z_1 \rangle; \]
then how are \((z_0, z_1)\) related to \((x_0, x_1)\) and \((y_0, y_1)\)? We define carry functions to expose such relationships.

**Definition 8.2.** The additive carry function \(\sigma_p\) and the multiplicative carry function \(\pi_p\) are defined by

\[
\begin{align*}
\sigma_p, \pi_p : \mathbb{F}_p \times \mathbb{F}_p & \rightarrow \mathbb{F}_p, \\
\langle x_0, x_1 \rangle + \langle y_0, y_1 \rangle & = \langle x_0 + y_0, x_1 + y_1 + \sigma_p(x_0, y_0) \rangle, \\
\langle x_0, x_1 \rangle \cdot \langle y_0, y_1 \rangle & = \langle x_0 y_0, x_0 y_1 + x_1 y_0 + \pi_p(x_0, y_0) \rangle,
\end{align*}
\]

for all \(x_0, x_1, y_0, y_1 \in \mathbb{F}_p\). (Note that the arithmetic inside the representation is, of course, \(\mathbb{F}_p\) arithmetic.)

**Theorem 8.3.** The carry functions \(\sigma_p\) and \(\pi_p\) are prime-field-complete.

**Proof.** It is easy to see that \(\sigma_p\) and \(\pi_p\) have polynomial-size Boolean circuits in the standard representation of \(\mathbb{F}_p\). Furthermore, the following identities are easy to verify:

\[
l_p(x) = \sigma_p \left( \frac{x}{2}, \frac{x}{2} \right) = \pi_p \left( \frac{x}{2}, \frac{x}{2} \right).
\]

We now introduce the Witt representation of \(\mathbb{Z}_{p^2}\). The theory of Witt vectors is described in \([2-5, 12]\). For the present, we simply render the Witt vector representation of \(\mathbb{Z}_{p^2}\) in a way that emphasizes the carryless nature of this representation, a property that is very attractive for our purposes.

**Definition 8.4.** The Witt representation of \(\mathbb{Z}_{p^2}\) is the unique bijection \(\chi_p : \mathbb{F}_p \times \mathbb{F}_p \rightarrow \mathbb{Z}_{p^2}\), denoted in this paper by \(\chi_p(x_0, x_1) = [x_0, x_1]\), satisfying the following four conditions:

\[
\begin{align*}
(a) \quad [x_0, x_1] \cdot [y_0, y_1] & = [x_0 y_0 + x_0 y_1 + x_1 y_0], \\
(b) \quad [0, x_1] + [0, y_1] & = [x_1 + y_1], \\
(c) \quad [x_0, 0] + [0, y_1] & = [x_0, y_1], \\
(d) \quad [0, 1] & = p,
\end{align*}
\]

for all \(x_0, x_1, y_0, y_1 \in \mathbb{F}_p\).

We will verify later that Definition 8.4 does define a representation, and that it is unique. Note that property (a) specifies multiplication as a convolution without carry. Thus, (a) can determine the representation only up to an automorphism of the multiplicative structure of \(\mathbb{Z}_{p^2}\). Addition cannot be specified without carry; otherwise, the resulting structure would be of characteristic \(p\). However, \(\mathbb{Z}_{p^2}\) contains a unique additive subgroup of order \(p\) consisting of nonunits. Property (b) forces simple
addition for this subgroup. Property (c) specifies addition between some units and nonunits. Properties (a)–(c) show that the Witt representation offers a very simple (if not the simplest possible) simulation of \( \mathbb{Z}_p^2 \) arithmetic by \( \mathbb{F}_p \) arithmetic. Property (d) ensures that the characteristic is \( p^2 \).

We show that the additive carry in the Witt representation is the function \( g_p : \mathbb{F}_p \times \mathbb{F}_p \to \mathbb{F}_p \) defined for all \( x, y \in \mathbb{F}_p \) by

\[
g_p(x, y) = \left( \frac{x^p + y^p - (x + y)^p}{p} \right) \mod p,
\]

where the arithmetic in parentheses is integer arithmetic. The division by \( p \) in this expression is exact, because the numerator is \( p \)-integral since

\[
x^p + y^p - (x + y)^p = - \sum_{k=1}^{p-1} \binom{p}{k} x^k y^{p-k}
\]

and all the binomial coefficients are divisible by \( p \).

