Research Article

Shizuo Kaji, Toshiaki Maeno, Koji Nuida* and Yasuhide Numata

Polynomial expressions of *p*-ary auction functions

https://doi.org/10.1515/jmc-2018-0016
Received April 5, 2018; accepted March 25, 2019

Abstract: One of the common ways to design secure multi-party computation is twofold: to realize secure fundamental operations and to decompose a target function to be securely computed into them. In the setting of fully homomorphic encryption, as well as some kinds of secret sharing, the fundamental operations are additions and multiplications in the base field such as the field $\mathbb{F}_2$ with two elements. Then the second decomposition part, which we study in this paper, is (in theory) equivalent to expressing the target function as a polynomial. It is known that any function over the finite prime field $\mathbb{F}_p$ has a unique polynomial expression of degree at most $p - 1$ with respect to each input variable; however, there has been little study done concerning such minimal-degree polynomial expressions for practical functions. This paper aims at triggering intensive studies on this subject, by focusing on polynomial expressions of some auction-related functions such as the maximum/minimum and the index of the maximum/minimum value among input values.

Keywords: Secure multi-party computation, polynomial expression of functions, finite fields

MSC 2010: 94A60, 68R05, 12Y05

1 Introduction

Secure multi-party computation (or simply *secure computation*) is a cryptographic technology that enables two or more parties to jointly compute some function value(s) from their local inputs in a way that, during a computation, each party can know the party's local output value but cannot learn anything about the other parties' local inputs/outputs. Among several existing frameworks to realize secure computation, some of the major directions in this area are those based on fully homomorphic encryption (FHE) [5] and on secret sharing (SS) [8]. In FHE-based secure computation (e.g., [2, 4]), the primitive data type is usually the binary field $\mathbb{F}_2$, and a target function to be securely computed has to be implemented by combining the addition and multiplication in $\mathbb{F}_2$ (each being equivalent to bit operations XOR and AND, respectively). On the other hand, in SS-based secure computation (e.g., [1]), a major primitive data type is again $\mathbb{F}_2$ and a target function has also to be implemented by combining the addition and multiplication. We note that there is also an FHE scheme that can directly handle the finite prime field $\mathbb{F}_p$ for small prime $p > 2$ as well [7]; and the basic idea of

***Corresponding author: Koji Nuida,** Graduate School of Information Science and Technology, The University of Tokyo, Tokyo; and National Institute of Advanced Industrial Science and Technology (AIST), Japan, e-mail: nuida@mist.i.u-tokyo.ac.jp. https://orcid.org/0000-0001-8259-9958
Shizuo Kaji, Institute of Mathematics for Industry, Kyushu University, Fukuoka; and Japan Science and Technology Agency (JST) PRESTO Researcher, Japan, e-mail: skaji@imi.kyushu-u.ac.jp. https://orcid.org/0000-0002-7856-6536
Toshiaki Maeno, Meijo University, Nagoya, Japan, e-mail: tmaeno@meijo-u.ac.jp
Yasuhide Numata, Department of Mathematics, Shinshu University, Matsumoto, Nagano, Japan, e-mail: nu@math.shinshu-u.ac.jp. https://orcid.org/0000-0002-1228-7067

SS-based secure computation can be extended straightforwardly to the primitive data type $\mathbb{F}_p$ instead of $\mathbb{F}_2$. Regarding this, in this paper we treat $\mathbb{F}_p$ for a general prime p (not just $\mathbb{F}_2$) as the base field of the argument.

In the two frameworks for secure computation mentioned above, a target function is supposed to be decomposed into a combination of addition and multiplication in the field $\mathbb{F}_p$; this is (in theory) equivalent to expressing the function as a polynomial over the field $\mathbb{F}_p$. Moreover, the multiplication in $\mathbb{F}_p$ is significantly more expensive than the addition in $\mathbb{F}_p$ when realized as secure computation. Indeed, in the FHE-based framework, the multiplication increases the "noise" of the ciphertexts (to be cancelled later by an inefficient "bootstrapping" procedure) much more rapidly than the addition; while in the SS-based framework, the multiplication requires communication between the parties, in contrast to the addition which can be done by local computation at each party only. Hence, we may naively expect that an expression of the target function as a low degree polynomial would involve less multiplications, and thus yield efficient secure computation for the function. On the other hand, it is known that *any* function over the prime field $\mathbb{F}_p$ can be expressed, in a unique manner, as a polynomial over $\mathbb{F}_p$ having degree at most $p - 1$ with respect to each input variable. We refer to such a polynomial as the *minimal polynomial* of the function.

However, such minimal polynomial expressions for practical functions were not studied well in the literature. One of the aims of this paper is to trigger intensive studies on this subject. We emphasize that, though the *existence* of the minimal polynomial expression of any function over $\mathbb{F}_p$ is theoretically guaranteed, it is still a non-trivial task to *concretely compute* the minimal polynomial expression. Among the rare studies of the minimal polynomials in the literature, the most successful and theoretically interesting result to the authors' best knowledge is the one by Sturtivant and Frandsen [9, Theorems 9.1 (a) and 11.2]; they showed that the carry function in multiplication of p-ary integers has a polynomial expression consisting of significantly fewer monomials, which uses number-theoretic objects such as the Bernoulli numbers and Wilson's quotient. (See also [6] for a different approach to the result and also for an expression of the carry function in the case of addition of p-ary integers.)

In this paper, we study minimal polynomial expressions of a certain kind of functions specified below. These functions are expected to be useful in some practical procedures such as auction and voting; here we refer to those functions as "auction functions". The types of auction functions considered in this paper and our results are summarized as follows.

In Section 3, we deal with the maximum function $\max(x)$ for inputs $x = (x_0, x_1, \dots, x_{n-1}) \in (\mathbb{F}_p)^n$. We provide a general (but less concrete) formula for the minimal polynomial for $\max(x)$ for any prime p, and also give more concrete minimal polynomial expressions of $\max(x)$ for $p = 2, 3$. Similar results are also given for the minimum function $\min(x)$.

In Section 4, we deal with the function $\operatorname{argmax}(x)$ that returns the least index i satisfying $x_i = \max(x)$. More precisely, to handle the integer-valued function $\operatorname{argmax}(x)$, we consider each, say r-th, digit $\operatorname{argmax}^{(r)}(x)$ of the p-ary expression of the value $\operatorname{argmax}(x)$. We provide a general formula for the minimal polynomial for $\operatorname{argmax}^{(r)}(x)$ for any $r \geq 0$ and any prime p, and also write down the formula for the special cases $p = 2, 3$. Similar results are also given for the function $\operatorname{argmin}(x)$ that returns the least index i satisfying $x_i = \min(x)$.

In Section 5, we focus on the case of two inputs, $x = (x_0, x_1)$, and provide minimal polynomial expressions of $\operatorname{argmax}(x_0, x_1) = \operatorname{argmax}^{(0)}(x_0, x_1)$ and $\max(x_0, x_1)$ for any p. (We note that, only the cases for small p such as $p = 2, 3$ for the function $\max(x)$ are discussed in Section 3.)

