FUNCTIONAL GRAPHS OF POLYNOMIALS OVER FINITE FIELDS

SERGEI V. KONYAGIN, FLORIAN LUCA, BERNARD MANS, LUKE MATHIESON, MIN SHA, AND IGOR E. SHPARLINSKI

Abstract. Given a function f in a finite field $\mathbb{F}_q$ of q elements, we define the functional graph of f as a directed graph on q nodes labelled by the elements of $\mathbb{F}_q$ where there is an edge from u to v if and only if $f(u) = v$. We obtain some theoretical estimates on the number of non-isomorphic graphs generated by all polynomials of a given degree. We then develop a simple and practical algorithm to test the isomorphism of quadratic polynomials that has linear memory and time complexities. Furthermore, we extend this isomorphism testing algorithm to the general case of functional graphs, and prove that, while its time complexity deviates from linear by a (usually small) multiplier dependent on graph parameters, its memory complexity remains linear. We exploit this algorithm to provide an upper bound on the number of functional graphs corresponding to polynomials of degree d over $\mathbb{F}_q$. Finally, we present some numerical results and compare function graphs of quadratic polynomials with those generated by random maps and pose interesting new problems.

1. Introduction

Let $\mathbb{F}_q$ be the finite field of q elements and of characteristic p. For a function $f : \mathbb{F}_q \to \mathbb{F}_q$ we define the functional graph of f as a directed graph $\mathcal{G}_f$ on q nodes labelled by the elements of $\mathbb{F}_q$ where there is an edge from u to v if and only if $f(u) = v$.

Clearly each connected component of $\mathcal{G}_f$ contains one cycle (possibly of length 1 corresponding to a fixed point) with several trees attached to some of the cycle nodes.

We note that when we talk about connectivity in a directed graph we always mean the connectivity of the *indirected graph* it induces in a natural way.

2010 *Mathematics Subject Classification.* 05C20, 05C85, 11T06, 11T24.

Key words and phrases. polynomial maps, functional graphs, finite fields, character sums, algorithms on trees.

Here we are mostly interested in the graphs $\mathcal{G}_f$ associated with polynomials $f \in \mathbb{F}_q[X]$ of given degree d.

Some of our motivation comes from the natural desire to better understand Pollard's ρ-algorithm (see [6, Section 5.2.1]). We note that although this algorithm has been used and explored for decades, there is essentially only one theoretical result due to Bach [1]. In fact, even a heuristic model adequately describing this algorithm is not quite clear, as the model of random maps, analysed by Flajolet and Odlyzko [9], does not take into account the restrictions on the number of preimages. The model analysed by MacFie and Panario [23] approximates Pollard's algorithm better but it perhaps still does not capture it in full. Polynomial maps can also be considered as building blocks for constructing hash functions. For these applications, it is important to understand the intrinsic randomness of such maps.

Further motivation to investigation of the graphs $\mathcal{G}_f$ comes from the theory of dynamical systems, as $\mathcal{G}_f$ fully encodes many of the dynamical characteristics of the map f, such as the distribution of period (or cycle) and pre-period lengths.

In particular, we denote by $N_d(q)$ the number of distinct (that is, non-isomorphic) graphs $\mathcal{G}_f$ generated by all polynomials $f \in \mathbb{F}_q[X]$ of degree $\deg f = d$. Since there are exactly $(q-1)q^d$ polynomials of degree d, we have a trivial upper bound:

$$N_d(q) < q^{d+1}.$$

Here, we use some ideas of Bach and Bridy [2] together with some new ingredients to obtain nontrivial bounds on $N_d(q)$. We also refer to a recent work of Ostafe and Sha [29] with several more related results on statistic of functional graphs.

We also design simple and practical, yet efficient, algorithms to test the isomorphism of graphs $\mathcal{G}_f$ and $\mathcal{G}_g$ associated with two maps f and g. Furthermore, we design an efficient algorithm that generates a unique label for each functional graph. We use these algorithms to design an efficient procedure to list all $N_d(q)$ non-isomorphic graphs generated by all the polynomials $f \in \mathbb{F}_q[X]$ of degree $\deg f = d$.

We conclude by presenting some numerical results for functional graphs of quadratic polynomials. These results confirm that many (but not all, see below) basic characteristics of these graphs, except the total number of inner nodes, resemble those generated by random maps, as analysed by [9]. A probabilistic model of the distribution of cycles for functional graphs generated by polynomials (and more generally, by rational functions) has been also developed and numerically verified in [3], see also [11]. Here, besides cycle lengths, we also examine other characteristics of functional graphs generated by quadratic polynomials, for example, the number of connected components and the distribution of their sizes. Furthermore, these numerical results exhibit some interesting statistical properties of the graphs $\mathcal{G}_f$ for which either there is no model in the setting of random graphs, or they deviate, in a regular way, from such a model.

We also note that the periodic structure of functional graphs associated with monomial maps $x \mapsto x^d$ over finite fields and rings has been extensively studied (see [5, 12, 14, 21, 24, 31, 33, 36, 37] and references therein). However, these graphs are expected to be very different from those associated with generic polynomials.

In characteristic zero, graphs generated by preperiodic points of a map ψ (that is, by points that lead to finite orbits under iterations of ψ), have also been studied, (see, for example, [7, 8, 10, 27, 28, 30]).

We note that throughout the paper all implied constants in "O" symbols are *absolute*, unless stated otherwise.

2. Bounds on the number of distinct functional graphs of polynomials

2.1. **Upper bound.** To estimate $N_d(q)$ from the above, we use an idea of Bach and Bridy [2] which is based on the observation that for any polynomial automorphism ψ the composition map $\psi^{-1} \circ f \circ \psi$ has the same functional graph as f. So the idea is that if we can count the polynomials that are inequivalent under affine conjugations, this gives an upper bound for the number of dynamically inequivalent polynomials and therefore also for $N_d(q)$.

Thus, it needs to be shown that for any d there exists a rather small set of polynomials $\mathcal{F}_d$ such that for any polynomial $f \in \mathbb{F}_q[X]$ of degree d there is a polynomial automorphism ψ such that $\psi \circ f \circ \psi^{-1} \in \mathcal{F}_d$. Then we have $N_d(q) \le \#\mathcal{F}_d$. To construct the set $\mathcal{F}_d$ we introduce a group of certain transformations (see $\phi_{\lambda,\mu}$ in the proof of Theorem 2.1) on the set of polynomials and show that

- polynomials in each orbit generate isomorphic graphs;
- each orbit is sufficiently long, see the bounds (2.6) and (2.8) on the "defect" of each orbit compared to the size of the above group;
- most of the orbits are of the size of the above group, see (2.6).

This approach has been used in [2] for $d = 2$ and $q = 2^n$, in which case it is especially effective and leads to the bound

$$N_2(2^n) = \exp\left(O\left(\frac{n}{\log\log n}\right)\right) = q^{O(1/\log\log\log q)}. \tag{2.1}$$

For general pairs (d, q) this approach loses some of its power but still leads to nontrivial results, explicitly in both d and q. Recall that p is the characteristic of $\mathbb{F}_q$.

Theorem 2.1. *For any $d \ge 2$ and q, we have*

$$N_d(q) \le \begin{cases} q^{d-1} + (s-1)q^{d-1-\varphi(d-1)}, & \text{if } p \nmid d, \\ q^{d-1} + (s-1)q^{d-1-\varphi(d-1)} + (q-1)q^{d/p-1}, & \text{if } p \mid d, \end{cases}$$

where $s = \gcd(q-1, d-1)$ and φ is the Euler function.

Proof. For $\lambda \in \mathbb{F}_q^*$ and $\mu \in \mathbb{F}_q$, we define the automorphism

$$\phi_{\lambda,\mu} : \ X \mapsto \lambda X + \mu \tag{2.2}$$

with inverse $\phi_{\lambda,\mu}^{-1} : \ X \mapsto \lambda^{-1}(X - \mu)$. Particularly, these automorphisms form a group of order $(q-1)q$ in the usual way, which acts on the set of polynomials of degree d as the map

$$f(X) \to \phi_{\lambda,\mu}^{-1} \circ f \circ \phi_{\lambda,\mu}(X).$$

The number of the orbits of this group action can be calculated by the Burnside counting formula. This implies that

$$N_d(q) \le \frac{1}{(q-1)q} \sum_{(\lambda,\mu)} M_d(\lambda, \mu), \tag{2.3}$$

where the sum runs through all the pairs $(\lambda, \mu) \in \mathbb{F}_q^* \times \mathbb{F}_q$, and $M_d(\lambda, \mu)$ is the number of polynomials of degree d fixed by $\phi_{\lambda,\mu}$.

Trivially, we have

$$M_d(1, 0) = (q-1)q^d. \tag{2.4}$$

In the following, we want to estimate $M_d(\lambda, \mu)$ by fixing a pair $(\lambda, \mu) \in \mathbb{F}_q^* \times \mathbb{F}_q \setminus \{(1, 0)\}$.

For any polynomial f of degree d satisfying $\phi_{\lambda,\mu}^{-1} \circ f \circ \phi_{\lambda,\mu}(X) = f(X)$, we have

$$f(\lambda X + \mu) = \lambda f(X) + \mu.$$

Comparing the leading coefficients we derive

$$\lambda^{d-1} = 1, \tag{2.5}$$

which implies that

$$M_d(\lambda, \mu) = 0 \tag{2.6}$$

for those pairs (λ, μ) not satisfying (2.5).

First, suppose that $\lambda = 1$. Note that $\mu \ne 0$. Comparing the coefficients of X^{d-1} in $f(X + \mu)$ and $f(X) + \mu$, we obtain

$$da_d = 0. \tag{2.7}$$

Thus, $p \mid d$. Moreover, comparing the coefficients of X^{j-1} in $f(X+\mu)$ and $f(X)+\mu$ for every $j = 1, \ldots, d$, we also obtain relations of the form

$$ja_j\mu = F_j(a_d, \ldots, a_{j+1}, \mu), \qquad j = 1, \ldots, d,$$

for some polynomials

$$F_j \in \mathbb{F}_q[Z_d, \ldots, Z_{j+1}, V],$$

where in the case $j = d$ we have $F_d(V) = 0$, which corresponds to (2.7). In particular, for every $j = 1, \ldots, d$ with $\gcd(j,p) = 1$, we see that a_j is uniquely defined by $a_d, \ldots, a_{j+1}, \mu$. Hence, for $\mu \neq 0$ we get that

$$M_d(1,\mu) \le \begin{cases} 0, & \text{if } p \nmid d, \\ (q-1)q^{d/p}, & \text{if } p \mid d. \end{cases} \tag{2.8}$$

Assume now that $\lambda^{d-1} = 1$ but $\lambda \neq 1$, which implies that $d \ge 3$. We see that for every $j = 0, 1, \ldots, d$ there are polynomials

$$G_j \in \mathbb{F}_q[Z_d, \ldots, Z_{j+1}, U, V]$$

such that

$$a_j(\lambda^j - \lambda) = G_j(a_d, \ldots, a_{j+1}, \lambda, \mu).$$

Since $\lambda \neq 0, 1$, and $\lambda^{d-1} = 1$, it follows that for every j with $\gcd(j-1, d-1) = 1$ we have $\lambda^j \neq \lambda$ and thus a_j is uniquely defined by $a_d, \ldots, a_{j+1}, \lambda, \mu$. So, for $d \ge 3$ and any pair (λ, μ) satisfying $\lambda^{d-1} = 1$ and $\lambda \neq 1$, we have

$$M_d(\lambda,\mu) \le (q-1)q^{d-1-\varphi(d-1)}. \tag{2.9}$$

Notice that since $\lambda^{d-1} = 1$ and $\lambda \neq 1$, the element λ can take at most $\gcd(q-1, d-1) - 1$ values.

Using (2.3) together with (2.4), (2.6), (2.8) and (2.9), we complete the proof. □

In particular, $N_2(q) \le 2q - 1$, and for any $d \ge 3$ we have $N_d(q) \le 3q^{d-1}$. Furthermore, $N_d(q) \le q^{d-1}$ if $p \nmid d$ and $\gcd(q-1, d-1) = 1$.

