Aalto University
School of Science
Bachelor's Programme in Science and Technology

Detecting multilinear monomials with algebraic fingerprints

Bachelor's Thesis

March 12, 2023

Onni Miettinen

Aalto University
School of Science
Bachelor's Programme in Science and Technology

ABSTRACT OF
BACHELOR'S THESIS

Author:	Onni Miettinen
Title of thesis:	Detecting multilinear monomials with algebraic fingerprints
Date:	March 11, 2023
Pages:	x
Major:	Computer Science
Code:	SCI3027
Supervisor:	Prof. Eero Hyvönen
Instructor:	Augusto Modanese (Department of Computer Science)

abstract to be

Keywords:	key, words, the same as in FIN/SWE
Language:	English

Contents

1 Introduction **4**

1.1 Algebraization of combinatorial problems 5

1.2 Reducing k-3D matching into multilinear monomial detection 6

2 Preliminaries **6**

2.1 Groups, rings and fields 7

2.2 Notation and other terminology 8

3 General multilinear monomial detection **8**

3.1 Non-algebraic methods 9

3.1.1 Dynamic programming for smart expansion of the polynomial . . 9

3.1.2 Non-deterministic color coding for faster evaluation 9

3.2 Non-deterministic color coding with matrices 10

3.3 Algebraic fingerprinting to prevent unwanted cancellation 10

3.4 Limits of general multilinear monomial detection 10

4 Problem specific implementations of algebraic fingerprints **10**

4.1 Fingerprinting for cancellation of non-solutions 11

5 Conclusion **11**

References **12**

1 Introduction

In recent years, there have been rapid advances in the algorithms for combinatorial problems. This has been greatly sparked by the development in algebraic techniques for solving the multilinear monomial detection problem, i.e., finding whether a multivariate polynomial contains a multilinear monomial, first introduced by Koutis in [Kou05], where the set packing problem is reduced to multilinear monomial detection.

Namely, technique of algebraic fingeprinting, first introduced in [Kou08] and further developed in [Wil09], has found great success for many combinatorial problems. With algebraic fingerprinting, the k-path problem, that previously could be solved in $\mathcal{O}^*(4^k)$ time by Chen et al. in [Che+07], could be solved in $\mathcal{O}^*(2^{3k/2})$ time in [Kou08]. This result was quickly improved in [Wil09], where a $\mathcal{O}^*(2^k)$ algorithm was given.

Of course, this technique was further developed, and soon after in [Bjö14] Björklund et al. showed an algorithm that solved the Hamiltonian problem (Hamiltonicity), i.e., finding whether a given graph contains a simple path that visits every vertex, in $\mathcal{O}^*(1.657^n)$ time. Soon enough, for k-path, a $\mathcal{O}^*(1.66^k)$ algorithm was found [Bjö+17]. The fastest algorithms for Hamiltonicity before this ran in $\mathcal{O}^*(2^n)$ and were known since 1962 [HK62], [Bel62]. This was a significant improvement on a problem that had seen no progress in nearly fifty years.

The goal of this thesis are to find out how multilinear monomial detection is relevant in combinatorial problems, how algebraic fingerprints can be utilized to design faster algorithms for problems that use multilinear monomial detection. Also, interesting ideas regarding algebraic fingerprints are explored for.

Multilinear monomial detection is a fundamental problem, since many important combinatorial problems can be reduced into it via a problem specific algebraization. The goal of such algebraization is to form an arithmetic circuit representing a multivariate polynomial that encodes the combinations, i.e. the solutions and non-solutions, into multivariate monomials where multilinear terms correspond to solutions to the given problem.

The technique of algebraic fingerprinting is present in multilinear monomial detection. With algebraic fingerprints, unwanted cancelation due to the characteristic of a field can be prevented. Moreover, algebraic fingerprints can be used to cancel non-solution monomials by abusing the characteristic.

In the next subsections, the thesis discusses algebraization and reduction into multilinear monomial detection. The section 2 covers preliminaries. In section 3, the thesis discusses general multilinear monomial detection. In section 4, some problem specific instances of multilinear monomial detection are given, and clever utilizations of algebraic fingerprints are shown. Section 5 concludes the thesis.

[TODO: go through the structure of the thesis]

1.1 Algebraization of combinatorial problems

A combinatorial problem asks whether a given finite set of objects satisfies some given constraints. For example, the k-path problem asks for, given a finite set of vertices and edges, a path of k vertices. The solutions and non-solutions to combinatorial problems can be thought of as combinations of the given objects. The solution space for the k-path problem consists of combinations of k vertices and k-1 edges. A non-solution combination would contain duplicate vertices or edges that contain vertices outside the combination.

Algebraization is reducing a given problem into an algebraic problem, i.e., a question regarding some algebraic property of some algebraic entity. In an algebraization of a combinatorial problem, the algebraic entity can be constructed from algebraic elements defined from the set of objects given as an input. The motivation behind the construction is some algebraic property that, when satisfied, gives a solution to the problem.

(TODO: rewrite this paragraph, explain with generating polynomial or arithmetic circuit and expansion into sum of monomials) Multilinear monomial detection has proven to be a useful algebraization. First, we introduce multiple variables that correspond to elements from the set of objects given as input. Then, we construct a multivariate polynomial such that it encodes all solutions and non-solutions as multivariate monomials, with solutions encoded specifically as multilinear monomials. Thus, the task of finding a satisfying combination to the combinatorial problem has been reduced to finding a multilinear monomial from the multivariate polynomial. It follows, that a decision problem is answered by the existence of a multilinear monomial and a counting problem by the number of multilinear monomials.