In Section 6, we briefly study two other functions $\operatorname{ismax}(y; x)$ and $\operatorname{nummax}(x)$, where $\operatorname{ismax}(y; x)$ for $y \in \mathbb{F}_p$ and $x = (x_0, \dots, x_{n-1}) \in (\mathbb{F}_p)^n$ returns 1 if $y = \max(x)$ and returns 0 otherwise, and $\operatorname{nummax}(x)$ returns the number of indices i satisfying $x_i = \max(x)$. We also discuss the cases for small p such as $p = 2, 3$ in slightly more detail.

Finally, in Section 7, we discuss about the possible extension of our results on the auction functions with *single-digit* inputs $x_i \in \mathbb{F}_p$ to the case of *multi-digit* inputs such as $x_i = (x_{i,0}, x_{i,1}, \dots, x_{i,\ell-1}) \in (\mathbb{F}_p)^\ell$ which is regarded as an integer $x_{i,0} + x_{i,1}p + \cdots + x_{i,\ell-1}p^{\ell-1}$. While leaving most of the cases as future research topics, in Section 7 we provide as an example a general formula for the multi-digit version of the function $\operatorname{argmax}(x)$ and write down the minimal polynomial expression for the smallest case $p = \ell = 2$. We also give the minimal polynomial expression for the multi-digit version of the function $\operatorname{ismax}(y; x)$ for the smallest case $p = \ell = 2$ as well.

2 Notation and Basic functions

In this section, we fix some notations used throughout the paper. Let p be a prime. A vector $(x_0, x_1, \dots, x_{n-1})$ of length n over the field $\mathbb{F}_p$ is often denoted by x. We introduce a linear ordering $<$ on $\mathbb{F}_p$ by naturally identifying $\mathbb{F}_p$ with the (naturally ordered) subset $\{0, 1, \dots, p-1\}$ of $\mathbb{Z}$. We define an involution on $\mathbb{F}_p$ by $\bar{x} = p - 1 - x$ for $x \in \mathbb{F}_p$, and extend it coordinate-wisely on $(\mathbb{F}_p)^n$. We denote by $e_i(x)$ the i-th elementary symmetric polynomial of $x_0, x_1, \dots, x_{n-1}$ so that $\prod_{i=0}^{n-1}(1 + x_i) = \sum_{i=0}^{n} e_i(x)$. For an integer $k \geq 0$, its r-th digit in the p-ary expansion is denoted by $k^{(r)}$; that is, $k = \sum_{r=0}^{\infty} k^{(r)} p^r$ with $k^{(r)} \in \{0, 1, \dots, p-1\}$ for each r.

Given any logical proposition $P(x)$ for an object x, we define its *truth function* by

$$\chi_P(x) = \begin{cases} 1 & (P(x) \text{ is true}), \\ 0 & (\text{otherwise}), \end{cases}$$

which is often abbreviated as $\chi_P(x) = \chi(P)$. We frequently use the same symbol for a function and its polynomial expression; see below for some examples.

Example 2.1. For $t \in \mathbb{F}_p$ and a variable x, Fermat's little theorem implies that the minimal polynomial for the *delta function* $\delta_t(x) = \chi(x = t)$ is given by

$$\delta_t(x) = 1 - (x - t)^{p-1} = -\prod_{i=1}^{p-1}(x - t + i)$$

(the last equality follows from Wilson's theorem: $(p-1)! \equiv -1 \pmod{p}$). Similarly, the minimal polynomial for the *low-pass function* $L_t(x) = \chi(x < t)$ is given by

$$L_t(x) = \sum_{0 \leq k < t} \delta_k(x) = \sum_{0 \leq k < t} \left(1 - (x - k)^{p-1}\right).$$

For notational convenience, we extend the definition of the low-pass function $L_t(x)$ to the case $t = p$, by setting $L_p(x) = 1$ (note that the relation $x < p$ *as integers* holds for any $x \in \mathbb{F}_p$).

3 Polynomial expressions of the max and the min functions

For a vector $x = (x_0, x_1, \dots, x_{n-1}) \in (\mathbb{F}_p)^n$, let $\max(x)$ (respectively, $\min(x)$) denote the maximum (respectively, minimum) among the n values $x_0, x_1, \dots, x_{n-1}$.

First, we note that for each $a \in \mathbb{F}_p$, the condition $a \geq t$ is satisfied for precisely a of the $p - 1$ values $t = 1, \dots, p-1$. Based on this fact and using the functions in Example 2.1, we obtain the minimal polynomial of $\max(x)$ as follows.

Proposition 3.1. *The minimal polynomial of* max *is given by*

$$\max(x) = \sum_{1 \leq t \leq p-1} \chi(x_i \geq t \textit{ for some } i) = \sum_{1 \leq t \leq p-1} \left(1 - \prod_{i=0}^{n-1} L_t(x_i)\right).$$

In particular, when $p = 2$ this simplifies (by noticing $L_1(x_i) = 1 + x_i$):

Corollary 3.2. *When $p = 2$, the minimal polynomial of* $\max(x)$ *is given by*

$$\max(x) = \prod_{i=0}^{n-1}(1 + x_i) - 1 = \sum_{i=1}^{n} e_i(x).$$

However, when $p > 2$, the general expression in Proposition 3.1 consists of a lot of terms. We compute a more concise expression for $p = 3$ later.

On the other hand, we note that $\max(x) + 1 = 0$ if $x_i = p - 1$ for some x_i. This implies that the minimal polynomial of $\max(x) + 1$ has $1 + x_i$ as a factor for every i. Therefore, we have

$$\max(x) = f_n(x) \prod_{i=0}^{n-1} (1 + x_i) - 1 = f_n(x) \sum_{i=0}^{n} e_i(x) - 1$$

for some polynomial $f_n(x)$ in which each variable x_i has degree at most $p - 2$. In particular, this observation yields another proof of Corollary 3.2 (where $p = 2$).

Now we give the following result for the case $p = 3$.

Proposition 3.3. *When $p = 3$, a minimal polynomial expression for* $\max(x)$ *is given by*

$$\max(x) = \sum_{i=0}^{\lfloor n/2 \rfloor} e_{2i}(x) \sum_{i=0}^{n} e_i(x) - 1.$$

Proof. Write the right-hand side as $P(x)$. As the minimality condition on the degree is satisfied for $P(x)$, it suffices to verify $\max(x) = P(x)$ for any $x \in (\mathbb{F}_p)^n$. First note that $\prod_{i=0}^{n-1}(1 - x_i) = \sum_{i=0}^{n} e_i(-x) = \sum_{i=0}^{n} (-1)^i e_i(x)$, where we write $-x = (-x_0, -x_1, \ldots, -x_{n-1})$, therefore $\prod_{i=0}^{n-1}(1 + x_i) + \prod_{i=0}^{n-1}(1 - x_i) = 2 \sum_{i=0}^{\lfloor n/2 \rfloor} e_{2i}(x)$. This implies (since $2^{-1} = 2$ in $\mathbb{F}_3$)

$$P(x) = \sum_{i=0}^{\lfloor n/2 \rfloor} e_{2i}(x) \sum_{i=0}^{n} e_i(x) - 1 = 2\left(\prod_{i=0}^{n-1} (1 + x_i)^2 + \prod_{i=0}^{n-1} (1 - x_i^2) \right) - 1.$$