2.2. **Lower bound: Idea of the proof.** Here we give a lower bound on $N_d(q)$ in the case of $\gcd(d, q-1) \ge 2$ and $\gcd(d-1, q) = 1$. In particular, this bound shows that (2.1) does not hold for fields of odd characteristic.

The idea is based on the following observation. Let $\mathcal{H}_a$ be the functional graph of $f_a(X) = X^d + a \in \mathbb{F}_q[X]$ with $a \in \mathbb{F}_q^*$. We note that the node a is the only node with in-degree 1, because the in-degree of every other node is

$$e = \gcd(d, q-1) \ge 2.$$

We now define the iterations of f_a

$$f_a^{(0)}(X) = X \qquad \text{and} \qquad f_a^{(k)}(X) = f_a\left(f_a^{(k-1)}(X)\right), \quad k = 1, 2, \ldots,$$

and consider the path of length J

$$a = f_a^{(0)}(a) \to f_a(a) = f_a^{(1)}(a) \to \cdots \to f_a^{(J)}(a) \tag{2.10}$$

originating from a. Then, the $(j+1)$-th node of this path has $e-1$ edges towards it from $\gamma f_a^{(j)}(a)$, where γ runs through the elements of the set

$$\Gamma_e^* = \Gamma_e \setminus \{1\},$$

where

$$\Gamma_e = \{\gamma \in \mathbb{F}_q \ : \ \gamma^e = 1\}.$$

Finally, we observe that $\gamma f_a^{(j)}(a)$ is an inner node if and only if the equation

$$z^d + a = \gamma f_a^{(j)}(a)$$

has a solution.

We now note that if two graphs $\mathcal{H}_a$ and $\mathcal{H}_b$ are isomorphic then, since a and b are unique nodes with the in-degree 1 in $\mathcal{H}_a$ and $\mathcal{H}_b$, respectively, the paths of the form (2.10) originating at a and b, and their neighbourhoods have to be isomorphic too.

For $j = 1, 2, \ldots$, we define $\eta_j(a)$ as the number of $\gamma \in \Gamma_e^*$ for which $\gamma f_a^{(j)}(a) - a$ is an e-th power nonresidue. Thus, $\eta_j(a)$ is the number of leaves amongst the nodes $\gamma f_a^{(j)}(a)$, $\gamma \in \Gamma_e^*$.

Therefore, for any J, the number of distinct vectors

$$\left(\eta_1(a), \ldots, \eta_J(a)\right), \qquad a \in \mathbb{F}_q^*, \tag{2.11}$$

gives a lower bound on $N_d(q)$. Our approach is to find a proper choice of J when q is sufficiently large such that each $\eta_j(a)$ $(j \ge 2)$ can run through the set $\{0, 1, \ldots, e-1\}$, then we can get a lower bound of the form

$$N_d(q) \ge e^{J-1}.$$

The idea is illustrated in Figure 2.1, where each "i" $(1 \le i \le e-1)$ in the circles represents a node defined by some $\gamma f_a^{(j)}(a)$, $\gamma \in \Gamma_e^*$.

We can express the appearance of a particular "pattern" among the leaves (2.11) algebraically, and use the Weil bound of multiplicative character sums (see [16, Theorem 11.23]) to show that, when J is not too large, all possible patterns appear; see Theorem 2.8 and its proof for more details. Note that this is similar to the well-known use of the Weil bound for showing that a sequence $\{1, \ldots, p\}$ contains any desired pattern of J consecutive residues and non-residues.

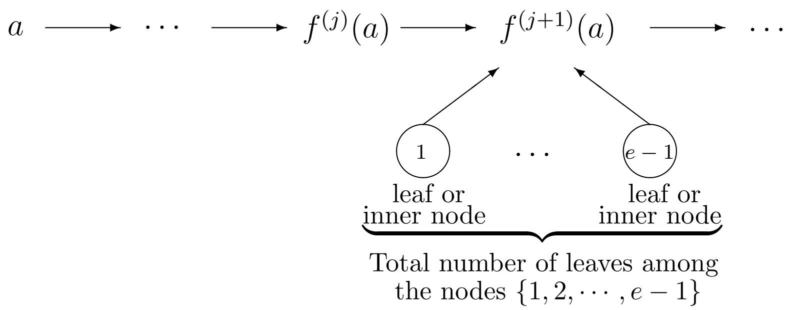

Figure 2.1. Idea of lower bound

2.3. **Lower bound: Technical details.** In order to realise the above approach, we need several technical statements.

As usual, we use $\overline{\mathbb{F}}_q$ to denote the algebraic closure of $\mathbb{F}_q$.

Let us consider the sequences of polynomials

$$F_0(X) = X \qquad \text{and} \qquad F_k(X) = (F_{k-1}(X))^d + X, \quad k = 1, 2, \ldots,$$

and also

$$G_{k,\gamma}(X) = \gamma F_k(X) - X.$$

Note that the roots of $G_{k,\gamma}$ are exactly those $z \in \overline{\mathbb{F}}_q$ for which F_k twists z by γ^{-1}.

We now investigate some arithmetic properties of polynomials $G_{k,\gamma}$ which we present in larger generality than we actually need for our purposes.

Lemma 2.2. *For any positive integers k and h and $\gamma, \delta \in \Gamma_e$, we have*

$$G_{k+h,\gamma} \equiv G_{h,\gamma} \pmod{G_{k,\delta}}.$$

Proof. We fix $\gamma, \delta \in \Gamma_e$ and prove the desired statement by induction on $h = 1, 2, \ldots$.

We note that for $\delta \in \Gamma_e$ we have

$$(F_k(X))^d = \left(\delta^{-1}\left(G_{k,\delta}(X) + X\right)\right)^d = \left(G_{k,\delta}(X) + X\right)^d. \tag{2.12}$$

For $h = 1$ we have $G_{1,\gamma}(X) = \gamma X^d + (\gamma - 1)X$. Hence, using (2.12), we derive

$$\begin{aligned} G_{k+1,\gamma}(X) &= \gamma\left((F_k(X))^d + X\right) - X \\ &= \gamma\left(G_{k,\delta}(X) + X\right)^d + (\gamma - 1)X \\ &\equiv \gamma X^d + (\gamma - 1)X \equiv G_{1,\gamma}(X) \pmod{G_{k,\delta}(X)}, \end{aligned}$$

so the desired congruence holds for $h = 1$.

Now assume that it also holds for $h = \ell$. Then

$$\begin{aligned} G_{k+\ell+1,\gamma}(X) &= \gamma\left(G_{k+\ell,\gamma}(X) + X\right)^d + (\gamma-1)X \\ &\equiv \gamma\left(G_{\ell,\gamma}(X) + X\right)^d + (\gamma-1)X \\ &\equiv G_{\ell+1,\gamma}(X) \pmod{G_{k,\delta}(X)}, \end{aligned}$$

which implies the desired result. □

Lemma 2.3. *For any positive integers k and m, we have*

$$\gcd(G_{k,\gamma}, G_{m,\gamma}) = G_{\gcd(k,m),\gamma}.$$

Proof. If $k = m$, then there is nothing to prove. Otherwise we note that for $m > k$, Lemma 2.2 implies $G_{m,\gamma} \equiv G_{m-k,\gamma} \pmod{G_{k,\gamma}}$. Thus

$$\gcd(G_{k,\gamma}, G_{m,\gamma}) = \gcd(G_{k,\gamma}, G_{m-k,\gamma}),$$

which immediately implies the desired result. □

Now, from Lemma 2.3 we derive that for $d = 2$ (and odd q), products of polynomials $G_{j,-1}$ over distinct integers are not perfect squares; see Lemma 2.4.

Lemma 2.4. *For $d = e = 2$, odd q, and any non-empty set $\mathcal{J}$ of positive integers, we have*

$$\prod_{j\in\mathcal{J}} G_{j,-1} \neq P^2$$

for any polynomial $P \in \overline{\mathbb{F}}_q[X]$.

Proof. Assume that m is the largest element of $\mathcal{J}$. Since $d = e = 2$, we have

$$F_0(X) = X, \quad F_k(X) = F_{k-1}(X)^2 + X, \quad G_{k,-1}(X) = -F_k(X) - X,$$

and thus

$$G_{1,-1}(X) = -X(X+2) \qquad \text{and} \qquad G_{2,-1} = -X(X+2)(X^2+1).$$

So, the cases $m = 1$ and $m = 2$ can be verified by direct calculations.

Now, we assume that $m \geq 3$. It suffices to show that $G_{m,-1}(X)$ has a simple root which is not a root of

$$Q_{m-1}(X) = \prod_{j=1}^{m-1} G_{j,-1}(X).$$

Let $f(m)$ be the number of distinct roots of $\gcd(G_{m,-1}(X), Q_{m-1}(X))$. By Lemma 2.3, these distinct roots are to be found among the distinct roots of

$$\prod_{1\le k\le m-1} \gcd(G_{m,-1}(X), G_{k,-1}(X)) = \prod_{1\le k\le m-1} G_{\gcd(m,k),-1}(X),$$

and the distinct roots of this last polynomial are the same as the distinct roots of the polynomial

$$\prod_{\substack{k|m\\1\le k<m}} G_{k,-1}(X),$$

which implies that

$$f(m) \le \sum_{\substack{k|m\\1\le k<m}} 2^k \le \sum_{1\le k\le m/s} 2^k = 2^{m/s+1} - 2, \tag{2.13}$$

where s is the minimal prime factor of m. More precisely, since the polynomial $G_{1,-1}(X)$ divides any other polynomial $G_{k,-1}(X)$, $k \ge 1$, we have

$$f(6) \le \deg G_{1,-1} + \deg G_{2,-1} + \deg G_{3,-1} - 2\deg G_{1,-1} = 10. \tag{2.14}$$

Now let us write

$$G_{m,-1}(X) = A(X)^2 B(X),$$

where $A(X), B(X) \in \mathbb{F}_q[X]$ are monic polynomials and $B(X)$ has only simple roots.

We claim that

$$\deg B(X) > f(m) \tag{2.15}$$

when $m \ge 5$. So, if $m \ge 5$, then $G_{m,-1}(X)$ has a root of odd multiplicity which is not a root of $Q_{m-1}(X)$, thus the desired result follows.