We will later see that the semantics of the Witt representation can be expressed in terms of the function \( f_p : \mathbb{F}_p \to \mathbb{F}_p \) for all \( x \in \mathbb{F}_p \), by

\[
f_p(x) = \left( \frac{x - x^p}{p} \right) \mod p,
\]

where once again the arithmetic in parentheses is integer arithmetic, and the expression in parentheses is \( p \)-integral since \( p | (x - x^p) \) for any integer \( x \). This is just a restatement of Fermat’s “little” theorem.

To prove the correctness of the Witt representation, we need the following lemma, that can easily be proved directly from the definition of \( g_p \).

**Lemma 8.5.** The function \( g_p \) has the following simple properties:

(a) \( g_p(x, 0) = 0 \),
(b) \( g_p(x, -x) = 0 \),
(c) \( g_p(x, y) = g_p(y, x) \),
(d) \( g_p(x, y) + g_p(x + y, z) = g_p(x, y + z) + g_p(x, z) \),
(e) \( g_p(xy, xz) = x \cdot g_p(y, z) \),

for all \( x, y, z \in \mathbb{F}_p \).

We may now prove the correctness of the Witt representation defined in Definition 8.4. The proof of the following theorem develops many useful properties of this representation.

**Theorem 8.6.** (a) Definition 8.4 uniquely determines a representation of the ring \( \mathbb{Z}_p^2 \).

(b) \( [x_0, x_1] + [y_0, y_1] - [x_0 + y_0, x_1 + y_1 + g_p(x_0, y_0)] \).

(c) \( [x_0, x_1] = x_0 + p \cdot (x_1 - f_p(x_0)) \).

(d) \( [x_0, x_1]^k = [x_0^k, kx_0^{k-1} x_1] \) for all positive natural numbers \( k \).
Proof. We shall begin by establishing (b) and (c), from which the uniqueness of the representation follows, and end by establishing existence.

Observe that the representations of 0 and 1 are uniquely determined by \(0 \cdot x = 0\) and \(1 \cdot x = x\) for all \(x \in \mathbb{Z}_{p^2}\). Thus, Definition 8.4(a) implies

\[
0 = [0, 0],
\]

\[
1 = [1, 0].
\]

In addition, the nonunits of \(\mathbb{Z}_{p^2}\) are precisely the set \(\{x \in \mathbb{Z}_{p^2} | x^2 = 0\}\), but, according to Definition 8.4(a), we have \([x_0, x_1]^2 = [x_0^2, 2x_0 x_1]\); so, the nonunits have \(x_0 = 0\) since they are just the set \(\{[x_0, x_1] | x_0^2 = 0, 2x_0 x_1 = 0\}\) and \(0 = [0, 0]\). Thus, we have

\[
\{[0, x] | x \in \mathbb{F}_p\} = \{xp | x \in \mathbb{F}_p\}.
\]

The multiplicative subgroup of \(\mathbb{Z}_{p^2}\) has order \(p(p-1)\) and is cyclic. Consequently, there exist unique subgroups of orders 2, \(p\) and \(p-1\). We may show, by induction on \(k\), from Definition 8.4(a) that

\[
[x_0, x_1]^k = [x_0^k, kx_0^{k-1} x_1^k]
\]

for any positive natural number \(k\) and, therefore, that

\[
[-1, 0] = p^2 - 1
\]

since \(\{-1, 0\}, \{1, 0\}\) represents \(\{x | x^2 = 1\}\). Similarly, it follows that

\[
\{[x, 0] | x \in \mathbb{F}_p^\times\} = \{x | x^{p-1} = 1\}
\]

and

\[
\{[1, x] | x \in \mathbb{F}_p\} = \{x | x^p = 1\} = \{1 + px | x \in \mathbb{F}_p\}.
\]

It follows from (4) and Definition 8.4(a) that

\[
-[x_0, x_1] = [-x_0, -x_1].
\]

It follows directly from Definition 8.4(a) that

\[
[x_0, x_1]^{-1} = [x_0^{-1}, -x_0^{-2} x_1],
\]

as may be verified by multiplying out. (In fact, if \(x_0 \neq 0\) (i.e. \([x_0, x_1]\) is a unit) then \([x_0, x_1] = [1, x_0^{-1} x_1][x_0, 0]\). This is an explicit decomposition giving the internal direct product of the cyclic subgroups of orders \(p-1\) and \(p\).) Now we investigate the consequences of Definition 8.4(d). Evaluating \([0, 1] \cdot ([x_0, 0] + [y_0, 0])\) in two different ways gives

\[
[x_0, 0] + [y_0, 0] = [x_0 + y_0, \gamma(x_0, y_0)]
\]

for some (unknown) function \(\gamma : \mathbb{F}_p \times \mathbb{F}_p \rightarrow \mathbb{F}_p\), because Definition 8.4(a) implies the identity \([0, 1] \cdot [z_0, z_1] = [0, z_0]\).
Using Definition 8.4(b) and (c) gives, from (8),

\[ [x_0, x_1] + [y_0, y_1] = [x_0 + y_0, x_1 + y_1 + \gamma(x_0, y_0)]. \tag{9} \]