When $\max(x) = 2$, there exists an index i with $x_i = 2$, and we have $1 + x_i = 0$ and $1 - x_i^2 = 0$ for such i. This implies that $P(x) = -1 = 2$ in this case. When $\max(x) = 1$, we have $\prod_{i=0}^{n-1}(1 + x_i)^2 = 1$ as $x_i \in \{0, 1\}$ for every i, and $\prod_{i=0}^{n-1}(1 - x_i^2) = 0$ as $x_i = 1$ for at least one i. This implies that $P(x) = 2 - 1 = 1$ in this case. Finally, when $\max(x) = 0$, we have $\prod_{i=0}^{n-1}(1 + x_i)^2 = 1$ and $\prod_{i=0}^{n-1}(1 - x_i^2) = 1$ as $x_i = 0$ for every i. This implies that $P(x) = 2 \cdot 2 - 1 = 0$ in this case. Hence, the claim holds. □

To obtain a minimal polynomial expression for $\min(x)$, we exploit the following duality between the functions max and min: $\overline{\min(x)} = \max(\bar{x})$ for any $x \in (\mathbb{F}_p)^n$. Thus, a minimal polynomial expression for max converts to that of min and vice versa. For example, Corollary 3.2 and Proposition 3.3 imply the following.

Corollary 3.4. *When $p = 2$, a minimal polynomial expression for* $\min(x)$ *is given by*

$$\min(x) = \prod_{i=0}^{n-1} x_i = e_n(x).$$

When $p = 3$, a minimal polynomial expression for $\min(x)$ *is given by*

$$\min(x) = \prod_{i=0}^{n-1} x_i^2 + \prod_{i=0}^{n-1} x_i(1 - x_i) = e_n\left(1 + \sum_{i=1}^{n} (-1)^i e_i + e_n\right).$$

For the next case $p = 5$, minimal polynomial expressions of $\max(x)$ for small values of n in terms of elementary symmetric polynomials can be determined by direct calculation:

Example 3.5. When $p = 5$, the following are minimal polynomial expressions:

$$\begin{aligned}
\max(x_0, x_1) &= (1 + e_1 + e_2)(1 + 2e_1^2 e_2 + 4e_1 e_2 + e_2) - 1, \\
\max(x_0, x_1, x_2) &= (1 + e_1 + e_2 + e_3)(1 + 2e_1^2 e_2 + e_1 e_2 e_3 + 2e_1 e_3^2 + e_2^2 e_3 \\
&\qquad + 2e_2 e_3^2 + 4e_1 e_2 + 3e_1 e_3 + e_2 e_3 + 3e_3^2 + e_2) - 1.
\end{aligned}$$

However, it seems to be difficult to obtain a general formula (such as Proposition 3.3) for $p \geq 5$. The function $\max(x)$ with $n = 2$ for any p will be revisited in Section 4.

Remark 3.6. The function $\max(x)$ is a symmetric function (in the variables $x_0, x_1, \ldots, x_{n-1}$), and satisfies $\max(x, 0) = \max(x)$ and an "associativity" in the following sense:

$$\max(x_0, x_1, \ldots, x_{n-1}, x_n) = \max(\max(x_0, \ldots, x_{n-1}), x_n) = \max(x_0, \max(x_1, \ldots, x_n)).$$

By using this property recursively, a minimal polynomial expression of the function max with two variables (i.e., for $n = 2$) yields a polynomial expression of the function max with any number of variables (i.e., for any n). However, the polynomial thus obtained is *not* the minimal polynomial for $\max(x)$ in general.

4 Polynomial expressions of the argmax function

Let $\operatorname{argmax}(x)$ be the least integer $i \geq 0$ with $x_i = \max(x)$. Note that $\operatorname{argmax}(x)$ takes a value in $\{0, 1, \dots, n-1\}$; to handle this function as a function over $\mathbb{F}_p$, we define, for $r \geq 0$,

$$\operatorname{argmax}^{(r)} \colon (\mathbb{F}_p)^n \to \mathbb{F}_p, \quad \operatorname{argmax}^{(r)}(x) = \operatorname{argmax}(x)^{(r)},$$

where $\operatorname{argmax}(x)^{(r)}$ is the r-th digit in the p-ary expansion of $\operatorname{argmax}(x)$.

We note that $\operatorname{argmax}(x)$ is equal to the number of integers $0 \leq i \leq n-2$ such that $x_j < \max(x)$ for every $0 \leq j \leq i$; indeed, the latter condition is satisfied if and only if $0 \leq i \leq \operatorname{argmax}(x) - 1$. This implies (since $\operatorname{argmax}(x) = 0$ if $\max(x) = 0$) that

$$\operatorname{argmax}(x) = \sum_{t=1}^{p-1}\Big(\chi(\max(x) = t)\sum_{i=0}^{n-2}\prod_{j\leq i}\chi(x_j < t)\Big) = \sum_{t=1}^{p-1}\sum_{i=0}^{n-2}\chi(\max(x) = t)\prod_{j\leq i}\chi(x_j < t),$$

which is considered as an integer rather than an element of $\mathbb{F}_p$. We note moreover that, conditioned on the case where $x_j < t$ for every $j \leq i$, we have $\max(x) = t$ if and only if $x_j < t + 1$ for every $j > i$ and $x_j = t$ for some $j > i$. Now the equality above implies

$$\operatorname{argmax}(x) = \sum_{t=1}^{p-1}\sum_{i=0}^{n-2}\Big(\prod_{j\leq i}\chi(x_j < t)\prod_{j>i}\chi(x_j < t+1) - \prod_{j=0}^{n-1}\chi(x_j < t)\Big), \tag{4.1}$$

again as an integer rather than an element of $\mathbb{F}_p$. Considering the right-hand side of (4.1) in $\mathbb{F}_p$ yields the minimal polynomial of $\operatorname{argmax}^{(0)}(x)$. Now we also have the following fact implied directly by the definition of $\operatorname{argmax}^{(r)}(x)$:

$$\operatorname{argmax}^{(r)}(x) = \operatorname{argmax}^{(0)}(\max(x_0, x_1, \dots, x_{p^r-1}), \dots, \max(x_{i\cdot p^r}, x_{i\cdot p^r+1}, \dots, x_{(i+1)\cdot p^r-1}), \dots), \tag{4.2}$$

where, in the right-hand side, the tuple $x = (x_0, \dots, x_{n-1})$ is divided into blocks of p^r consecutive components (the last block may consist of less than p^r elements). Let $S(r, n)$ be the set of indices for the last elements of all but the last blocks in x, namely,

$$S(r, n) = \{h \cdot p^r - 1 \mid 1 \leq h \leq \lfloor (n-1)/p^r \rfloor\}.$$

Then, by combining (4.2) with (4.1), we have (in $\mathbb{F}_p$)

$$\operatorname{argmax}^{(r)}(x) = \sum_{t=1}^{p-1}\sum_{i\in S(r,n)}\Big(\prod_{j\leq i}\chi(x_j < t)\prod_{j>i}\chi(x_j < t+1) - \prod_{j=0}^{n-1}\chi(x_j < t)\Big)$$

where we used the fact (for any h, h', and s) that $\max(x_h, x_{h+1}, \dots, x_{h'}) < s$ if and only if $x_j < s$ for every $h \leq j \leq h'$. Since $|S(r, n)| \equiv (n-1)^{(r)} \pmod{p}$, the equality above can be rewritten as

$$\operatorname{argmax}^{(r)}(x) = \sum_{t=1}^{p-1}\Big(\sum_{i\in S(r,n)}\prod_{j\leq i}\chi(x_j < t)\prod_{j>i}\chi(x_j < t+1) - (n-1)^{(r)}\cdot\prod_{j=0}^{n-1}\chi(x_j < t)\Big).$$

Hence, we have the following result.