We prove the claim by contradiction. Hence, we suppose that

$$\deg B(X) \le f(m) \qquad \text{and} \qquad m \ge 5. \tag{2.16}$$

Note that since $X = 0$ is a simple root of $G_{m,-1}(X)$, we have $\deg B(X) \ge 1$. Since $A(X)$ divides $\gcd(G_{m,-1}(X), G'_{m,-1}(X))$ and $G_{m,-1}(X) = -F_{m-1}(X)^2 - 2X$, we obtain

$$F_{m-1}(X)^2 + 2X \equiv F_{m-1}(X)F'_{m-1}(X) + 1 \equiv 0 \pmod{A(X)},$$

which yields that

$$\begin{aligned} 0 &\equiv F_{m-1}(X)^2 F'_{m-1}(X) + F_{m-1}(X) \\ &\equiv -2XF'_{m-1}(X) + F_{m-1}(X) \pmod{A(X)}. \end{aligned}$$

Moreover, since

$$F_{m-1}(X) = F_{m-2}^2(X) + X \equiv X^2 + X \pmod{X^3} \tag{2.17}$$

and

$$\begin{aligned} F'_{m-1}(X) &= 2F_{m-2}(X)F'_{m-2}(X) + 1 \\ &\equiv 2X(2X+1) + 1 \equiv 2X + 1 \pmod{X^2}, \end{aligned}$$

it follows that

$$\begin{aligned} F_{m-1}(X) - 2XF'_{m-1}(X) &\equiv X^2 + X - 2X(2X+1) \\ &\equiv -3X^2 - X \pmod{X^3}, \end{aligned}$$

so this last polynomial is not the zero polynomial. In particular, there is a non-zero polynomial $C(X) \in \mathbb{F}_q[X]$ such that

$$F_{m-1}(X) - 2XF'_{m-1}(X) = A(X)C(X).$$

Because the degree of $F_{m-1}(X) - 2XF'_{m-1}(X)$ is at most 2^{m-1}, we deduce that

$$\deg C(X) \le 2^{m-1} - \deg A(X) = \frac{1}{2} \deg B(X). \tag{2.18}$$

We can also write

$$A(X) = \frac{F_{m-1}(X) - 2XF'_{m-1}(X)}{C(X)}.$$

Thus, we get that

$$-F_{m-1}(X)^2 - 2X = G_{m,-1}(X) = \frac{(F_{m-1}(X) - 2XF'_{m-1}(X))^2 B(X)}{C(X)^2},$$

and

$$-(F_{m-1}(X)^2 + 2X)C(X)^2 = (F_{m-1}(X) - 2XF'_{m-1}(X))^2 B(X). \tag{2.19}$$

Using the relation

$$F_{m-1}(X) = F_{m-2}(X)^2 + X, \qquad F'_{m-1}(X) = 2F_{m-2}(X)F'_{m-2}(X) + 1,$$

we reduce (2.19) modulo $F_{m-2}(X)$ to obtain

$$-(X^2 + 2X)C(X)^2 \equiv X^2 B(X) \pmod{F_{m-2}(X)}.$$

Notice that by (2.18) the polynomial on the left hand side has degree at most $\deg B(X) + 2$, and the polynomial on the right hand side has degree $\deg B(X) + 2$. Thus, in view of (2.16), if $\deg F_{m-2}(X) > f(m) + 2$, then the above congruence must in fact be an equality. Using (2.13) and (2.14), the above inequality is satisfied if

$$2^{m-2} > 2^{m/s+1} \quad \text{or} \quad m = 6,$$

where s is the minimal prime factor of m. The above inequality is also true for $m = 5$ and any integer $m \ge 7$.

So, if $m \geq 5$, we must have

$$-(X^2+2X)C(X)^2 = X^2B(X).$$

Furthermore,

$$\frac{B(X)}{C(X)^2} = -\frac{X+2}{X},$$

so that

$$\begin{aligned}-F_{m-1}(X)^2 - 2X &= \frac{(F_{m-1}(X) - 2XF'_{m-1}(X))^2B(X)}{C(X)^2}\\ &= -\frac{(X+2)}{X}(F_{m-1}(X) - 2XF'_{m-1}(X))^2,\end{aligned}$$

and then

$$X(F_{m-1}(X)^2+2X) = (X+2)(F_{m-1}(X) - 2XF'_{m-1}(X))^2.$$

We reduce the above relation modulo $F_{m-3}(X)$ using the fact that

$$\begin{aligned}F_{m-1}(X) = F_{m-2}(X)^2 + X &= (F_{m-3}(X)^2+X)^2+X\\ &\equiv X^2+X \pmod{F_{m-3}(X)}\end{aligned}$$

and

$$\begin{aligned}F'_{m-1}(X) &= 2(F^2_{m-3}(X)+X)(2F_{m-3}(X)F'_{m-3}(X)+1)+1\\ &\equiv 2X+1 \pmod{F_{m-3}(X)},\end{aligned}$$

to get that

$$X((X^2+X)^2+2X) \equiv (X+2)(X^2+X-2X(2X+1))^2 \pmod{F_{m-3}(X)}.$$

This leads to

$$8X^5+22X^4+12X^3 \equiv 0 \pmod{F_{m-3}(X)}, \tag{2.20}$$

which is impossible when $m \geq 5$. Indeed, for $k \geq 0$ we obviously have $F_k(X) \equiv X \pmod{X^2}$, so (2.20) implies

$$8X^3+22X^2+12X \equiv 0 \pmod{F_{m-3}(X)}$$

(see also (2.17)), which is impossible as $m \geq 5$ we have

$$\deg(8X^3+22X^2+12X) = 3 < 2^{m-3} = \deg F_{m-3}(X),$$

so (2.16) is impossible. Thus, (2.15) holds and we have the desired result for such values of m.

Finally, in order to finish the proof, it remains to handle the cases $m = 3,4$. We need to show that for $m = 3,4$, the polynomial $G_{m,-1}(X)$ has a simple root which is not a root of $G_{k,-1}(X)$ for any proper divisor k of m. For this we treat such $G_{m,-1}(X)$ as polynomials with integer coefficients and compute their discriminants. Notice that only the prime factors ℓ of such discriminants can be characteristics of fields where $G_{m,-1}(X)$ has double roots. Then, we factor such $G_{m,-1}(X)$ over the corresponding finite fields $\mathbb{F}_\ell$ for such primes ℓ, and can see that there is always an irreducible factor of $G_{m,-1}(X)$ with multiplicity 1 which is not a factor of any $G_{k,-1}(X)$ $(k|m, k < m)$ over $\mathbb{F}_\ell$. This completes the proof. □

When $d > 2$ we can study the arithmetic structure of the polynomials $G_{k,\gamma}$ by use a polynomial version of the ABC-conjecture of Mason [25, page 156, Corollary]. We note that it has also been discovered independently by Silverman [32] and Stothers [35, Theorem 1.1], see also [34]). We present it in Lemma 2.5 below.

For a polynomial $F \in \mathbb{F}_q[X]$ we use $\operatorname{rad}(F)$ to denote the product of all monic irreducible divisors of F.

Lemma 2.5. *Let A, B, C be nonzero polynomials in $\mathbb{F}_q[X]$ satisfying $A+B+C=0$ and $\gcd(A,B,C)=1$. If $\deg A \geq \deg \operatorname{rad}(ABC)$, then $A' = B' = C' = 0$.*

We are now ready to prove our main technical statement that we use for $d \geq 3$ which asserts that some general products of polynomials $G_{k,\gamma}$ over distinct integers are not perfect e-th powers.

Lemma 2.6. *Suppose that $\gcd(d-1,q)=1$. Then, for $d \geq 3$, $e = \gcd(d, q-1) \geq 2$, any $J \geq 2$ and any collection of integers not all equal to zero*

$$\mathcal{A} = \{\alpha_{j,\gamma} \in \{0,\ldots,e-1\} \ : \ 2 \leq j \leq J, \ \gamma \in \Gamma_e^*\},$$

we have

$$\prod_{j=2}^{J} \prod_{\gamma \in \Gamma_e^*} G_{j,\gamma}^{\alpha_{j,\gamma}} \neq P^e$$

for any polynomial $P \in \overline{\mathbb{F}}_q[X]$.

Proof. Clearly, we observe that for $j = 1,2,\ldots$ we have $X \mid G_{j,\gamma}$ but $X^2 \nmid G_{j,\gamma}$ for any $\gamma \in \Gamma_e^*$. Hence, define

$$G_{j,\gamma}^* = G_{j,\gamma}/X, \qquad j = 1,2,\ldots, \ \gamma \in \Gamma_e^*.$$

By counting the common roots, for distinct $\gamma, \delta \in \Gamma_e^*$ we have

$$\gcd(G_{j,\gamma}^*, G_{j,\delta}^*) = 1. \tag{2.21}$$

Therefore, applying Lemma 2.2, for any positive integers k and h and $\delta \in \Gamma_e^*$ we obtain

$$\begin{aligned}\deg\gcd\left(G^*_{k+h,\delta}, \prod_{\gamma\in\Gamma_e^*} G_{k,\gamma}\right) &= \deg\gcd\left(G^*_{k+h,\delta}, \prod_{\gamma\in\Gamma_e^*} G^*_{k,\gamma}\right)\\ &= \deg\gcd\left(G^*_{h,\delta}, \prod_{\gamma\in\Gamma_e^*} G^*_{k,\gamma}\right) \le d^h - 1.\end{aligned} \tag{2.22}$$

For any $k \ge 1$ and $\gamma \in \Gamma_e^*$, since $X \mid F_{k-1}$ and $\gcd(d-1,q) = 1$, we get that $(F^d_{k-1}/X)' \neq 0$; then applying Lemma 2.5 with $A = -G^*_{k,\gamma}$, $B = \gamma F^d_{k-1}/X$ and $C = \gamma - 1$, we derive

$$\begin{aligned}d^k - 1 < \deg\operatorname{rad}\left(G^*_{k,\gamma}F^d_{k-1}/X\right) &= \deg\operatorname{rad}\left(G^*_{k,\gamma}F_{k-1}\right)\\ &\le \deg\operatorname{rad}\left(G^*_{k,\gamma}\right) + d^{k-1}.\end{aligned}$$

Thus,

$$\deg\operatorname{rad}\left(G^*_{k,\gamma}\right) \ge (d-1)d^{k-1}, \quad k = 1,2,\ldots, \ \gamma \in \Gamma_e^*. \tag{2.23}$$

Denote

$$Q_{J,\mathcal{A}} = \prod_{j=2}^{J}\prod_{\gamma\in\Gamma_e^*} G_{j,\gamma}^{\alpha_{j,\gamma}},$$

and assume that $Q_{J,\mathcal{A}} = P^e$ for some $P \in \overline{\mathbb{F}}_q[X]$. Let

$$\widetilde{Q}_{J,\mathcal{A}} = \prod_{j=2}^{J}\prod_{\gamma\in\Gamma_e^*} G_{j,\gamma}^{\widetilde{\alpha}_{j,\gamma}},$$

where $\widetilde{\alpha}_{j,\gamma} = e - \alpha_{j,\gamma}$ if $\alpha_{j,\gamma} \neq 0$ and $\widetilde{\alpha}_{j,\gamma} = 0$ otherwise. Then, since each $\alpha_{j,\gamma} \le e-1$, we have

$$P \mid \prod_{j=2}^{J}\prod_{\gamma\in\Gamma_e^*,\, \alpha_{j,\gamma}\neq 0} G_{j,\gamma}.$$

Noticing that

$$\widetilde{Q}_{J,\mathcal{A}} = \left(\frac{\prod_{j=2}^{J}\prod_{\gamma\in\Gamma_e^*,\, \alpha_{j,\gamma}\neq 0} G_{j,\gamma}}{P}\right)^e,$$

we conclude that $\widetilde{Q}_{J,\mathcal{A}}$ is also a perfect e-th power. Let $k \ge 2$ be the largest $j \in \{2,\ldots,J\}$ for which one of the integers $\alpha_{j,\gamma}$, $\gamma \in \Gamma_e^*$ is positive. Considering, if necessary, $\widetilde{Q}_{J,\mathcal{A}}$ we can always assume that

$$\alpha = \min_{\gamma\in\Gamma_e^*}\{\alpha_{k,\gamma} \ : \ \alpha_{k,\gamma} > 0\} \le e/2.$$

We now fix some $\delta \in \Gamma_e^*$ with $\alpha_{k,\delta} = \alpha$, $1 \le \alpha \le e/2$. If a polynomial $H \in \mathbb{F}_q[X]$ is such that the product $H(G_{k,\delta}^*)^{\alpha_{k,\delta}}$ is a perfect e-th power, then

$$\mathrm{rad}\,(G_{k,\delta}^*)^e \mid H(G_{k,\delta}^*)^{\alpha_{k,\delta}}.$$

So,

$$\mathrm{rad}\,(G_{k,\delta}^*)^{e-\alpha_{k,\delta}} \mid H\left(G_{k,\delta}^*/\mathrm{rad}\,(G_{k,\delta}^*)\right)^{\alpha_{k,\delta}},$$

which, combining with (2.23), implies that

$$\begin{aligned}(2.24)\quad &\deg\gcd\left((G_{k,\delta}^*)^{e-\alpha_{k,\delta}}, H\right)\\ &\quad\ge \deg\gcd\left(\mathrm{rad}\,(G_{k,\delta}^*)^{e-\alpha_{k,\delta}}, H\right)\\ &\quad\ge (e-\alpha_{k,\delta})\deg\mathrm{rad}\,(G_{k,\delta}^*) - \alpha_{k,\delta}\left(\deg G_{k,\delta}^* - \deg\mathrm{rad}\,(G_{k,\delta}^*)\right)\\ &\quad\ge e(d-1)d^{k-1} - \alpha_{k,\delta}(d^k-1) \ge \frac{e}{2}(d-2)d^{k-1}.\end{aligned}$$

If $k = 2$, that is $J = 2$ and

$$Q_{J,\mathcal{A}} = \prod_{\gamma\in\Gamma_e^*} G_{k,\gamma}^{\alpha_{k,\gamma}} = P^e,$$

then by (2.24), we obtain

$$\deg\gcd\left((G_{k,\delta}^*)^{e-\alpha_{k,\delta}}, Q_{J,\mathcal{A}}/\left(G_{k,\delta}^*\right)^{\alpha_{k,\delta}}\right) \ge \frac{e}{2}(d-2)d^{k-1} > 0.$$

However, combining (2.21) with $x \nmid G_{k,\delta}^*$, we have

$$\deg\gcd\left((G_{k,\delta}^*)^{e-\alpha_{k,\delta}}, Q_{J,\mathcal{A}}/\left(G_{k,\delta}^*\right)^{\alpha_{k,\delta}}\right) = 0,$$

which leads to a contradiction. So, we must have $k \ne 2$, that is $k \ge 3$.