Thus, by (2) and (9), we have that \( h_p([x_0, x_1]) = x_0 \) and that \([x_0, 0] = x_0^p \) by (5). In addition, by Definition 8.4(b) and (d), we have \([0, x_1] = px_1 \). Thus,

\[ [x_0, x_1] = [x_0, 0] + [0, x_1] \quad \text{by Definition 8.4(c)} \]

\[ = x_0^p + px_1 \]

\[ = x_0 + p(x_1 - f_p(x_0)) \quad \text{by simple algebraic manipulation}. \]

Thus, we have

\[ [x_0, x_1] = x_0 + p' (x_1 - f_p(x_0)) \tag{10} \]

and, consequently,

\[ [x_0, f_p(x_0)] = x_0. \tag{11} \]

We have now established the uniqueness part of Theorem 8.6 and may confirm that \( \gamma = g_p \) using the Witt representation:

\[ g_p(x_0, y_0) = \frac{1}{p} (x_0^p + y_0^p - (x_0 + y_0)^p) \mod p \]

\[ = \frac{1}{p} ([x_0, 0] + [y_0, 0] - [x_0 + y_0, 0]) \mod p \]

\[ = \frac{1}{p} ([x_0 + y_0, \gamma(x_0, y_0)] - [x_0 + y_0, 0]) \mod p \]

\[ = \frac{1}{p} ([0, \gamma(x_0, y_0)] - [x_0 + y_0, 0]) \mod p \]

\[ = \gamma(x_0, y_0). \]

Finally, we must check that \( \{ [x_0, x_1] | x_0, x_1 \in \mathbb{F}_p \} \) as constructed is a commutative ring: this is checked by trivial computations using Lemma 8.5, Definition 8.4(a) and (1), (2), (7), (7') and (9) above. \( \square \)

9. The prime-field-completeness of \( f_p \)

We now show that the function \( f_p \), shown to be so intimately involved with Witt vectors in the previous section, has great expressive power and, indeed, is prime-field-complete.
Theorem 9.1. (a) \( \pi_p(x, y) = x \cdot f_p(y) + y \cdot f_p(x) - f_p(xy) \).

(b) \( \sigma_p(x, y) = f_p(x) + f_p(y) - f_p(x + y) + g_p(x, y) \).

(c) \( l_p(x) = x \cdot f_p(2) + 2 \cdot f_p \left( \frac{x}{2} \right) - f_p(x) \).

(d) \( g_p(x, 1) = \begin{cases} 0 & \text{if } x = p - 1, \\ f_p(x + 1) - f_p(x) & \text{otherwise}. \end{cases} \)

(e) \( g_p(x, y) = y \cdot (x \cdot y^{p-2} + 1)^{p-1} \cdot (f_p(x \cdot y^{p-2} + 1) - f_p(x \cdot y^{p-2})) \).

Proof. (a) Translation into the Witt representation gives

\[
\langle x_0, x_1 \rangle \cdot \langle y_0, y_1 \rangle = [x_0, x_1 \cdot f_p(x_0)] \cdot [y_0, y_1 \cdot f_p(y_0)].
\]

Multiplying out according to the rules of the two representations gives

\[
\langle x_0 y_0, x_0 y_1 + x_1 y_0 + \pi_p(x_0, y_0) \rangle = [x_0 y_0, x_0 y_1 + y_0 x_1 + y_0 f_p(x_0) + x_0 f_p(y_0)].
\]

Finally, translation of the left-hand side into the Witt representation gives

\[
[x_0 y_0, x_0 y_1 + x_1 y_0 + \pi_p(x_0, y_0) + f_p(x_0 y_0)].
\]

Thus, \( \pi_p(x_0, y_0) + f_p(x_0 y_0) = y_0 f_p(x_0) + x_0 f_p(y_0) \), which gives the stated identity.