Proposition 4.1. *Let* $S(r, n) = \{h \cdot p^r - 1 \mid 1 \leq h \leq \lfloor (n-1)/p^r \rfloor\}$. *Then for* $x = (x_0, \dots, x_{n-1})$, *the minimal polynomial for* $\operatorname{argmax}^{(r)}(x)$ *is given by*

$$\operatorname{argmax}^{(r)}(x) = \sum_{t=1}^{p-1}\Big(\sum_{i\in S(r,n)}\prod_{j=0}^{i}L_t(x_j)\prod_{k=i+1}^{n-1}L_{t+1}(x_k) - (n-1)^{(r)}\cdot\prod_{j=0}^{n-1}L_t(x_j)\Big).$$

Remark 4.2. Let argmin(x) be the function that returns the least index i with $\min(x) = x_i$. A minimal polynomial expression of the function argmin is obtained from that of the function argmax via the duality $\operatorname{argmin}(x) = \operatorname{argmax}(\bar{x})$, similar to the case of the function min discussed in Section 3.

Below we write down the general formula in Proposition 4.1 for the case $p \in \{2, 3\}$.

Example 4.3. We consider the case $p = 2$. Now the set $S(r, n)$ is $S(r, n) = \{h \cdot 2^r - 1 \mid 1 \le h \le \lfloor (n-1)/2^r \rfloor\}$, and the relations $L_1(x_j) = 1 + x_j$ and $L_2(x_j) = 1$ hold. Then Proposition 4.1 implies (since the characteristic of $\mathbb{F}_p$ is now 2)

$$\operatorname{argmax}^{(r)}(x) = \sum_{i \in S(r,n)} \prod_{j=0}^{i} L_1(x_j) + \chi((n-1)^{(r)} = 1) \cdot \prod_{j=0}^{n-1} L_1(x_j).$$

Moreover, by setting $S'(r, n) = S(r, n)$ if $(n-1)^{(r)} = 0$ and $S'(r, n) = S(r, n) \cup \{n-1\}$ if $(n-1)^{(r)} = 1$, we have the following minimal polynomial expression of $\operatorname{argmax}^{(r)}$:

$$\operatorname{argmax}^{(r)}(x_0, x_1, \ldots, x_{n-1}) = \sum_{i \in S'(r,n)} (1 + x_0)(1 + x_1) \cdots (1 + x_i).$$

Example 4.4. We consider the case $p = 3$. Now the set $S(r, n)$ is $S(r, n) = \{h \cdot 3^r - 1 \mid 1 \le h \le \lfloor (n-1)/3^r \rfloor\}$, and the relations $L_1(x_j) = 1 - x_j^2$, $L_2(x_j) = (1 + x_j)^2$ and $L_3(x_j) = 1$ hold. Then Proposition 4.1 yields the following minimal polynomial expression of $\operatorname{argmax}^{(r)}$:

$$\begin{aligned}\operatorname{argmax}^{(r)}(x_0, x_1, \ldots, x_{n-1}) = {} & \sum_{i \in S(r,n)} \prod_{j=0}^{i} (1 - x_j^2) \prod_{k=i+1}^{n-1} (1 + x_k)^2 - (n-1)^{(r)} \cdot \prod_{j=0}^{n-1} (1 - x_j^2) \\ & + \sum_{i \in S(r,n)} \prod_{j=0}^{i} (1 + x_j)^2 - (n-1)^{(r)} \cdot \prod_{j=0}^{n-1} (1 + x_j)^2,\end{aligned}$$

or, equivalently,

$$\begin{aligned}\operatorname{argmax}^{(r)}(x_0, x_1, \ldots, x_{n-1}) = {} & \sum_{i \in S(r,n)} \left(\prod_{j=0}^{i} (1 - x_j^2) \prod_{k=i+1}^{n-1} (1 + x_k)^2 + \prod_{j=0}^{i} (1 + x_j)^2 \right) \\ & - (n-1)^{(r)} \left(\prod_{j=0}^{n-1} (1 - x_j^2) + \prod_{j=0}^{n-1} (1 + x_j)^2 \right).\end{aligned}$$

Finally, we also mention the following recursive relation:

$$\begin{aligned}\operatorname{argmax}^{(r)}(x_0, \ldots, x_{n-1}, x_n) = {} & \operatorname{argmax}^{(r)}(x_0, \ldots, x_{n-1}) \cdot (1 - \operatorname{argmax}^{(0)}(\max(x_0, \ldots, x_{n-1}), x_n)) \\ & + n^{(r)} \cdot \operatorname{argmax}^{(0)}(\max(x_0, \ldots, x_{n-1}), x_n),\end{aligned}$$

implied by the definition of $\operatorname{argmax}^{(r)}$ and the following fact:

$$\operatorname{argmax}(x_0, \ldots, x_{n-1}, x_n) = \begin{cases} \operatorname{argmax}(x_0, \ldots, x_{n-1}) & \text{if } \max(x_0, \ldots, x_{n-1}) \ge x_n, \\ n & \text{if } \max(x_0, \ldots, x_{n-1}) < x_n. \end{cases}$$

This formula yields a (in general, not minimal) polynomial expression of $\operatorname{argmax}^{(r)}(x)$ from those of $\operatorname{argmax}^{(0)}(x_0, x_1)$ and $\max(x)$.

5 Polynomial expressions of max and argmax functions for two variables

First we note that $x_0 < x_1$ if and only if we have (as integers) $\overline{x_0} + x_1 \ge p$, that is, $\operatorname{argmax}^{(0)}(x_0, x_1)$ is equal to the carry to the next digit by the p-ary addition of two single-digit values $\overline{x_0}$ and x_1. A minimal polynomial expression of this carry function, denoted by φ_1, has been determined in [6, 7].

Lemma 5.1 ([6, 7]). *For $y_0, y_1 \in \mathbb{F}_p$, we have*

$$\varphi_1(y_0, y_1) = \sum_{d=1}^{p-1} (-1)^d d^{-1} y_0(y_0 - 1) \cdots (y_0 - d + 1) y_1(y_1 - 1) \cdots (y_1 - (p - d) + 1),$$

where the d^{-1} in the right-hand side means the inverse of d as an element of $\mathbb{F}_p$.

Combining this with $\operatorname{argmax}^{(0)}(x_0, x_1) = \varphi_1(\overline{x_0}, x_1)$, we obtain the following proposition.