Now, combining (2.21) with (2.24), we have

$$\begin{aligned}(2.25)\quad &\deg\gcd\left(\left(G_{k,\delta}^*\right)^{e-\alpha_{k,\delta}}, \prod_{j=2}^{k-1}\prod_{\gamma\in\Gamma_e^*} G_{j,\gamma}^{\alpha_{j,\gamma}}\right)\\ &\quad= \deg\gcd\left(\left(G_{k,\delta}^*\right)^{e-\alpha_{k,\delta}}, Q_{J,\mathcal{A}}/\left(G_{k,\delta}^*\right)^{\alpha_{k,\delta}}\right)\\ &\quad\ge \frac{e}{2}(d-2)d^{k-1}.\end{aligned}$$

On the other hand, we deduce that

$$\begin{aligned}\deg\gcd&\left(\left(G_{k,\delta}^*\right)^{e-\alpha_{k,\delta}},\prod_{j=2}^{k-1}\prod_{\gamma\in\Gamma_e^*}G_{j,\gamma}^{\alpha_{j,\gamma}}\right)\\ &\le\deg\gcd\left(\left(G_{k,\delta}^*\right)^{e-\alpha_{k,\delta}},\prod_{\gamma\in\Gamma_e^*}G_{k-1,\gamma}^{\alpha_{k-1,\gamma}}\right)+\sum_{j=2}^{k-2}\sum_{\gamma\in\Gamma_e^*}\alpha_{j,\gamma}\deg G_{j,\gamma}\\ &\le\deg\gcd\left(\left(G_{k,\delta}^*\right)^{e-1},\prod_{\gamma\in\Gamma_e^*}G_{k-1,\gamma}^{e-1}\right)+(e-1)^2\sum_{j=2}^{k-2}d^j.\end{aligned}$$

Using (2.22) (with $h=1$) we get

$$\deg\gcd\left(\left(G_{k,\delta}^*\right)^{e-1},\prod_{\gamma\in\Gamma_e^*}G_{k-1,\gamma}^{e-1}\right)\le(e-1)(d-1).$$

Collecting the above estimates, we obtain

$$\begin{aligned}\deg\gcd&\left(\left(G_{k,\delta}^*\right)^{e-\alpha_{k,\delta}},\prod_{j=2}^{k-1}\prod_{\gamma\in\Gamma_e^*}G_{j,\gamma}^{\alpha_{j,\gamma}}\right)\\ &\le(e-1)(d-1)+(e-1)^2\frac{d^{k-1}-1}{d-1}.\end{aligned}\tag{2.26}$$

It is now easy to verify that (2.26) contradicts (2.25) when $d>3$. Indeed, combining (2.25) with (2.26), we get

$$\frac{e}{2}(d-2)d^{k-1}\le(e-1)(d-1)+(e-1)^2\frac{d^{k-1}-1}{d-1}.$$

Then, since $d>3$, we can get

$$e(d-1)d^{k-1}\le(e-1)(d-1)^2+(e-1)^2d^{k-1}-(e-1)^2,$$

and thus

$$(e-1)d^{k-1}\le(e-1)(d-1)^2-(e-1)^2,$$

which is impossible by noticing that $k\ge 3$.

Finally, we take $d=3$. Then, $e=3$. From (2.24), we have

$$\begin{aligned}\deg\gcd&\left(\left(G_{k,\delta}^*\right)^{e-\alpha_{k,\delta}},\prod_{j=2}^{k-1}\prod_{\gamma\in\Gamma_e^*}G_{j,\gamma}^{\alpha_{j,\gamma}}\right)\\ &\ge e(d-1)d^{k-1}-\alpha_{k,\delta}(d^k-1)\ge 3^k+1,\end{aligned}$$

which contradicts (2.26). The desired result now follows. □

Let $\mathcal{X}_e$ be the group of all multiplicative characters of $\mathbb{F}_q^*$ of order e, that is, characters χ with $\chi^e = \chi_0$, where χ_0 is the principal character. We also define $\mathcal{X}_e^* = \mathcal{X}_e \setminus \{\chi_0\}$.

We recall the following special case of the Weil bound of character sums (see [16, Theorem 11.23]).

Lemma 2.7. *For any polynomial $Q(X) \in \mathbb{F}_q[X]$ with Z distinct zeros in $\overline{\mathbb{F}}_q$ and which is not a perfect e-th power in the ring of polynomials over $\overline{\mathbb{F}}_q$, and $\chi \in \mathcal{X}_e^*$, we have*

$$\left| \sum_{a \in \mathbb{F}_q} \chi\left(Q(a)\right) \right| \le Z q^{1/2}.$$

We are now ready to establish a lower bound on $N_d(q)$.

Theorem 2.8. *Suppose that* $\gcd(d-1, q) = 1$. *Then, for any $d \ge 2$ and $e = \gcd(d, q-1) \ge 2$, we have*

$$N_d(q) \ge q^{\rho_{d,e} + o(1)}$$

as $q \to \infty$, where

$$\rho_{d,e} = \frac{1}{2(e - 1 + \log d / \log e)}.$$

Proof. We define J by the inequalities

$$(de^{e-1})^{J+1} \le q^{1/2}/\log q < (de^{e-1})^{J+2}.$$

Note that for the fixed d, we have $J \ge 2$ when q is sufficiently large. For each $j = 2, \ldots, J$ and $\gamma \in \Gamma_e^*$ we choose a representative $\sigma_{j,\gamma}$ of the quotient group $\mathbb{F}_q^*/\Delta_e$, where Δ_e is the group of e-th powers, and consider the collection

$$\boldsymbol{\sigma} = \{\sigma_{j,\gamma} \ : \ j = 2, \ldots, J, \ \gamma \in \Gamma_e^*\}.$$

Let $A(\boldsymbol{\sigma})$ denote the number of $a \in \mathbb{F}_q^*$ such that

$$\gamma f_a^{(j)}(a) - a \in \sigma_{j,\gamma}\Delta_e, \qquad \text{for all } j = 2, \ldots, J, \ \gamma \in \Gamma_e^*.$$

Clearly, if for any $\boldsymbol{\sigma}$ as in the above we have

$$A(\boldsymbol{\sigma}) > 0, \tag{2.27}$$

then the vector (2.11) takes all e^{J-1} possible values and thus we have

$$N_d(q) \ge e^{J-1},$$

which implies the desired result. So, it remains to prove (2.27) for sufficiently large q and the above choice of J.

Let χ be a primitive character of order e (that is, a generator of $\mathcal{X}_e$). We can now express $A(\boldsymbol{\sigma})$ using character sums

$$A(\boldsymbol{\sigma}) = \sum_{a\in\mathbb{F}_q^*} \frac{1}{e^{(e-1)(J-1)}} \prod_{j=2}^{J} \prod_{\gamma\in\Gamma_e^*} \sum_{\alpha_{j,\gamma}=0}^{e-1} \chi^{\alpha_{j,\gamma}}\left(G_{j,\gamma}(a)/\sigma_{j,\gamma}\right).$$

This follows directly from the relation $G_{j,\gamma}(a) = \gamma f_a^{(j)}(a) - a$ and the following orthogonality relation

$$\sum_{\alpha_{j,\gamma}=0}^{e-1} \chi^{\alpha_{j,\gamma}}\left(G_{j,\gamma}(a)/\sigma_{j,\gamma}\right) = \begin{cases} e & \text{if } G_{j,\gamma}(a)/\sigma_{j,\gamma} \in \Delta_e, \\ 0 & \text{otherwise,} \end{cases}$$

by noticing that χ is of order e and is a generator of $\mathcal{X}_e$ (see, also [16, Section 3.1]).

Expanding the product, and changing the order of summation we obtain $e^{(e-1)(J-1)}$ character sums parametrized by different choices of $\alpha_{j,\gamma} \in \{0,\ldots,e-1\}$, $j = 2,\ldots,J$, $\gamma \in \Gamma_e^*$.

Note that $\chi^0\left(G_{j,\gamma}(a)/\sigma_{j,\gamma}\right) = 1$ if and only if $G_{j,\gamma}(a) \neq 0$, and each polynomial $G_{j,\gamma}$ has degree d^j. So, by estimating the number of distinct zeros of the polynomial $\prod_{j=2}^{J}\prod_{\gamma\in\Gamma_e^*} G_{j,\gamma}$, the term corresponding to the choice $\alpha_{j,\gamma} = 0$, $j = 2,\ldots,J$, and $\gamma \in \Gamma_e^*$, is not less than $q - 1 - (e-1)d^{J+1}$.

For the other terms, using Lemma 2.4 (if $d = 2$) and Lemma 2.6 (if $d \geq 3$), we apply Lemma 2.7 to each of them by noticing that each corresponding polynomial has at most $(e-1)d^{J+1}$ distinct zeros. Hence, we obtain

$$\begin{aligned} A(\boldsymbol{\sigma}) &\geq \frac{q - 1 - (e-1)d^{J+1} - (e^{(e-1)(J-1)} - 1)(e-1)d^{J+1}q^{1/2}}{e^{(e-1)(J-1)}} \\ &\geq \frac{1}{e^{(e-1)(J-1)}}\left(q - 1 - (de^{e-1})^{J+1}q^{1/2}\right), \end{aligned}$$

which implies (2.27) for sufficiently large q and the above choice of J. □

We remark that

$$\max_{d,\ e|d} \rho_{d,e} = \rho_{2,2} = 1/4$$

is the exponent corresponding to quadratic polynomials over $\mathbb{F}_q$ with an odd q.

3. Isomorphism testing of functional graphs

3.1. **Preliminaries.**

In this section, we give a practical and efficient isomorphism testing algorithm for functional graphs of quadratic polynomials that is linear (in time and memory). We also extend this isomorphism testing algorithm from quadratic polynomials to any arbitrary function with only a slight increase in the time complexity.

We note that the class of functional graphs over $\mathbb{F}_q$ coincides with the class of directed graphs on q nodes with all out-degrees equal to 1. However, the in-degrees depend on the particular function f associated with this graph.

Our algorithms do not depend on the arithmetic structure of q (for example, that this is a prime power) or on algebraic properties of the underlying domain (for example, that it has a structure of a field). Hence we present them for functional graphs over an arbitrary set of n elements.

Clearly, any functional graph is extremely sparse (with exactly n arcs) and the size of the input that is to be considered for efficient isomorphism testing is linear in the size of an adjacency list (that is, $O(n \log n)$), rather than an adjacency matrix (that is, $O(n^2)$). We first introduce several graph related notations.