(b) As (a), except using \( \langle x_0, x_1 \rangle + \langle y_0, y_1 \rangle \).

(c) Follows from (a) and Theorem 8.3, i.e. \( l_p(x) = \pi_p(2, x/2) \).

(d) Follows immediately from the definitions of \( f_p \) and \( g_p \).

(e) Lemma 8.5(e) asserts that \( g_p(xz, yz) = z \cdot g_p(x, y) \). Thus, since \( y^{p-1} - 1 \) if \( y \) is nonzero, \( g_p(x \cdot y^{p-2}, 1) = y^{p-2} \cdot g_p(x, y) \) in this case. If \( y \) is zero then \( g_p(x, y) \) is also zero and, so, the same identity holds. The result then follows from (d).

Of course, \( f_p \) has polynomial-size Boolean circuits in the standard representation of \( \mathbb{F}_p \), as described earlier.

In contrast to \( f_p \), \( \pi_p \) and \( \sigma_p \), the function \( g_p \) is not known to be prime-field-complete. On the other hand, neither is \( g_p \) known to have polynomial-size arithmetic circuits. Furthermore, \( g_p \) has an interesting form related to the integral of a geometric series. If we define a polynomial \( g_p(x) \) of degree less than \( p \) by \( g_p(x) = g_p(-x, 1) \), then it is easy to see that \( g_p(x) = \frac{1}{x^p - 1} \cdot \frac{1}{x^{p-1} - 1} \cdot \frac{1}{x - 1} \). This polynomial has the derivative \( g'_p(x) = \sum_{k=0}^{p-2} x^k - (1 - x^{p-1})/(1-x) \), which gives \( g'_p(x) \) a polynomial-size circuit. The status of \( g_p(x) \) is unresolved at the time of writing. (Note that \( g_p(x, y) \) easily reduces to \( g_p(x) \) using similar techniques to the proof of Theorem 9.1(e).)
10. Reduction of \( f_p \) to \( g_p \) and \( m_p \)

We now give a reduction of \( f_p \) to \( g_p \) and \( m_p \) together and vice versa. (Recall that \( m_p : \mathbb{Z}_{p^2} \rightarrow \mathbb{Z}_{p^2} \) is the "concrete modulo-\( p \)" function.) We also determine the coefficients of the unique polynomial of degree less than \( p \) that defines the function \( f_p \). Hereafter, we take \( f_p \) to be that polynomial, i.e. \( f_p(x) \in \mathbb{F}_p[x] \), \( \deg(f_p(x)) \leq p \) and \( \forall a \in \mathbb{F}_p, \ f_p(a) = ((a - a^p)/p) \mod p \).

As before, we regard \( \mathbb{F}_p \) and \( \mathbb{Z}_{p^2} \) concretely as the integers \( \{0, 1, \ldots, p-1\} \) and \( \{0, 1, \ldots, p^2-1\} \), respectively, so that \( \mathbb{F}_p \subseteq \mathbb{Z}_{p^2} \). Thus, \( f_p(x) \in \mathbb{Z}_{p^2}[x] \) as well. This will be important in what follows.

**Theorem 10.1.** In \( \mathbb{Z}_{p^2} \) we have \( m_p(x) = x^p + p \cdot f_p(x) \), where we regard \( f_p(x) \in \mathbb{Z}_{p^2}[x] \) as above.

**Proof.** By definition, regarding \( \mathbb{F}_p \subseteq \mathbb{Z}_{p^2} \) and taking the Witt representation of \( x \) as \( x = [x_0, x_1] \), we have, following Theorem 8.6(c),

\[
m_p(x) = m_p([x_0, x_1]) = [x_0, f_p(x_0)].
\]

Now, decompose this using Definition 8.4(c):

\[
m_p(x) = [x_0, 0] + [0, f_p(x_0)].
\]

But Theorem 8.6(d), with \( k = p \), gives \([x_0, x_1]^p = [x_0, 0]\), and Definition 8.4(a) gives that \([0, f_p(x_0)] = [0, 1] \cdot [f_p(x_0), a]\) for any \( a \in \mathbb{F}_p \). Furthermore, Definition 8.4(d) asserts that \( [0, 1] = p \). Putting all this together gives

\[
m_p(x) = x^p + p \cdot [f_p(x_0), a].
\]