Proposition 5.2. *When $n = 2$, a minimal polynomial expression of* $\operatorname{argmax}^{(0)}(x_0, x_1)$ *is given by*

$$\operatorname{argmax}^{(0)}(x_0, x_1) = \sum_{d=1}^{p-1} d^{-1}(x_0 + 1)(x_0 + 2) \cdots (x_0 + d) x_1(x_1 - 1) \cdots (x_1 - (p - d) + 1).$$

Example 5.3. By using Proposition 5.2, for small primes p, we have the following minimal polynomial expressions of $\operatorname{argmax}^{(0)}(x_0, x_1)$:

$$\begin{aligned}
\text{when } p = 2, &\quad \operatorname{argmax}^{(0)}(x_0, x_1) = (x_0 + 1)x_1, \\
\text{when } p = 3, &\quad \operatorname{argmax}^{(0)}(x_0, x_1) = -(x_0 + 1)(x_0 - x_1)x_1, \\
\text{when } p = 5, &\quad \operatorname{argmax}^{(0)}(x_0, x_1) = -(x_0 + 1)(x_0^2 - x_0x_1 + x_0 + x_1^2)(x_0 - x_1)x_1, \\
\text{when } p = 7, &\quad \operatorname{argmax}^{(0)}(x_0, x_1) = -(x_0^4 + 5x_0^3x_1 + 2x_0^3 + 3x_0^2x_1^2 + x_0^2x_1 + 4x_0^2 + 5x_0x_1^3 \\
&\qquad\qquad + 6x_0x_1^2 + 3x_0 + x_1^4)(x_0 + 1)(x_0 - x_1)x_1.
\end{aligned}$$

We also have the following relation between the functions max and argmax deduced from their definitions.

Lemma 5.4. *We have*

$$\max(x) = \sum_{i=0}^{n-1} x_i \cdot \chi(\operatorname{argmax}(x) = i)$$

for any n. In particular, we have

$$\max(x_0, x_1) = x_0 \cdot (1 - \operatorname{argmax}(x_0, x_1)) + x_1 \cdot \operatorname{argmax}(x_0, x_1).$$

A straightforward substitution of the result of Proposition 5.2 into the right-hand side of Lemma 5.4 yields an almost, but not yet minimal, polynomial expression of $\max(x_0, x_1)$. This expression can be converted to a minimal polynomial expression. Indeed, for $p = 2$, the results above imply that $\max(x_0, x_1) = x_0 \cdot (1 - (x_0 + 1)x_1) + x_1 \cdot (x_0 + 1)x_1$, which is equal to the correct value $x_0 + x_1 + x_0x_1$ for any $x_0, x_1 \in \mathbb{F}_p$ due to the relations $x_0^2 = x_0$ and $x_1^2 = x_1$. On the other hand, we have the following result for $p > 2$.

Theorem 5.5. *When $p \geq 3$, we have the following minimal polynomial expression of* $\max(x_0, x_1)$*:*

$$\begin{aligned}
\max(x_0, x_1) = (x_1 - x_0) &\sum_{d=2}^{p-2} d^{-1}(x_0 + 1)(x_0 + 2) \cdots (x_0 + d) x_1(x_1 - 1) \cdots (x_1 - (p - d) + 1) \\
&+ x_0 + (x_0 + 1)^2(1 - (x_1 + 1)^{p-1}) + (1 - x_0^{p-1})x_1^2.
\end{aligned}$$

Proof. Throughout the proof, a notation $f \equiv g$ means that f and g define an identical function on $\mathbb{F}_p$. First, since $p \geq 3$, Proposition 5.2 implies

$$\begin{aligned}
x_0 \operatorname{argmax}(x_0, x_1) &= x_0 \sum_{d=1}^{p-1} d^{-1}(x_0 + 1)(x_0 + 2) \cdots (x_0 + d) x_1(x_1 - 1) \cdots (x_1 - (p - d) + 1) \\
&= x_0 \sum_{d=2}^{p-2} d^{-1}(x_0 + 1)(x_0 + 2) \cdots (x_0 + d) x_1(x_1 - 1) \cdots (x_1 - (p - d) + 1) \\
&\quad + x_0(x_0 + 1)x_1(x_1 - 1) \cdots (x_1 - (p - 2)) \\
&\quad - x_0(x_0 + 1)(x_0 + 2) \cdots (x_0 + p - 1)x_1,
\end{aligned}$$

and we have $x_0(x_0+1)(x_0+2)\cdots(x_0+p-1)x_1 \equiv 0$ for the last term above. Similarly, we have

$$\begin{aligned}x_1\operatorname{argmax}(x_0,x_1) &= x_1\sum_{d=1}^{p-1} d^{-1}(x_0+1)(x_0+2)\cdots(x_0+d)x_1(x_1-1)\cdots(x_1-(p-d)+1)\\ &= x_1\sum_{d=2}^{p-2} d^{-1}(x_0+1)(x_0+2)\cdots(x_0+d)x_1(x_1-1)\cdots(x_1-(p-d)+1)\\ &\quad+(x_0+1)x_1^2(x_1-1)\cdots(x_1-(p-2))\\ &\quad-(x_0+1)(x_0+2)\cdots(x_0+p-1)x_1^2,\end{aligned}$$

and, for the last two terms above, we have

$$\begin{aligned}(x_0+1)x_1^2(x_1-1)\cdots(x_1-(p-2)) &\equiv -(x_0+1)x_1(x_1-1)\cdots(x_1-(p-2)),\\ (x_0+1)(x_0+2)\cdots(x_0+p-1)x_1^2 &\equiv (p-1)!\cdot\delta_0(x_0)x_1^2 = -(1-x_0^{p-1})x_1^2,\end{aligned}$$

where we used $x_1^2 \equiv x_1((x_i-(p-1))-1)$ and Wilson's theorem $(p-1)! \equiv -1 \pmod p$.

By combining these results to Lemma 5.4, we have

$$\begin{aligned}\max(x_0,x_1) &= x_0 - x_0\operatorname{argmax}(x_0,x_1) + x_1\operatorname{argmax}(x_0,x_1)\\ &\equiv (x_1-x_0)\sum_{d=2}^{p-2} d^{-1}(x_0+1)(x_0+2)\cdots(x_0+d)x_1(x_1-1)\cdots(x_1-(p-d)+1)\\ &\quad+x_0-x_0(x_0+1)x_1(x_1-1)\cdots(x_1-(p-2))\\ &\quad-(x_0+1)x_1(x_1-1)\cdots(x_1-(p-2))+(1-x_0^{p-1})x_1^2\\ &= (x_1-x_0)\sum_{d=2}^{p-2} d^{-1}(x_0+1)(x_0+2)\cdots(x_0+d)x_1(x_1-1)\cdots(x_1-(p-d)+1)\\ &\quad+x_0-(x_0+1)^2x_1(x_1-1)\cdots(x_1-(p-2))+(1-x_0^{p-1})x_1^2,\end{aligned}$$

and, for the second last term above, we have

$$(x_0+1)^2x_1(x_1-1)\cdots(x_1-(p-2)) \equiv (x_0+1)^2\cdot(p-1)!\cdot\delta_{p-1}(x_1) = -(x_0+1)^2(1-(x_1+1)^{p-1}),$$

where we used Wilson's theorem again. Hence, we have

$$\begin{aligned}\max(x_0,x_1) &\equiv (x_1-x_0)\sum_{d=2}^{p-2} d^{-1}(x_0+1)(x_0+2)\cdots(x_0+d)x_1(x_1-1)\cdots(x_1-(p-d)+1)\\ &\quad+x_0+(x_0+1)^2(1-(x_1+1)^{p-1})+(1-x_0^{p-1})x_1^2,\end{aligned}$$

which is our claim in the statement. □

6 Polynomial expressions of some other functions

In this section, we study the following two $\mathbb{F}_p$-valued functions relevant to functions max and argmax:

$$\begin{aligned}\operatorname{ismax}(y;x) &= \chi(\max(x)=y),\\ \operatorname{nummax}^{(r)}(x) &= \#\{x_i \mid \max(x)=x_i)\}^{(r)},\end{aligned}$$

where $x \in (\mathbb{F}_p)^n$ and $y \in \mathbb{F}_p$. These functions would be useful in practical situations where there can be "ties" in the vote.