3.2. **Notations and graph input size.**

Given two functions f and h, we denote the functional graph $\mathcal{G}_f$ of f as $\mathcal{G}$ and the functional graph $\mathcal{G}_h$ of h as $\mathcal{H}$. Given a functional graph $\mathcal{G}$, we collect its connected components of the same size in the sets $C_i^{\mathcal{G}}$ with $1 \le i \le s^{\mathcal{G}}$, where $s^{\mathcal{G}}$ is the total number of distinct sizes of components of $\mathcal{G}$. For each set $C_i^{\mathcal{G}}$ we denote the size of the components in the set by $k_i^{\mathcal{G}}$ and the size of the set itself by $c_i^{\mathcal{G}} = \#C_i^{\mathcal{G}}$. Let

$$k_*^{\mathcal{G}} = \max_{1 \le i \le s^{\mathcal{G}}} k_i^{\mathcal{G}} \qquad \text{and} \qquad c_*^{\mathcal{G}} = \max_{1 \le i \le s^{\mathcal{G}}} c_i^{\mathcal{G}}. \tag{3.1}$$

One can think of the component sizes as a kind of "spectrum" of $\mathcal{G}$, $k_*^{\mathcal{G}}$ as the larges entry in this spectrum (that is, the size of the largest component of $\mathcal{G}$) and with $c_*^{\mathcal{G}}$ as the largest occurring multiplicity. In particular, $c_*^{\mathcal{G}}$ is bounded by the number of components (which is small in expectation for random graphs), and if $\mathcal{G}$ is connected then $c_*^{\mathcal{G}} = 1$ and $k_*^{\mathcal{G}} = \#\mathcal{G}$.

To give a small example, the function $f(X) = X^2 + 6$ over the field $\mathbb{F}_{13}$ has 3 components, of sizes $(2, 2, 9)$. Hence $s^{\mathcal{G}_f} = 3$, $c_1^{\mathcal{G}_f} = 2$, $c_2^{\mathcal{G}_f} = 1$, $k_1^{\mathcal{G}_f} = 2$, $k_2^{\mathcal{G}_f} = 9$. This gives $c_*^{\mathcal{G}_f} = 2$ and $k_*^{\mathcal{G}_f} = 9$.

When there is no ambiguity, we omit the superscript $\mathcal{G}$. For convenience we denote the *in-degree* of a vertex v as $d^{-}(v)$ and the corresponding *in-neighbourhood* as $N^{-}(v)$. Since the *out-degree* of any vertex is 1, each connected component C in a functional graph has exactly one cycle (which may be a self-loop), which we denote $\mathrm{Cyc}(C)$. Each vertex is the root of a (possibly empty) tree.

3.3. Isomorphism testing of functional graphs of quadratic polynomials.

We now present our meta-algorithm to test the isomorphism. It comprises three phases:

Phase 1: Given two functional graphs $\mathcal{G}$ and $\mathcal{H}$, we first identify the connected components in each graph, and the associated cycle and trees in each component.

Phase 2: For each component we produce a canonical encoding.

Phase 3: Finally we construct a prefix tree (formally a trie [20]), using the encodings of $\mathcal{G}$, noting at each vertex of the trie the number of code strings that terminate at that vertex. Then for each encoded component of $\mathcal{H}$ we match the code string against the trie, and decrement the counter at the appropriate trie vertex.

If all counters are zero after this is complete, the two graphs are isomorphic.

The first phase is achieved by combining a cycle detection algorithm and depth-first search, as laid out in Algorithm 1.

Algorithm 1 IDENTIFICATION OF CONNECTED COMPONENTS

```
while unassigned vertices remain do
    Pick an unassigned vertex $v$.
    Perform Floyd's cycle detection algorithm starting at $v$.
    for each cycle vertex $u$ do
        Perform a depth-first search on the tree attached at $u$.
    end for
end while
```

The cycle detection algorithm can be done in linear time and space (in the size of each connected component) with Floyd's algorithm [19] using only two pointers. The depth-first search is a simple pre-order traversal of the tree and thus only requires linear time and space [18]. In total, the complexity of the first phase is thus linear in time and space with the size of the graph. Note that this phase is independent of the function f (it has linear complexity for any function f), leading to the following lemma.

Lemma 3.1. *For any functional graph $\mathcal{G}$ of n vertices, Algorithm 1 identifies all Connected Components and has linear time and memory complexities.*

On the other hand, the second phase depends on the nature of the function. In this section, we focus on quadratic polynomials which provide an especially interesting case when considering the isomorphism of functional graphs. A k-ary tree is *full* (or *proper*) if every non-leaf vertex has exactly k children. (Note that, here, a full k-ary tree need not have all root-leaf paths have the same length.) As a quadratic polynomial can have at most one repeated root, the functional graph is almost a full binary tree. This allows certain savings in building a canonical labelling of the graph. We note that if there is a repeated root, we can deal with the containing component specially by noting which vertex has one child, and adding a dummy second child, then in the two graphs under consideration the dummy vertices must be matched to each other in any isomorphism.

We recall that the number of different binary trees on n nodes is the n-th Catalan number, which for large n is roughly proportional to $4^n/n^{3/2}$. It is well-known that binary trees can be encoded with exactly $2n+1$ bits [18]: by first extending the original tree by adding "special" nodes whenever a null subtree is present (two for leaves and one for non-full internal nodes), and then doing a pre-order traversal of the tree labelling original nodes with ones and special nodes with zeroes. When the binary tree is full (which is our case with quadratic polynomials), only n bits suffice to encode the original tree by using a similar technique (simply making the original leaves the special nodes and then performing a pre-order traversal of the tree labelling internal nodes with ones and leaves with zeroes). Our canonical labelling extends this bound by including the cycle with a minimal number of extra bits.

To produce the canonical labelling of a functional graph derived from a quadratic polynomial we employ Algorithms 2 and 3, where ε is the empty string, s_i is the string s after circular shift to the bit position i, and $\mathrm{val}(s)$ is the interpretation of the string s as a number. In the description of the algorithms we denote string concatenation by $\circ$.

Algorithm 2 runs on each component in turn and produces a canonical label for the component by applying Algorithm 3, that is, function $\textsc{Label}(v)$, to each tree rooted on a vertex of the component's cycle (Figure 3.1 gives an example), concatenating these labels in the order given by the cycle, then shifting circularly the concatenated label to begin with the cycle vertex that gives the greatest value. Note that

Algorithm 2 Canonical Labelling

```
Require: component $C$
  s := $\varepsilon$
  for each vertex v in Cyc(C) do
    s := s $\circ$ Label(v)
  end for
  max := val($s_1$)
  maxpos := 1
  for i := 2 to # Cyc(C) do
    if val($s_i$) > max then
      max := val($s_i$)
      maxpos := i
    end if
  end for
  return $s_{maxpos}$
```

Algorithm 3 Label(v)

```
Require: vertex v.
if $d^-(v) = 0$ then
  return "0"
else
  left := Label(left(v))
  right := Label(right(v))
  if left < right then
    return 1 $\circ$ right $\circ$ left
  else
    return 1 $\circ$ left $\circ$ right
  end if
end if
```

if t such vertices exist (that is, t possible circular shifts leading to the greatest value), the component must have at least a t-fold symmetry of rotation around the cycle. Thus, this maximal orientation of the cycle is unique up to automorphism.

Algorithm 3 encodes a full, rooted, binary tree by assigning each vertex a single bit: 1 if the vertex is internal, 0 if it is a leaf. The label is then recursively built by concatenating the assigned bit of the current vertex v to the lexicographically sorted labels of its left child, $left(v)$, and right child, $right(v)$. In effect this produces a traversal of the tree where we traverse higher weight subtrees first. As each vertex

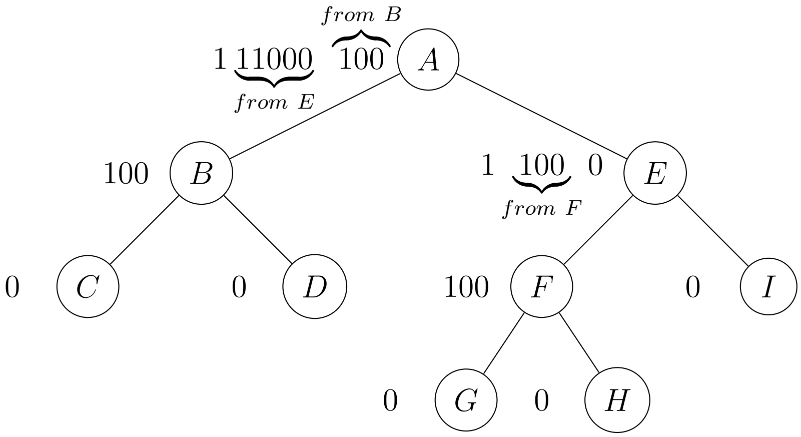

Figure 3.1. An example binary tree labelled with the canonical coding generated at each level by Algorithm 3.

contributes one bit to the label, the total length of the label is k bits for a component of size k and thus n bits for the entire graph.

Lemma 3.2. *For any functional graph $\mathcal{G}$ of a quadratic polynomial over $\mathbb{F}_q$ with $n = q$ vertices, Algorithms 2 and 3 build an n-bit size canonical labelling of $\mathcal{G}$ and have linear time and memory complexities.*

Proof. From the description of the traversal process in Algorithms 2 and 3, it is clear that each node v in the tree is associated with a canonical coding $\textsc{Label}(v)$ of size $|T_v|$ bits, where T_v is the subtree with root v. All leaves are labelled with 0, and the canonical label of the whole tree T_v has exactly $k = |T_v|$ bits. The overall memory requirement remains linear: both child labels can be discarded, on the fly, as a parent label is generated.

The worst-case time complexity is slightly more involved, a (lexicographic) sorting is required at each internal node. More precisely, each internal node v requires a number of (lexicographic) bit comparisons $\mathrm{comp}(v)$ equal to the size of the smallest label among both children:

$$\begin{aligned}\mathrm{comp}(v) &= \min\{|\textsc{Label}(\mathrm{left}(v))|, |\textsc{Label}(\mathrm{right}(v))|\}\\ &= \min\{|T_{left}(v)|, |T_{right}(v)|\} \le \left\lfloor \frac{|T(v)|-1}{2} \right\rfloor.\end{aligned} \tag{3.2}$$

Hence, we see that the worst case for the number of bit comparisons occurs when each subtree is balanced, that is when the full binary tree is complete. Using this simple recurrence, it is easy to see that this leads to less than $n \log n$ bit comparison s for any binary tree of size n, which is linear in the size of the input and completes the proof. □

Note that to finally test the isomorphism between two graphs $\mathcal{G}$ and $\mathcal{H}$, it remains to compare the canonical labellings of each connected components of each graph with one another (Phase 3). A general brute-force approach (by comparing canonical labellings of connected components pair-wise) could be ineffective (as shown in the next section). To keep it linear in the size of the input, the third phase builds a trie (or prefix tree) using the encodings of the functional graph $\mathcal{G}$ by inserting the canonical labelling of each connected components, obtained after Phase 2, one after the other. Each node in the trie is also equipped with a counter initialised to zero and incremented each time the node represents the terminating node of a newly inserted canonical labelling of a connected component. It then suffices to check that each canonical labelling of each connected component of $\mathcal{H}$ is represented in the trie, decrementing the respective counter each time there is a match. The two functional graphs are isomorphic if there is no mismatch for all canonical labellings of $\mathcal{H}$ (all counters are zero after all components have been considered), and are non-isomorphic otherwise.

Lemma 3.3. *For any functional graph $\mathcal{G}$ and $\mathcal{H}$, each with an n-bit canonical labelling, Phase 3 tests their isomorphism by comparing the canonical labelling of $\mathcal{G}$ and $\mathcal{H}$ and has linear time and memory complexities.*

Proof. It is easy to see that the trie built for the functional graph $\mathcal{G}$ has at most n nodes. This case is only possible if all canonical labellings of connected components are disjoint (that is, generate disjoint branches in the tree). As more canonical labels overlap, fewer nodes are created. If the labels match, the respective counter (and its size) are incremented, but the cost of increasing the counter remains lower than the cost of creating a distinct branch in the trie. Thus, the overall size remains $O(n)$ in memory space. It is also easy to see that creating the initial trie with the canonical labels of $\mathcal{G}$ takes $O(n)$ time and memory, and the same cost occurs for matching all canonical labels of $\mathcal{H}$ (and may stop before if the two graphs are not isomorphic). □

Again it is interesting to note that the complexity of Phase 3 does not depend on the type of functional graph but depends solely on the size of the canonical labelling.