Now, because \( f_p \) is defined to be a polynomial, for \( x \in \mathbb{Z}_{p^2} \) we have the following identity: \( f_p(x) = f_p([x_0, x_1]) = [f_p(x_0), a] \) for some \( a \in \mathbb{F}_p \). This follows from Definition 8.4(a) and Theorem 8.6(b) and (c), since \( f_p([x_0, x_1]) \) may be computed by multiplication, addition, and the use of constants, respectively. \( \square \)

**Theorem 10.2.** (i) If \( f_p \) has a polynomial-size \( \mathbb{F}_p \)-arithmetic circuit then \( m_p \) has a polynomial-size \( \mathbb{Z}_{p^2} \)-arithmetic circuit, and \( g_p \) has a polynomial-size \( \mathbb{F}_p \)-arithmetic circuit.

(ii) If \( m_p \) has a polynomial-size \( \mathbb{Z}_{p^2} \)-arithmetic circuit, and \( g_p \) has a polynomial-size \( \mathbb{F}_p \)-arithmetic circuit, then \( f_p \) has a polynomial-size \( \mathbb{F}_p \)-arithmetic circuit.

**Proof.** (i) Assume that \( c_p \) is a polynomial-size \( \mathbb{F}_p \)-arithmetic circuit for \( f_p \). Now regard \( c_p \) as a circuit over \( \mathbb{Z}_{p^2} \); add one final operation to multiply the output of \( c_p \) by \( p \), and call the resulting circuit \( c'_p \). Note that \( c'_p \) computes the function \( p \cdot f_p(x) \) that is well-defined because it is equal to \( p \cdot f_p(m_p(x)) \). (The factor of \( p \) ensures that only the modulo \( p \) part of \( f_p(x) \) affects the output, and because \( f_p \) is now defined to be a polynomial (indeed it is computed with arithmetic), for \( x \in \mathbb{Z}_{p^2} \) we have
\( f_p(x) = f_p(x + a \cdot p) \mod p \) for any \( a \). Now, using Theorem (10.1), \( m_p(x) = x^p + p \cdot f_p(x) \) may be easily computed by a polynomial-size \( \mathbb{Z}_p \)-arithmetic circuit that uses \( c'_p \). The other part follows from Theorem 9.1(c).

(ii) Assume that \( c_p \) is a polynomial-size \( \mathbb{Z}_p \)-arithmetic circuit computing \( m_p \). We now simulate this circuit in the Witt representation, creating a new circuit \( c'_p \) that takes input \( (x_0, x_1) \in \mathbb{F}_p^2 \), with \( x = [x_0, x_1] \) the original input. The operations of \( c_p \) are simulated by circuits over \( \mathbb{F}_p \), using the identities given in Definition 8.4(a) for multiplication, and in Theorem 8.6(b) for addition. The latter requires that we use an \( \mathbb{F}_p \)-circuit for \( g_p \), and we may assume we have such a polynomial-size circuit. The output of \( c'_p \) is a pair \( (y_0, y_1) \) representing \( m_p(x) \). Thus, by Theorem 8.6(c),

\[
(y_0, y_1) = m_p([x_0, x_1]) = [x_0, f_p(x_0)],
\]

so that \( y_1 = f_p(x_0) \), and \( c'_p \) can be used to compute \( f_p \). \( \square \)

11. The relationship to the Bernoulli polynomials

We now relate \( m_p \) to the Bernoulli polynomials, but first we summarize some properties of Bernoulli numbers and polynomials (see e.g. [6, 8]).

The Bernoulli numbers \( \{B_k\} \) and the Bernoulli polynomials \( \{B_k(x)\} \) are, respectively, rational numbers and polynomials over the rational numbers defined by

\[
\frac{t}{e^t - 1} = \sum_{k=0}^{\infty} B_k \frac{t^k}{k!} \quad \text{and} \quad \frac{t \cdot e^{tx}}{e^t - 1} = \sum_{k=0}^{\infty} B_k(x) \frac{t^k}{k!}.
\]

(B1) \( B_0 = 1, B_1 = -\frac{1}{2}, B_{2k+1} = 0 \) for \( k \geq 1 \).

For (B2) and (B3), suppose that \( x \) is a residue class of the integers modulo \( p-1 \). Let \( k, \ell, a \in \mathbb{Z} \), and \( k, \ell > 0 \).

(B2) Kummer's congruence. If \( 0 \notin a \) then \( B_k \) and \( B_l \) are \( p \)-integral and

\[
\left( \begin{array}{c} B_k \\ k \end{array} \right) \equiv \left( \begin{array}{c} B_l \\ l \end{array} \right) \mod p.
\]

(B3) Von Staudt's congruence. If \( 0 \notin \ell \) then \( (pB_k) \equiv -1 \mod p \).