By a careful interpretation of the definitions, we obtain minimal polynomials of these functions (which, however, consist of a lot of terms).

Proposition 6.1. *Using the notation from Section 2, the following are minimal polynomial expressions:*

$$
\begin{aligned}
\operatorname{ismax}(y;x) &= \sum_{t=0}^{p-1} \delta_t(y) \sum_{i=0}^{n-1}\Big(\prod_{j<i} L_t(x_j)\cdot \delta_t(x_i)\cdot \prod_{k>i} L_{t+1}(x_k)\Big),\\
\operatorname{nummax}^{(0)}(x) &= \sum_{i=0}^{n-1} \chi(\max(x)=x_i)\\
&= \sum_{i=0}^{n-1}\sum_{0\le t\le p-1}\Big(\delta_t(x_i)\prod_{j\ne i} L_{t+1}(x_j)\Big),\\
\operatorname{nummax}^{(r)}(x) &= \sum_{k=1}^{n} k^{(r)}\cdot \chi(\#\{i \mid \max(x)=x_i\}=k)\\
&= \sum_{k=1}^{n} k^{(r)}\Big(\sum_{I\in\binom{n}{k}}\sum_{0\le t\le p-1}\Big(\prod_{i\in I}\delta_t(x_i)\prod_{j\notin I} L_t(x_j)\Big)\Big),
\end{aligned}
$$

where the notation "$I \in \binom{n}{k}$" means that I is a k-element subset of $\{0,1,\dots,n-1\}$.

Proof. For the function ismax, given a constant $t \in \mathbb{F}_p$, we have $\max(x) = t$ if and only if there is an index i such that $x_j < t$ for every $j < i$, $x_i = t$, and $x_k \le t$ for every $k > i$; such an index i is unique if it exists. This observation (in particular, the uniqueness of i) implies our claim.

For the function $\operatorname{nummax}^{(0)}$, the function value is obtained by first counting the number of indices i with $\max(x) = x_i$ (or equivalently, $\chi(\max(x) = x_i) = 1$) and then taking the remainder of the number modulo p (i.e., just considering the number in $\mathbb{F}_p$). Moreover, given a constant $t \in \mathbb{F}_p$, we have $\max(x) = x_i = t$ if and only if $x_i = t$ and $x_j \le t$ for every $j \ne i$. This observation implies our claim.

For the function $\operatorname{nummax}^{(r)}$, given an integer $k \ge 1$ and a constant $t \in \mathbb{F}_p$, we have $\max(x) = t$ and $\#\{i \mid \max(x) = x_i\} = k$ if and only if there is a k-element set I of indices such that $x_i = t$ for every $i \in I$ and $x_j < t$ for every $j \notin I$; such a set I is unique if it exists. This observation (in particular, the uniqueness of I) implies our claim (note that $0^{(r)} = 0$ for any r). □

When $p \in \{2, 3\}$, we give the following explicit minimal polynomial expressions of $\operatorname{ismax}(y; x)$.

Proposition 6.2. *When $p = 2$, a minimal polynomial expression of* $\operatorname{ismax}(y; x)$ *is given by*

$$\operatorname{ismax}(y;x) = y + \prod_{i=0}^{n-1}(1+x_i).$$

When $p = 3$, a minimal polynomial expression of $\operatorname{ismax}(y; x)$ *is given by*

$$\operatorname{ismax}(y;x) = -y^2 + y\Big(\prod_{i=0}^{n-1}(1+x_i)^2 + \prod_{i=0}^{n-1}(1-x_i^2) + 1\Big) + \prod_{i=0}^{n-1}(1-x_i^2).$$

Proof. First, we note that $\operatorname{ismax}(y; x) = 1 - (y - \max(x))^{p-1}$, by the definition of the function. When $p = 2$, the right-hand side becomes $y + \max(x) + 1$ and now the claim follows from Corollary 3.2.

On the other hand, when $p = 3$, we have

$$\operatorname{ismax}(y;x) = 1 - (y-\max(x))^2 = -y^2 - y\max(x) + 1 - \max(x)^2.$$

Now we have $1 - \max(x)^2 = 1$ if $x_i = 0$ for all i, and $= 0$ otherwise. This implies that

$$1-\max(x)^2 = \prod_{i=0}^{n-1}\delta_0(x_i) = \prod_{i=0}^{n-1}(1-x_i^2),$$

and now the claim follows from Proposition 3.3. □

Example 6.3. When $p = 2$, a minimal polynomial expression of $\operatorname{nummax}^{(r)}(x)$ is given by

$$\operatorname{nummax}^{(r)}(x) = e_{2^r} + n^{(r)}\prod_{i=0}^{n-1}(1-x_i).$$

This can be seen by the following argument. When $\max(x) = 0$, i.e., $x_i = 0$ for all i, we have $\mathrm{nummax}^{(r)} = n^{(r)}$ for any r, which accounts for the second term. As $(\sum_{i=0}^{n-1} x_i)^{(r)} \equiv e_{2^r}(x) \bmod 2$ by the result of [3] (see also [6, Example 1]), we obtain the equality.

7 Future subject: Multi-digit cases

We note that the previous sections studied functions with single-digit input values taken from $\mathbb{F}_p$; in such a formulation, to handle larger input values we have to choose a larger prime p as well, which will result in polynomial expressions of the functions with higher degrees and much more involved structures. Another option to handle larger values is to express the input values in *multi-digit* forms; now each component of the input is identified with its p-ary expansion, therefore the entire input is regarded as a two-dimensional matrix over $\mathbb{F}_p$ rather than a one-dimensional vector (over a larger field). In the latter model, the base field $\mathbb{F}_p$ can be kept small even if the input values become larger. On the other hand, a large input value will then increase the total number of components of the input matrix, but this shortcoming might sometimes be avoidable in practice by implementation techniques such as parallel computation. This suggests that polynomial expressions of functions with multi-digit inputs are important as well. However, even if the polynomial expression of a given function is understood well for single-digit input cases, it is in general a non-trivial task to deduce a polynomial expression of the function for multi-digit input cases.

To study multi-digit versions of the functions, for an ℓ-digit parameter $t = t_{\ell-1}p^{\ell-1} + \cdots + t_1 p + t_0 \in \{0, \ldots, p^\ell - 1\}$, with $t_0, t_1, \ldots, t_{\ell-1} \in \{0, \ldots, p-1\}$, and a tuple z of ℓ variables $z_0, z_1, \ldots, z_{\ell-1}$ over $\mathbb{F}_p$, we define the multi-digit low-pass function $L_t(z)$ by

$$L_t(z) = \chi(z_{\ell-1}p^{\ell-1} + \cdots + z_1 p + z_0 < t \text{ as integers}).$$

We also extend the definition to the case $t = p^\ell$ by setting $L_{p^\ell}(z) = 1$ for any z.