Combining Lemmas 3.1, 3.2 and 3.3, we obtain the following theorem.

Theorem 3.4. *For any functional graphs $\mathcal{G}$ and $\mathcal{H}$ of quadratic functions with n vertices, Phases 1, 2 and 3 combined provide an isomorphism test that has linear time and memory complexities.*

It is also interesting to note that the trie built in Phase 3 provides a canonical representation of size $O(n)$ for any functional graph of size n. We exploit this property to present an algorithm to enumerate all functional graphs corresponding to polynomials of degree d over $\mathbb{F}_q$ in Section 3.5.

3.4. **General functional graph isomorphism.** In a general setting, there are numerous standard results that can be used. The graph isomorphism problem can be solved in time linear in the number of vertices for connected planar graphs [15] and (rooted) trees [17]. Also the fact that our graphs are directed is not an issue as there is a linear-time reduction from directed graph isomorphism to undirected graph isomorphism [26].

We first give a simple example of how these techniques can be combined to prove a simple upper bound for Functional Graph Isomorphism for arbitrary functions. Other combinations of these standard techniques are possible to give similar near-linear time. However we prove in the remainder of this section that simple and practical, yet efficient (in time and memory), techniques can be used by extending the algorithms of Section 3.3 to arbitrary functions.

Theorem 3.5. *For any functional graphs $\mathcal{G}$ and $\mathcal{H}$ of arbitrary functions with n vertices, there is an isomorphism test using standard algorithms with $O(c_*n)$ time complexity, where $c_* = \max\{c_*^{\mathcal{G}}, c_*^{\mathcal{H}}\}$ and $c_*^{\mathcal{G}}$, $c_*^{\mathcal{H}}$ are defined by* (3.1)*.*

Proof. A simple approach that can be applied to functional graphs is to run Algorithm 1 (that builds each connected component) and then compare the connected components of the two graphs pairwise, using the appropriate algorithm as a subroutine (for components with a cycle, we can use the planar graph algorithm, for components with a self-loop, we can use the rooted tree algorithm where we treat the vertex with the self-loop as the root). This involves at most $\binom{n}{2}$ comparisons and thus gives an $O(n^2)$ algorithm overall.

Using the sizes of the various components, we can refine this analysis slightly. Given two functional graphs $\mathcal{G}$ and $\mathcal{H}$, if we have the isomorphism $\mathcal{G} \cong \mathcal{H}$ then $s^{\mathcal{G}} = s^{\mathcal{H}}$ and for all $i \in [1, s^{\mathcal{G}}]$ we also have $c_i^{\mathcal{G}} = c_i^{\mathcal{H}}$ and $k_i^{\mathcal{G}} = k_i^{\mathcal{H}}$. (If the graphs are isomorphic, $c_* = c_*^{\mathcal{G}} = c_*^{\mathcal{H}}$.) On the other hand, if one of these pairs of values disagree then $\mathcal{G} \ncong \mathcal{H}$. Then, denoting these common values as s and c_i, k_i, $1 \le i \le s$, clearly for both graphs we have

$$\sum_{i=1}^{s} c_i k_i = n,$$

where n is the order of the graphs.

Clearly we only need to compare components in the same size class. This gives a running time proportional to:

$$\sum_{i=1}^{s} c_i^2 k_i \leq c_* \sum_{i=1}^{s} c_i k_i = c_* n,$$

where $c_* = \max_{i=1,\dots,s} c_i$. □

If each size class C_i is bounded, then this naïve algorithm is linear in the number of vertices. In the general case however it is likely there are numerous components of the same size [9] thus possibly leading to a worst-case bound of $O(n^2)$ time. Fortunately even in this case, as we now show that we can still solve the isomorphism problem with linear memory complexity and by increasing slightly the cost of building the canonical labels.

The challenge is that, in the general case, we cannot assume that the trees associated with each component are full, nor necessarily have any particular bound on the number of children (note that polynomials of degree d do however have at most d children in the trees).

For the general case we replace Algorithm 3 with Algorithms 4 and 5, and replace the call to Label in Algorithm 2 with a call to LeftLabel with the root vertex of the tree.

Algorithm 4 LeftLabel

```
Require: vertex $v$
label$_v$ := $\varepsilon$
labelSet := $\emptyset$
finalLabel := $\varepsilon$
if $v$ is not a leaf then
    label$_v$ := $1 \circ$ LeftLabel($left(v)$)
    if $v$ has a right child then
        labelSet := RightLabel($right(v)$)
    end if
    labelSet := labelSet $\cup$ {label$_v$}
    Sort(labelSet)
    for $i$ := 1 to #labelSet do
        finalLabel := finalLabel $\circ$ labelSet[$i$]
    end for
else
    finalLabel := 0
end if
return finalLabel
```

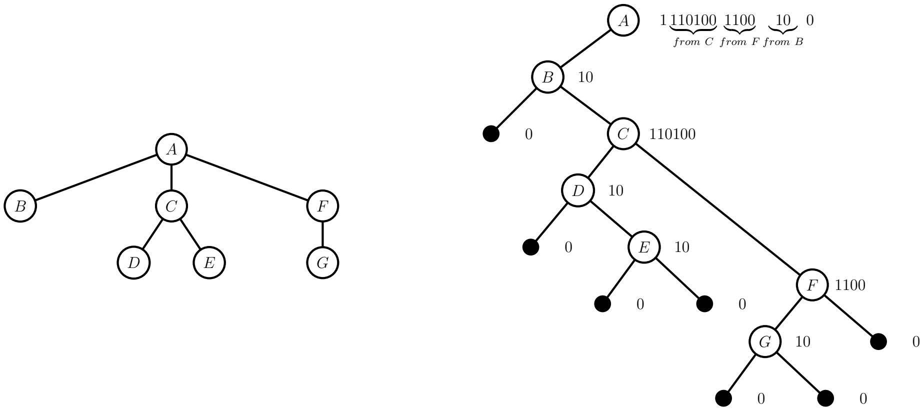

Figure 3.2. An example non-binary tree (left) and the equivalent binary tree (right) labelled with the canonical coding generated at each level by Algorithms 4 and 5. The black vertices in the binary tree on the right are the ad ded vertices.

Algorithm 5 RightLabel

```
Require: vertex $v$
label$_v$ := $\varepsilon$
labelSet := $\emptyset$
if $v$ is not a leaf then
    label$_v$ := $1 \circ$ LeftLabel($left(v)$)
else
    label$_v$ := 0
end if
if $v$ has a right child then
    labelSet := RightLabel($right(v)$)
end if
return labelSet $\cup$ {label$_v$}
```

That is, the second phase, in the general case, is achieved by Algorithms 2, 4 and 5, which take each component of the input graph(s), produce a canonical label by first labelling each tree rooted at a cycle vertex, concatenating these labels then shifting the label to obtain the maximum value. Ultimately, we consider these labels as bit strings with the final label of a component taking $2k$ bits where k is the number of vertices in the component. We can then encode the graph as a whole with $2n$ bits. To obtain this bound we represent the trees attached to the cycles with left-child-right-sibling binary trees (e.g. see Knuth [18] for binary representation of trees), in which the right child of a vertex is a sibling and the left child is the first child (we can take any ordering for our purposes).

The two tree labelling algorithms (LeftLabel and RightLabel) together produce the canonical labelling of the tree in several steps. First the tree is implicitly extended to a full binary tree by adding leaf vertices whenever a vertex is missing a child, except at the root, as it cannot have siblings, so the terminating leaf is superfluous. Each internal vertex is labelled with "1" and each leaf with "0". Each vertex extends its label by concatenating its label with the label of its left subtree, then adding this label to the set of labels received from its right subtree. If a vertex is a left child (that is, it is the first child of its parent in the normal representation), it sorts this set of labels, largest to smallest, concatenates them and passes this label to its parent (Figure 3.2 illustrates the process).

Lemma 3.6. *The combined Algorithms 4 and 5 perform at most $O(k^2)$ bit comparisons and use linear memory space to build a canonical label of size $2k$ bits for any component of size k.*

Proof. We only need to prove that these two algorithms perform at most $O(k^2)$ bit comparisons. Notice that the main cost at each internal node is to lexicographically sort the labels of its children, and the lexicographic sort of m labels of size n bits costs $O(mn)$ bit comparisons. Fix an arbitrary component C of size k. Suppose that there are t trees rooted at the cycle of C with sizes d_i, $1 \leq i \leq t$. Then, for labelling C, the number of bit comparisons is proportional to

$$\sum_{i=1}^{t} d_i \cdot k = k \sum_{i=1}^{t} d_i = k^2,$$

which concludes the proof. □

Combining the costs of labelling for all components, with the rest of the meta-algorithm, we obtain the following result for testing isomorphism.

Theorem 3.7. *For any functional graphs $\mathcal{G}$ and $\mathcal{H}$ of arbitrary functions with n vertices, there is an isomorphism test using $O(k_* \cdot n)$ bit comparisons and linear memory complexity, where $k_* = \max\{k_*^{\mathcal{G}}, k_*^{\mathcal{H}}\}$ and $k_*^{\mathcal{G}}$, $k_*^{\mathcal{H}}$ are defined by (3.1).*

Proof. We need to label all components. Using the sizes of various classes of components in the graph, that is, Lemma 3.6, the overall running time is proportional to:

$$\sum_{i=1}^{s} c_i k_i^2 \le k_* \sum_{i=1}^{s} c_i k_i = k_* n, \tag{3.3}$$

where

$$k_* = \max_{i=1,\dots,s} |k_i|.$$

Combining this result with Lemmas 3.1 and 3.3 completes the proof. □

It is interesting to note the trade-off between c_*, the maximum number of components of same size used in Theorem 3.5 and k_*, the largest component, used in Theorem 3.7, as it seemingly provides a choice among algorithms to test the isomorphism depending of related features of the graph. However, it should be emphasized that the comparison is not straightforward as the algorithm of Theorem 3.7 considers bit comparisons as the metric of the time cost, while Theorem 3.5 employs more involved algorithms.

We note that the bound (3.3) used in the proof of Theorem 3.7 together with Lemma 3.6 also lead to an upper bound on the size of the labelling of any functional graph.

Corollary 3.8. *The meta-algorithm used for isomorphism testing in Theorem 3.7 uses at most $O(k_* \cdot n)$ bit comparisons and linear memory space to build canonical labels of size of $2n$ bits that can be represented in a trie of size $O(n)$ for any functional graph of size n, where k_* is defined by (3.1).*

Proof. As each connected component of k vertices contributes $2k$ bits to the final labelling of the graph of size n, the total number of bits for representing all components is $2n$. Finally, using Phase 3, we can built a trie of at most $2n$ nodes to encode all canonical encodings. □

3.5. Counting functional graphs.

We now present an algorithm to enumerate all functional graphs corresponding to polynomials of degree d over $\mathbb{F}_q$ except that $d = 2$ and $2 \mid q$.