(B4) \( B_k(x) = \sum_{i=0}^{k} \binom{k}{i} B_{k-i} x^i \).

(B5) \( B_k(x+1) - B_k(x) = k \cdot x^{k-1} \).

(B6) \( B_k(x+1) = (-1)^k B_k(-x) \).

Property (B2) ensures that \( B_1, \ldots, B_{p-2} \) and \( B_p \) are defined in \( \mathbb{F}_p \) and \( \mathbb{Z}_p \). In fact, \( B_p = 0 \) by property (B1). Property (B3) ensures that \( (p \cdot B_{p-1}) \) is defined in \( \mathbb{F}_p \) and \( \mathbb{Z}_p \). Thus, property (B4) ensures that \( B_k(x) \) is defined in \( \mathbb{F}_p \) and \( \mathbb{Z}_p \) when \( k < p-1 \) and when \( k = p \). The key property is (B5), which, when summed from \( 0 \) to \( x \), gives

\[
B_k(x+1) - B_k(0) = k \cdot \sum_{k=1}^{x} N^{k-1}.
\]
Theorem 11.1. (i) $m_p(x) = B_p(x + 1) + (1 - p)x$.

(ii) $m_p(x) = \sum_{k=1}^{p} a_k x^k$, where

$$a_k = \begin{cases} 1 & \text{if } k = p, \\ p \cdot B_{p-1} + (1 - p) = p \cdot \sum_{i=1}^{p-2} \binom{B_i}{i} & \text{if } k = 1, \\ p \cdot \frac{B_{p-k}}{k} & \text{otherwise}. \end{cases}$$

Proof. (i) By (B5), $B_p(x + 1) = p \cdot \sum_{i=0}^{p} k^{p-1}$ in $\mathbb{Z}_p$, since $B_p(0) = 0$ by (B4) and (B2). Let $x = (x_0, x_1) \in \mathbb{Z}_p^2$; then we have $\sum_{k=0}^{p} k^{p-1} \equiv x_0 - x_1 \mod p$ since units contribute 1 to the sum and nonunits zero. Thus, since $x_0 = m_p(x)$,

$$p \cdot x + m_p(x) - x = \langle 0, x_0 \rangle + \langle x_0, 0 \rangle - \langle x_0, x_1 \rangle = \langle 0, x_0 \rangle + \langle 0, -x_1 \rangle = \langle 0, x_0 - x_1 \rangle = p \cdot (x_0 - x_1) = B_p(x + 1),$$

from which the result follows.

(ii) This follows from Theorem 11.1(i) by substituting (B4) with $k = p$ and using the identity $\binom{p}{i} = \binom{p-1}{i-1}$ for $i$ not 0 or $p$ and the identity $p \cdot (p^{i-1}) = p \cdot (-1)^i$ in $\mathbb{Z}_p$. The use of (B6), together with the observations that $B_p = 0, B_0 = 1$ and $(p \cdot B_{p-1})$ is well-defined, is sufficient for all but the alternative form for $a_1$ which follows from Theorem 11.1(i) with $x = 1$ and $m_p(1) = 1$ and (B6) with $x = 1$. □

Theorem 11.2. $f_p(x) = \sum_{k=1}^{p} b_k x^k$, where

$$b_k = \begin{cases} \frac{(p \cdot B_{p-1} + 1)}{p} & \text{if } k = 1, \\ \frac{B_{p-k}}{k} & \text{otherwise}. \end{cases}$$

Proof. Follows immediately from Theorem 10.1, which relates $f_p$ to $m_p$, and Theorem 11.1(ii), which gives the coefficients of $m_p$. Note that, by (B3), $b_1$ is well-defined. □

12. Conclusion and open problems

The precise power of finite-field arithmetic as compared to Boolean operations depends on the arithmetic complexity of $f_p$ and, thus, on the complexities of $m_p$ and $g_p$. 
These complexities are as yet unknown. The status of $g_p$ is particularly interesting: should $g_p$ prove to have polynomial-size arithmetic circuits, then the Witt representation of $\mathbb{Z}_p^2$ can perform arithmetic with polynomial-size arithmetic circuits, and yet entering and leaving that representation uses $f_p$ and is, therefore, prime-field-complete.

References