Here we consider the multi-digit version of the function $\mathrm{argmax}^{(r)}$ as an example that is relatively easier to handle. For $\ell \geq 1$ and for ℓ-digit input values $0 \leq x_i \leq p^\ell - 1$ $(i = 0, \ldots, n-1)$, $\mathrm{argmax}^{(r)}(x_0, \ldots, x_{n-1})$ is defined to be the r-th digit of the least index i with $x_i = \max(x_0, \ldots, x_{n-1})$. Then the same argument as that in Section 4 implies the following result.

Proposition 7.1. *Let $\ell \geq 1$. For $0 \leq i \leq n-1$, let the i-th component x_i of the input be given by $x_i = x_{i,\ell-1}p^{\ell-1} + \cdots + x_{i,1}p + x_{i,0}$, with $x_{i,0}, x_{i,1}, \ldots, x_{i,\ell-1} \in \mathbb{F}_p$ (naturally identified with $\{0, 1, \ldots, p-1\}$). Let $S(r,n) = \{h \cdot p^r - 1 \mid 1 \leq h \leq \lfloor (n-1)/p^r \rfloor\}$. Then we have*

$$\mathrm{argmax}^{(r)}(x) = \sum_{1 \leq t \leq p^\ell - 1} \Big(\sum_{i \in S(r,n)} \prod_{0 \leq j \leq i} L_t(x_j) \prod_{i+1 \leq k \leq n-1} L_{t+1}(x_k) - (n-1)^{(r)} \cdot \prod_{0 \leq j \leq n-1} L_t(x_j) \Big).$$

To obtain a polynomial expression of $\mathrm{argmax}^{(r)}(x)$ in terms of the input components $x_{i,j}$, we study polynomial expressions of the ℓ-digit low-pass function $L_t(z)$, where $t = (t_0, t_1, \ldots, t_{\ell-1})$ and $z = (z_0, z_1, \ldots, z_{\ell-1})$ are naturally identified with $t_0 + t_1 p + \cdots + t_{\ell-1}p^{\ell-1}$ and $z_0 + z_1 p + \cdots + z_{\ell-1}p^{\ell-1}$, respectively. We note that, we have $z < t$ if and only if there is an integer $h \in \{0, \ldots, \ell-1\}$ such that $z_j = t_j$ for any $h+1 \leq j \leq \ell-1$ and $z_h < t_h$; and this condition is satisfied for at most one h. This implies that

$$L_t(z) = \sum_{h=0}^{\ell-1} L_{t_h}(z_h) \prod_{j=h+1}^{\ell-1} \delta_{t_j}(z_j) \tag{7.1}$$

and, similarly,

$$L_{t+1}(z) = L_{t_0+1}(z_0) \prod_{j=1}^{\ell-1} \delta_{t_j}(z_j) + \sum_{h=1}^{\ell-1} L_{t_h}(z_h) \prod_{j=h+1}^{\ell-1} \delta_{t_j}(z_j) \tag{7.2}$$

(we recall that we have extended the definition of the single-digit low-pass function as $L_p(z_j) = 1$). Substituting the minimal polynomials for $L_t(z)$ and $L_{t+1}(z)$ in (7.1) and (7.2) into the equality in Proposition 7.1 yields the minimal polynomial for the multi-digit version of $\operatorname{argmax}^{(r)}$. However, the expression thus obtained will consist of too many terms as the number ℓ of input digits increases; we give an example only for a small case below and leave a more concise expression for the function in the general case as a future research topic.

Example 7.2. Let $p = 2$ and $\ell = 2$. Then the set $S(r, n)$ is $S(r, n) = \{h \cdot 2^r - 1 \mid 1 \le h \le \lfloor (n-1)/2^r \rfloor\}$. By setting $S'(r, n) = S(r, n)$ if $(n-1)^{(r)} = 0$ and $S'(r, n) = S(r, n) \cup \{n-1\}$ if $(n-1)^{(r)} = 1$, in the same way as Example 4.3, it follows from Proposition 7.1 and relations (7.1) and (7.2) that

$$\begin{aligned}&\operatorname{argmax}^{(r)}(x)\\&\quad = \sum_{1\le t\le 3}\Bigg(\sum_{i\in S'(r,n)}\prod_{0\le j\le i}(L_{t_0}(x_{j,0})\delta_{t_1}(x_{j,1}) + L_{t_1}(x_{j,1}))\prod_{i+1\le k\le n-1}(L_{t_0+1}(x_{k,0})\delta_{t_1}(x_{k,1}) + L_{t_1}(x_{k,1}))\Bigg).\end{aligned} \tag{7.3}$$

By the relations $L_0(x_{j,h}) = 0$, $\delta_0(x_{j,h}) = L_1(x_{j,h}) = 1 + x_{j,h}$, $\delta_1(x_{j,h}) = x_{j,h}$ and $L_2(x_{j,h}) = 1$, the summand in the right-hand side of (7.3) for each $t \in \{1, 2, 3\}$ is: when $t = 1$ (i.e., $(t_0, t_1) = (1, 0)$),

$$\begin{aligned}&\sum_{i\in S'(r,n)}\prod_{0\le j\le i}(L_1(x_{j,0})\delta_0(x_{j,1}) + L_0(x_{j,1}))\prod_{i+1\le k\le n-1}(L_2(x_{k,0})\delta_0(x_{k,1}) + L_0(x_{k,1}))\\&\qquad = \sum_{i\in S'(r,n)}\prod_{0\le j\le i}(1 + x_{j,0})(1 + x_{j,1})\prod_{i+1\le k\le n-1}(1 + x_{k,1}),\end{aligned}$$

when $t = 2$ (i.e., $(t_0, t_1) = (0, 1)$),

$$\begin{aligned}&\sum_{i\in S'(r,n)}\prod_{0\le j\le i}(L_0(x_{j,0})\delta_1(x_{j,1}) + L_1(x_{j,1}))\prod_{i+1\le k\le n-1}(L_1(x_{k,0})\delta_1(x_{j,1}) + L_1(x_{j,1}))\\&\qquad = \sum_{i\in S'(r,n)}\prod_{0\le j\le i}(1 + x_{j,1})\prod_{i+1\le k\le n-1}(1 + x_{k,0}x_{k,1}),\end{aligned}$$

when $t = 3$ (i.e., $(t_0, t_1) = (1, 1)$),

$$\sum_{i\in S'(r,n)}\prod_{0\le j\le i}(L_1(x_{j,0})\delta_1(x_{j,1}) + L_1(x_{j,1}))\prod_{i+1\le k\le n-1}(L_2(x_{k,0})\delta_1(x_{k,1}) + L_1(x_{k,1})) = \sum_{i\in S'(r,n)}\prod_{0\le j\le i}(1 + x_{j,0}x_{j,1}).$$

Summarizing, we have the following minimal polynomial expression for $\operatorname{argmax}^{(r)}(x)$:

$$\begin{aligned}\operatorname{argmax}^{(r)}(x) = \sum_{i\in S'(r,n)}\Bigg(&\prod_{0\le j\le i}(1 + x_{j,0})(1 + x_{j,1})\prod_{i+1\le k\le n-1}(1 + x_{k,1})\\&+ \prod_{0\le j\le i}(1 + x_{j,1})\prod_{i+1\le k\le n-1}(1 + x_{k,0}x_{k,1}) + \prod_{0\le j\le i}(1 + x_{j,0}x_{j,1})\Bigg).\end{aligned}$$

Such multi-digit extensions of the results on the other functions in this paper seem to be difficult, which we leave as a future research topic. Here we just conclude this section with a small example.