Theorem 3.9. *For any d and q except for $d = 2$ and $2 \mid q$, we can create a list of all $N_d(q)$ distinct functional graphs generated by all degree d polynomials $f \in \mathbb{F}_q[X]$ in $O(d^2 q^d \log^2 q)$ arithmetic operations and comparisons of bit strings of length $O(q^2)$.*

Proof. Let $m = \gcd(d-1, q-1)$ and let $\Omega = \{\omega_1, \dots, \omega_m\}$ be a set of representatives of the quotient group $\mathbb{F}_q^*/\mathcal{H}_m$, where $\mathcal{H}_m$ is the group of m-th powers in $\mathbb{F}_q$, that is

$$\mathcal{H}_m = \{\eta^m : \eta \in \mathbb{F}_q^*\}.$$

Recall the automorphism $\phi_{\lambda,\mu}$ defined in (2.2) for any $\lambda \in \mathbb{F}_q^*$ and $\mu \in \mathbb{F}_q$. We verify that for a polynomial

$$f(X) = \sum_{j=0}^{d} a_j X^j \in \mathbb{F}_q[X], \qquad \deg f = d, \tag{3.4}$$

we have

$$\begin{aligned}\phi_{\lambda,\mu}^{-1} \circ f \circ \phi_{\lambda,\mu}(X) &= \lambda^{-1}\left(f(\lambda X + \mu) - \mu\right)\\ &= \sum_{j=0}^{d} A_j X^j,\end{aligned}$$

for some coefficient $A_j \in \mathbb{F}_q$, $j = 0, \ldots, d$. In particular, we have

$$\begin{aligned}&A_d = \lambda^{d-1} a_d,\\ &A_{d-1} = \lambda^{d-2}\mu d a_d + \lambda^{d-2} a_{d-1},\\ &A_{d-2} = \lambda^{d-3}\mu^2 \frac{d(d-1)}{2} a_d + \lambda^{d-3}\mu(d-1)a_{d-1} + \lambda^{d-3} a_{d-2}.\end{aligned} \tag{3.5}$$

We claim that for any polynomial $f \in \mathbb{F}_q[X]$ of the form (3.4) we can find $\lambda \in \mathbb{F}_q^*$ such that $A_d(a_d; \lambda, \mu) \in \Omega$. Indeed, we can assume that $a_d = \omega_i \eta^m$ for some i $(1 \le i \le m)$ and $\eta \in \mathbb{F}_q^*$. Since there exist two integers s, t such that $s(d-1) + t(q-1) = m$, we have $a_d = \omega_i \eta^{s(d-1)}$. Then choosing $\lambda = \eta^{-s}$, we get $A_d(a_d; \lambda, \mu) = \omega_i \in \Omega$.

If $\gcd(d, q) = 1$, then we can find an element $\mu \in \mathbb{F}_q$ such that $A_{d-1} = 0$. Thus, it suffices to consider the polynomial F of the form $F(X) = A_d X^d + g(X)$ where $A_d \in \Omega$ and $g(X) \in \mathbb{F}_q[X]$ is of degree $d-2$. Therefore, it is enough to examine the graphs $\mathcal{G}_F$ only for such $mq^{d-1} < dq^{d-1}$ polynomials F.

Assume that $\gcd(d, q) \ne 1$. Then, we must have $d > 2$. Noticing that $d(d-1)/2$ is divisible by the characteristic p of $\mathbb{F}_q$, if $A_{d-1} \ne 0$ (that is $a_{d-1} \ne 0$), we can choose $\mu \in \mathbb{F}_q$ such that $A_{d-2} = 0$. So, it is enough to examine the graphs $\mathcal{G}_F$ only for such $2mq^{d-1} < 2dq^{d-1}$ polynomials F (that is satisfying $A_d \in \Omega$ together with $A_{d-1} = 0$ or $A_{d-2} = 0$).

Given such a polynomial $f \in \mathbb{F}_q[X]$ of degree d, we can construct the graph $\mathcal{G}_f$ in time $O(dq \log^2 q)$ (see [4]). After this, by Corollary 3.8, for each graph, in time $O(q^2)$ we compute its canonical label. Using the above discussion and inserting these labels in an ordered list of length at most $N_d(q)$ (or discarding if the label already in the list) gives an overall time of $O(dq^{d-1} \cdot dq \log^2 q) = O(d^2 q^d \log^2 q)$. □

In particular, the running time of the algorithm of Theorem 3.9 is at most $d^2 q^{d+2+o(1)}$.

4. Numerical results

4.1. **Preliminaries.** We note that the periodic structure of functional graphs has been extensively studied numerically (see, for example, [3]). These results indicate that "generic" polynomials, even of degree which is relatively smaller then q, lead to graphs with cycle lengths with the same distribution as of those associated with random maps (see [3, Section 5]). It is useful to recall that if the degree is not restricted than any map over $\mathbb{F}_q$ is represented by a polynomial.

It is not difficult to see that for an odd q, the functional graph of any quadratic polynomial over $\mathbb{F}_q$ has $(q-1)/2$ leaves. Indeed, for $f(X) = X^2 + a$ the node a is always an inner node with in-degree 1 while other nodes are of in-degree 0 or 2. Thus there are $1 + (q-1)/2 = (q+1)/2$ inner nodes and $(q-1)/2$ leaves. On the other hand, the graph of a random map on p nodes is expected to have $p/e \approx 0.3679\, p$ leaves. It is possible that there are some other structural distinctions. Motivated by this, we have studied numerically several other parameters of functional graph.

Our tests have been limited to quadratic polynomials in prime fields, which can be further limited to polynomials of the form $f(X) = X^2 + a$, $a \in \mathbb{F}_p$. Various properties of the corresponding function graphs $\mathcal{G}_f$ have been tested for all p polynomials of this form for the following sequences of primes:

- all odd primes up to 100 (mostly for the purpose of testing our algorithms, but this has also revealed an interesting property of $N_2(17)$);
- for the sequence of primes between 101 and 102407 where each prime is approximately twice the size of its predecessor;
- for the sequence of 30 consecutive primes between 204803 (which could also be viewed as the last element of the previous group) and 205171;
- for the sequence of 10 consecutive primes between 500009 and 500167;
- for the prime 1000003.

For these primes, we tested the number of distinct primes and also average and extreme values of several basic parameters of the graphs $\mathcal{G}_f$.

Our numerical results revealed that some of these parameters are the same as those of random graphs, but some (besides the aforementioned number of inner nodes) deviate in a rather significant way. More importantly (and as far as we are aware), some of these parameters of graphs have never been discussed in the literature before this work. Using our practical algorithms of Section 3 we have initiated the study of these interesting parameters.

We present some of our numerical results (limited to those that show some new and unexpected aspects in the statistics of the graphs $\mathcal{G}_f$), only for the primes of the last two groups, that is, for the set of primes

$$\{500009, 500029, 500041, 500057, 500069, 500083,\\ 500107, 500111, 500113, 500119, 500153, 500167, 1000003\}.$$

4.2. **Number of distinct graphs.**

We recall that Theorems 2.1 and 2.8 imply that

$$p^{1/4+o(1)} \le N_2(p) \le p$$

for all odd primes p. For all tested primes we have $N_2(p) = p$ except for $p = 2, 17$ in which cases $N_2(2) = 3$ and $N_2(17) = 16$. This indicates that most likely we have $N_2(p) = p$ for any odd prime p, except for $p = 17$. However, proving this may be difficult as the case of $p = 17$ shows that there is no intrinsic reason for this to be true (apart from the fact that, as p grows, it is natural to expect that the "probability" for a coincidence of two functional graphs of p distinct quadratic polynomials becomes smaller).

Finally, we note that $p = 17$ is a Fermat prime. Hence we have also checked the two next Fermat primes $p = 257$ and $p = 65537$ for which we still have $N_2(257) = 257$ and $N_2(65537) = 65537$. Note the no larger Fermat primes are known or expected to exist.

4.3. **Cyclic points and the giant components.**

Our numerical tests show that the average values of

- the number of cyclic points,
- the size of the largest connected components,

behave like expected from random maps, which are predicted to be $\sqrt{\pi p/2}$, (see [9, Theorem 2 (ii)]) and γp where $\gamma = 0.75788\ldots$, (see [9, Theorem 8 (ii)]), respectively.

It is also interesting to investigate the extreme values. More precisely, let $c(f)$ be the number of cyclic points of $\mathcal{G}_f$ and let

$$C(p) = \max\{c(f) \ : \ f(X) = X^2 + a, \ a \in \mathbb{F}_q\}.$$

In all our tests, except for the primes $p = 5, 13, 17$, the value of $c(f)$ is maximised on the function graphs of one of the polynomials $f_0(X) = X^2$ and $f_{-2}(X) = X^2 - 2$, for which

$$c(f_0) = r + 1 \qquad \text{and} \qquad c(f_{-2}) = (r + s)/2,$$

where r is the largest odd divisor of $p - 1$ and s is the largest odd divisor of $p + 1$, see [36, Theorem 6 (b)] and [36, Corollary 18 (b)], respectively (note that in [36] the polynomials are considered as acting on $\mathbb{F}_p^*$). In particular, if $p \equiv 3 \pmod 4$ then the function graph of X^2 has the largest possible number of cyclic points, which is $(p + 1)/2$. Hence,

$$C(p) = (p + 1)/2, \qquad \text{for } p \equiv 3 \pmod 4.$$

We also note that for any $p \geq 3$,

$$C(p) \geq \max\{r + 1, (r + s)/2\} \geq (p + 3)/4. \tag{4.1}$$

Furthermore, if $f(X) = X^2 + a$ with $a \in \mathbb{F}_p^*$ then the number of cyclic points of $\mathcal{G}_f$ is at most $3p/8 + O(1)$. Indeed, let $\mathcal{V}_f = \{f(x) \ : \ x \in \mathbb{F}_p\}$ be the value set of f (that is, the set of inner nodes of $\mathcal{G}_f$). Clearly, $v \in \mathcal{V}_f$ if $v - a$ is quadratic residue modulo p. Since for the sums of Legendre symbols modulo p we have

$$\left| \sum_{v \in \mathbb{F}_p} \left(\frac{(v - a)(-v - a)}{p} \right) \right| = 1$$

(see [22, Theorem 5.48]), we see that there are $p/4 + O(1)$ values of $v \in \mathbb{F}_p$ with $v, -v \in \mathcal{V}_f$. However, because $f(v) = f(-v)$, it is clear that only one value out of v and $-v$ can be a cyclic point. Hence, the number of cyclic points in $\mathcal{G}_f$ for $f(X) = X^2 + a$ with $a \in \mathbb{F}_p^*$ is at most $3p/8 + O(1)$. In particular, we now see from (4.1) that

$$C(p) = 3p/8 + O(1), \qquad \text{for } p \equiv 5 \pmod 8.$$

The smallest number of cyclic points has achieved the value 2 for all tested primes except $p = 3$ and $p = 7$ (for which this is 1).

In Table 4.1, we provide some numerical data for the number of cyclic points taken over all polynomials except for the above two special polynomials. In particular, we give the results for

$$C^*(p) = \max\{c(f) \ : \ f(X) = X^2 + a, \ a \in \mathbb{F}_q \setminus \{0, -2\}\}.$$

We remark using the transformation $\phi_{\lambda,\mu}$ of the proof of Theorem 2.1 one can reduce the study of arbitrary quadratic polynomials to polynomials of the above shape.

Prime p	Min	Max	Average	Expected
500009	2	3578	886.2239149	886.2349015
500029	2	3620	885.9897086	886.2526257
500041	2	3798	885.0688786	886.2632600
500057	2	3468	884.9626481	886.2774389
500069	2	3556	885.8313906	886.2880730
500083	2	3596	884.9700189	886.3004792
500107	2	3527	884.5065536	886.3217460
500111	2	3732	884.3407057	886.3252912
500113	2	3805	885.1602624	886.3270634
500119	2	3873	884.5585953	886.3323802
500153	2	3472	884.8337362	886.3625078
500167	2	3644	884.7563204	886.3749130
1000003	2	5101	1252.451837	1253.316017

Table 4.1. Statistics of the number $c(f)$ of cyclic points for polynomials $X^2 + a$, $a \neq 0, -2$, over $\mathbb{F}_p$ for different primes p

It is quite apparent from Table 4.1 (and from our results for smaller primes) that both the maximum values (that is, $C^*(p)$) and the average values behave regularly and, as we have mentioned, the average value fits the model of a random map quite precisely. We have not attempted to explain the behaviour of $C^*(p)$.

The size of the largest component achieved the largest possible value p in all tested cases (thus, for any p some quadratic polynomial generates a graph with just one connected component, see Table 4.2 below). On the other hand, the smallest achieved size of the largest component does not seem to have a regular behaviour or even monotonicity.

4.4. **Number of components.** On the other hand, the average number of connected components has exhibited a consistent (but slowly decreasing) positive bias of about 9.5% over the predicted value $0.5 \log p$, see [9, Theorem 2 (i)].

For every tested prime, at least one graph $\mathcal{G}_f$ has just 1 component, while the largest number of components has been behaving quite chaotically in all tested ranges.

The above is illustrated in Table 4.2:

Prime p	Min	Max	Average	Expected	Ratio
500009	1	135	7.19772	6.561190689	1.097014298
500029	1	631	7.20138	6.561210688	1.097568778
500041	1	58	7.19640	6.561222687	1.096807766
500057	1	139	7.19259	6.561238685	1.096224409
500069	1	48	7.19785	6.561250684	1.097024081
500083	1	56	7.19328	6.561264682	1.096325228
500107	1	129	7.19792	6.561288677	1.097028397
500111	1	104	7.19801	6.561292676	1.097041445
500113	1	160	7.19402	6.561294676	1.096432999
500119	1	81	7.19518	6.561300675	1.096608791
500153	1	143	7.19312	6.561334665	1.096289150
500167	1	77	7.19699	6.561348661	1.096876629
1000003	1	22	7.54330	6.907756779	1.092004285

TABLE 4.2. Statistics of the number of connected components for polynomials $X^2 + a \in \mathbb{F}_p[X]$ for different primes p

4.5. **Most popular component size.** As we have mentioned, motivated by the complexity bounds of the algorithms of Section 3, we calculated the most popular size of the connected components of $\mathcal{G}_f$. That is, in the notation of Section 3.2 we present the statistics of

$$K_*^{\mathcal{G}} = \max\{k_i^{\mathcal{G}} \ : \ c_i^{\mathcal{G}} = c_*^{\mathcal{G}}, \ 1 \le i \le s^{\mathcal{G}}\}.$$

for functional graphs of quadratic polynomials over finite fields (note that if there are several most popular sizes, then we chose the largest one).

Our results for large primes are given in Table 4.3, where we present some numerical data for

$$K_*(f) = K_*^{\mathcal{G}_f}.$$

Say, for the example of Section 3.2 we have $K_*(f) = 2$. For all tested primes p, the minimal value of the most common size is 1 or 2 (in fact, 2 becomes more common than 1 as p grows), while the largest value is p, as in accordance with Table 4.2, for every p there is always a connected graph $\mathcal{G}_f$. The average value certainly shows a regular growth. However, there does not seem to be any results for this parameter for graphs of random maps, so we have not been able to compare the graphs $\mathcal{G}_f$ with such graphs. Our numerical results seems to suggest that the average of the most common size is proportional to $p^{1/2}$. However, we believe that more numerical experiments are needed before one can confidently formulate any conjectures. One can also consider

$$\kappa_*^{\mathcal{G}} = \min\{k_i^{\mathcal{G}} \ : \ c_i^{\mathcal{G}} = c_*^{\mathcal{G}},\ 1 \le i \le s^{\mathcal{G}}\},$$

that is, the smallest one out most popular sizes.

Prime	Min	Max	Average
500009	1	500009	1689.24
500029	2	500029	1642.27
500041	2	500041	1604.86
500057	1	500057	1670.49
500069	2	500069	1638.32
500083	2	500083	1628.07
500107	2	500107	1635.19
500111	2	500111	1657.12
500113	2	500113	1655.44
500119	2	500119	1573.22
500153	2	500153	1690.84
500167	2	500167	1638.63
1000003	2	1000003	2272.39

TABLE 4.3. Statistics of the number of the values $K_*(f)$ of the most common size of connected components for polynomials $X^2 + a \in \mathbb{F}_p[X]$ for different primes p

Furthermore, we have also computed the number of components of the most popular size (see Table 4.4), that is,

$$c_*(f) = c_*^{\mathcal{G}_f}.$$

Clearly, the minimal value has been 1 for all tested primes (as before, we appeal to Table 4.2 that provied some numerical data for that shows that for every p there is connected graph $\mathcal{G}_f$). However, the largest multiplicity exhibits a surprising chaotic behavior.

The average value clearly converges to a certain constant. However, we made no attempt to conjecture the nature of this constant.

As above with the case of the most common size, this parameter has not been studied and there is no random map model to compare against our results.

Prime	Min	Max	Average
500009	1	75	1.18909
500029	1	465	1.18856
500041	1	18	1.18776
500057	1	104	1.18739
500069	1	18	1.18811
500083	1	24	1.18729
500107	1	56	1.18853
500111	1	40	1.18767
500113	1	80	1.18835
500119	1	24	1.18710
500153	1	108	1.18818
500167	1	54	1.18826
1000003	1	4	1.18843

Table 4.4. Statistics of the number $c_*(f)$ of components of the most common size for polynomials $X^2 + a \in \mathbb{F}_p[X]$ for different primes p

5. Further Directions

It is certainly interesting to study multivariate analogues of our results, that is, to study graphs on q^m vertices, generated by a system of m polynomials in m variables over $\mathbb{F}_q$. It is possible that some results and ideas of [13] can be useful here.

Polynomial graphs over residue rings are also interesting and apparently totally unexplored objects of study. They may also exhibit some new and rather unexpected features.

Finally, we pose an open question of obtaining reasonable approximations to the expected values of the quantities $k_*^{\mathcal{G}}$ and $c_*^{\mathcal{G}}$ for a graph associated with a random map.

Acknowledgement

The authors are very grateful to Domingo Gómez-Pérez and to the referees for many useful comments and suggestions.

The research of S. V. K. was partially supported by Russian Fund for Basic Research, Grant N. 14-01-00332, and Program Supporting Leading Scientific Schools, Grant Nsh-3082.2014.1; that of F. L. by a Marcos Moshinsky fellowship; that of B. M. by Australian Research Council Grants DP110104560 and DP140100118 ; that of M. S. by Australian Research Council Grant DP130100237; and that of I. E. S. by Australian Research Council Grants DP130100237 and DP140100118.

References

[1] E. Bach, 'Toward a theory of Pollard's rho method', *Inform. and Comp.*, **90** (1991), 139–155.

[2] E. Bach and A. Bridy, 'On the number of distinct functional graphs of affine-linear transformations over finite fields', *Linear Algebra Appl.*, **439** (2013), 1312–1320.

[3] R. L. Benedetto, D. Ghioca, B. Hutz, P. Kurlberg, T. Scanlon and T. J. Tucker, 'Periods of rational maps modulo primes', *Math. Ann.*, **355** (2013), 637–660.

[4] R. Brent and P. Zimmerman, *Modern computer arithmetic*, Cambridge Univ. Press, 2010.

[5] W.-S. Chou and I. E. Shparlinski, 'On the cycle structure of repeated exponentiation modulo a prime', *J. Number Theory*, **107** (2004), 345–356.

[6] R. Crandall and C. Pomerance, *Prime numbers: A computational perspective*, 2nd edition, Springer-Verlag, New York, 2005.

[7] J. R. Doyle, X. Faber and D. Krumm, 'Preperiodic points for quadratic polynomials over quadratic fields', *New York J. Math.*, **20** (2014), 507–605.

[8] X. Faber, 'Benedetto's trick and existence of rational preperiodic structures for quadratic polynomials', *Proc. Amer. Math. Soc.*, to appear.

[9] P. Flajolet and A. M. Odlyzko, 'Random mapping statistics', *Lecture Notes in Comput. Sci.*, vol. 434, Springer-Verlag, Berlin, 1990, 329–354.

[10] E. V. Flynn, B. Poonen, and E. F. Schaefer, 'Cycles of quadratic polynomials and rational points on a genus-2 curve', *Duke Math. J.*, **90** (1997), 435–463.

[11] R. Flynn and D. Garton, 'Graph components and dynamics over finite fields', *Intern. J. Number Theory*, **10** (2014), 779–792.

[12] J. B. Friedlander, C. Pomerance and I. E. Shparlinski, 'Period of the power generator and small values of Carmichael's function', *Math. Comp.*, **70** (2001), 1591–1605.

[13] G. Fusco and E. Bach, 'Phase transition of multivariate polynomial systems', *Mathem. Struct. Comp. Sci.*, **19** (2009), 9–23.

[14] T. A. Gassert, 'Chebyshev action on finite fields', *Discr. Math.*, **315–316** (2014), 83–94.

[15] J. E. Hopcroft and J. K. Wong, 'Linear time algorithm for isomorphism of planar graphs (Preliminary Report)', *Proc. 6th Ann. ACM Symp. on Theory of Comp.*, 1974, 172–184.

[16] H. Iwaniec and E. Kowalski, *Analytic number theory*, Amer. Math. Soc., Providence, RI, 2004.

[17] P. J. Kelly, 'A congruence theorem for trees', *Pacific J. Math.*, **7** (1957), 961–968.

[18] D. E. Knuth, *The art of computer programming, vol. I: Fundamental algorithms*, Addison-Wesley, 1968.

[19] D. E. Knuth, *The art of computer programming, vol. II: Seminumerical algorithms*, Addison-Wesley, 1969.

[20] D. E. Knuth, *The art of computer programming, vol. III: Sorting and Searching*, Addison-Wesley, 1973.

[21] P. Kurlberg and C. Pomerance, 'On the period of the linear congruential and power generators', *Acta Arith.*, **119** (2005), 149–169.

[22] R. Lidl and H. Niederreiter, *Finite fields*, Cambridge Univ. Press, Cambridge, 1997.

[23] A. MacFie and D. Panario, 'Random mappings with restricted preimages', *Lecture Notes in Comput. Sci.*, vol. 7533, Springer-Verlag, Berlin, 2012, 254–270.

[24] G. Martin and C. Pomerance, 'The iterated Carmichael λ-function and the number of cycles of the power generator', *Acta Arith.*, **118** (2005), 305–335.

[25] R. C. Mason, 'Equations over function fields', *Lecture Notes in Math.*, vol. 1068, Springer-Verlag, Berlin, 1984, 149–157.

[26] G. L. Miller, 'Graph isomorphism, general remarks', *JCSS*, **2** (1979), 128–142.

[27] P. Morton, 'Arithmetic properties of periodic points of quadratic maps. II', *Acta Arith.*, **87** (1998), 89–102.

[28] P. Morton and J. H. Silverman, 'Rational periodic points of rational functions', *Internat. Math. Res. Notices*, **2** (1994), 97–110.

[29] A. Ostafe and M. Sha, 'Counting dynamical systems over finite fields', *Contemp. Math.*, Amer. Math. Soc., (to appear).

[30] B. Poonen, 'The classification of rational preperiodic points of quadratic polynomials over $\mathbb{Q}$: a refined conjecture', *Math. Zeit.*, **228** (1998), 11–29.

[31] M. Sha and S. Hu, 'Monomial dynamical systems of dimension one over finite fields', *Acta Arith.*, **148** (2011), 309–331.

[32] J. H. Silverman, 'The S-unit equation over function fields', *Proc. Camb. Philos. Soc.*, **95** (1984), 3–4.

[33] L. Somer and M. Křížek, 'The structure of digraphs associated with the congruence $x^k \equiv y \pmod{n}$', *Czechoslovak Math. J.*, **61** (2011), 337–358.

[34] N. Snyder, 'An alternate proof of Mason's theorem', *Elemente Math.*, **55** (2000), 93–94.

[35] W. W. Stothers, 'Polynomial identities and hauptmoduln', *Quart. J. Math.* **32** (1981), 349–370.

[36] T. Vasiga and J. O. Shallit, 'On the iteration of certain quadratic maps over GF(p)', *Discr. Math.*, **277** (2004), 219–240.

[37] A. M. Zubkov and V. E. Tarakanov, 'Cycle structure of power mappings in a residue classes ring', *Discrete Math. Appl.*, **23** (2013), 273–298.

Steklov Mathematical Institute, 8, Gubkin Street, Moscow, 119991, Russia
E-mail address: konyagin@mi.ras.ru

School of Mathematics, University of the Witwatersrand, P. O. Box Wits 2050, South Africa
E-mail address: florian.luca@wits.ac.za

Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
E-mail address: bernard.mans@mq.edu.au

Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
E-mail address: luke.mathieson@mq.edu.au

School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia
E-mail address: shamin2010@gmail.com

School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia
E-mail address: igor.shparlinski@unsw.edu.au