Proposition 7.3. *Let $p = 2$, and consider the two-bit inputs $y = 2y_1 + y_0 \in \{0, 1, 2, 3\}$ and $x_i = 2x_{i,1} + x_{i,0} \in \{0, 1, 2, 3\}$ for $0 \le i \le n-1$, where $y_j, x_{i,j} \in \mathbb{F}_2$. Then the following is a minimal polynomial expression of the two-bit version of the function* ismax*:*

$$\begin{aligned}&\operatorname{ismax}(y; x) = \operatorname{ismax}(y; x_0, x_1, \ldots, x_{n-1})\\&\quad = y_1y_0 + y_1\prod_{i=0}^{n-1}(1 + x_{i,1}x_{i,0}) + (y_1 + y_0)\prod_{i=0}^{n-1}(1 + x_{i,1}) + (y_1 + 1)\prod_{i=0}^{n-1}(1 + x_{i,1})(1 + x_{i,0}).\end{aligned}$$

Proof. As the right-hand side of the statement satisfies the minimality conditions for the degrees, it suffices to verify that the values of both terms are equal for any input values.

First we note that, for any set I of index pairs (i, j), we have

$$\prod_{(i,j)\in I}(1 + x_{i,j}) = \chi(x_{i,j} = 0 \text{ for all } (i, j) \in I).$$

Similarly, we have

$$\prod_{i}(1 + x_{i,1}x_{i,0}) = \chi(\text{for any } i, \text{ either } x_{i,1} = 0 \text{ or } x_{i,0} = 0 \text{ holds}).$$

We divide the argument according to the values of y_1 and y_0. When $y_1 = y_0 = 0$, we have $\mathrm{ismax}(y; x) = 1$ if and only if $x_{i,1} = x_{i,0} = 0$ for every index i. Now the right-hand side of the statement becomes $\prod_{i=0}^{n-1}(1 + x_{i,1}) \cdot (1 + x_{i,0})$, which coincides with $\mathrm{ismax}(y; x)$ by the remark above.

When $y_1 = 0$ and $y_0 = 1$, the right-hand side of the statement becomes $\prod_{i=0}^{n-1}(1 + x_{i,1}) + \prod_{i=0}^{n-1}(1 + x_{i,1}) \cdot (1 + x_{i,0})$. Now if at least one of $x_{i,1}$ is 1, then we have $\mathrm{ismax}(y; x) = 0$ by definition, while the value of the polynomial becomes 0 as well, by the remark above, as desired. In the remaining case where $x_{i,1} = 0$ for every i, we have $\mathrm{ismax}(y; x) = 1$ if and only if $x_{i,0} = 1$ for some i; while the polynomial now becomes $1 + \prod_{i=0}^{n-1}(1 + x_{i,0})$. By the remark above, the value of the polynomial coincides with $\mathrm{ismax}(y; x)$, as desired.

When $y_1 = 1$ and $y_0 = 0$, the right-hand side of the statement becomes $\prod_{i=0}^{n-1}(1 + x_{i,1}x_{i,0}) + \prod_{i=0}^{n-1}(1 + x_{i,1})$. Now if $x_{i,1} = 0$ for every i, then we have $\mathrm{ismax}(y; x) = 0$ by definition, while the value of the polynomial becomes $1 + 1 = 0$ as well, by the remark above, as desired. In the remaining case where $x_{i,1} = 1$ for some i, let I denote the set of indices i with $x_{i,1} = 1$ (hence now $I \neq \emptyset$). In this case, we have $\mathrm{ismax}(y; x) = 1$ if and only if $x_{i,0} = 0$ for every $i \in I$; while the polynomial now becomes $\prod_{i \in I}(1 + x_{i,0})$. By the remark above, the value of the polynomial coincides with $\mathrm{ismax}(y; x)$, as desired.

Finally, when $y_1 = y_0 = 1$, the right-hand side of the statement becomes $1 + \prod_{i=0}^{n-1}(1 + x_{i,1}x_{i,0})$. By the remark above, this polynomial takes the value 1 if and only if $x_{i,1} = x_{i,0} = 1$ for some index i; this condition is precisely the same as the condition for $\mathrm{ismax}(y; x)$ in the present case to take the value 1, by definition. This completes the proof. □

Acknowledgment: The authors would like to thank Takuro Abe for fruitful discussions.

Funding: The first named author was partially supported by JST PRESTO Grant Number JPMJPR16E3. The second named author was partially supported by KAKENHI, Grant-in-Aid for Scientific Research (C) 16K05083. The third named author was partially supported by JST PRESTO Grant Number JPMJPR14E8. The fourth named author was partially supported by KAKENHI, Grant-in-Aid for Young Scientists (B) JP5800009.

References

[1] T. Araki, J. Furukawa, Y. Lindell, A. Nof and K. Ohara, High-throughput semi-honest secure three-party computation with an honest majority, in: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, New York (2016), 805–817.

[2] R. Bost, R. A. Popa, S. Tu and S. Goldwasser, Machine learning classification over encrypted data, IACR Cryptology ePrint Archive (2014), https://eprint.iacr.org/2014/331.pdf.

[3] J. Boyar, R. Peralta and D. Pochuev, On the multiplicative complexity of Boolean functions over the basis (cap, +, 1), *Theoret. Comput. Sci.* **235** (2000), no. 1, 43–57.

[4] J. H. Cheon, M. Kim and M. Kim, Search-and-compute on encrypted data, in: *Proceedings of Financial Cryptography and Data Security 2015—FC 2015*, Lecture Notes in Comput. Sci. 8976, Springer, Berlin (2015), 142–159.

[5] C. Gentry, Fully homomorphic encryption using ideal lattices, in: *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing—STOC'09*, ACM, New York (2009), 169–178.

[6] S. Kaji, T. Maeno, K. Nuida and Y. Numata, Polynomial expressions of carries in p-ary arithmetics, preprint (2015), http://arxiv.org/abs/1506.02742.

[7] K. Nuida and K. Kurosawa, (Batch) fully homomorphic encryption over integers for non-binary message spaces, in: *Advances in Cryptology–EUROCRYPT 2015*, Lecture Notes in Comput. Sci. 9056, Springer, Berlin (2015), 537–555.

[8] A. Shamir, How to share a secret, *Commun. ACM* **22** (1979), no. 11, 612–613.

[9] C. Sturtivant and G. S. Frandsen, The computational efficacy of finite-field arithmetic, *Theoret. Comput. Sci.* **112** (1993), 291–309.