Appropriate definitions for the variables are problem specific. In the following section, this thesis shows a reduction into multilinear monomial detection, introduced in [KW15], for the k-3D matching problem. Another simple example, for the set packing problem, can be found in [Kou05].

1.2 Reducing k-3D matching into multilinear monomial detection

The k-3D matching problem is defined as follows:

k-3D MATCHING

Input: Three disjoint sets A, B and C, and a set of triples $T \subset A \times B \times C$.

Question: Is there a subset $M \subseteq T$, such that $|M| = k$ and $\forall m \in M$: None of the elements in m appear in $M \backslash \{m\}$?

We begin by defining new variables corresponding to the elements in A, B and C, labeled as a_i, b_j and c_k, respectively, where $i \in [|A|]$, $j \in [|B|]$ and $k \in [|C|]$.

For every triple $t \in T$, we define a multilinear monomial x that is a product of the elements in t. We introduce a set X that satisfies the following:

$$\forall x \in X\colon\ x = abc : \ (a, b, c) \in T.$$

Next, we define multivariate polynomials P_1 and P_k as follows:

$$P_1 = \sum_X \ , \ P_k = P_1^k.$$

Following this construction, we observe that P_k, when expanded into a sum of multivariate monomials, contains a multilinear term if and only if the original k-3D matching instance can be answered in the positive. Furthermore, every multilinear monomial in the expanded P_k corresponds to a solution to the problem, and the solutions can be directly found from the variables in the multilinear monomial. Thus, a successful reduction into multilinear monomial detection has been given for the k-3D mapping.

An example instance of k-3D matching with this exact algebraization can be found in [KW15]. [show the example?]

2 Preliminaries

It is necessary to recall basic algebraic concepts before further discussing multilinear monomial detections and algebraic fingerprinting. In this section, definitions for a group, ring and field are given, and some useful concepts regarding them. The second subsection goes through general notation and terminology used throughout the thesis.

2.1 Groups, rings and fields

A group $\mathbf{G}$ is a tuple $(G, +)$, where G is a set of elements, $+ : G \times G \longrightarrow G$ is a binary operation closed under the elements in G, $+$ is associative, every element $g \in G$ has an inverse $g^{-1} \in G$, and G contains an identity element e such that $g + e = g$, $g + g^{-1} = e$ and $e = e^{-1}$. Moreover, $\mathbf{G}$ is called *Abelian* if $+$ is also commutative.

A ring $\mathbf{R}$ is a tuple $(R, \cdot)$, where $R = (G, +)$ is an Abelian group, $\cdot : G \times G \longrightarrow G$ is a binary operation closed under G. We call the binary operations $+$ and $\cdot$ as addition and multiplication, respectively. Note, that from here on we use R as the set of elements defined for $\mathbf{R}$. In general, a bold typeface $\mathbf{X}$ represents a group, ring or field and X its set of elements. R must contain a multiplicative identity $\mathbf{1} \in R$ such that $\forall a \in R\colon\ a \cdot \mathbf{1} = a$. We notate the additive identity e required for the group R as $\mathbf{0}$ from here on. Observe, that for any $\mathbf{R} \neq \{\mathbf{0}\}$, $\mathbf{1} \neq \mathbf{0}$. Left and right distributive laws hold for rings, i.e.,

$$\forall a, b, c \in R\colon\ a \cdot (b + c) = (a \cdot b) + (a \cdot c) \wedge (b + c) \cdot a = (b \cdot a) + (c \cdot a).$$

$u \in R$ is called *unit* if it holds that $\exists v \in R\colon\ u \cdot v = v \cdot u = \mathbf{1}$, i.e., it has a multiplicative inverse $v \in R$.

A field $\mathbf{F} = (F, +, \cdot)$ is defined with the following conditions:

- $(F, +)$ is an Abelian group
- $(F \backslash \{\mathbf{0}\}, \cdot)$ is an Abelian group
- Left and right distributive laws hold for $\mathbf{F}$

Equivalently, a ring is a field if every non-zero element is unit, $\mathbf{1} \neq \mathbf{0}$, and multiplication is commutative. The *characteristic* of a field $\mathbf{F}$ is defined as follows:

$$char(\mathbf{F}) = \begin{cases} min\{n \in \mathbb{N} : n \cdot \mathbf{1} = \mathbf{0}\} & \\ 0 & \text{if such } n \text{ does not exist} \end{cases} \tag{1}$$

Note, that a field $\mathbf{F}$ with characeristic 2 satisfies the following:

$$\forall u \in F\colon\ u + u = u \cdot (\mathbf{1} + \mathbf{1}) = u \cdot \mathbf{0} = \mathbf{0}$$

TODO: polynomial rings, group algebra

2.2 Notation and other terminology

TODO: create a table or like, list terms: multilinearity, multivariety, sum of monomials form & generating form (arithmetic circuit) of polynomial, degree of multivariate monomial, $\mathcal{O}$, Θ, FPT, $\mathcal{O}^*$, determinism & non-determinism (or use 'Monte Carlo' c:)

3 General multilinear monomial detection

The detection of multilinear monomials is a fundamental problem, since many important problems can be reduced to it [TODO: quick examples (just refs?)]. Therefore, any progress in general multilinear monomial detection directly implies faster algorithms for all problems, that are reduced to and solved with general multilinear monomial detection. The general, parameterized multilinear monomial problem is defined as follows:

k-MULTILINEAR MONOMIAL DETECTION

Input: A commutative arithmetic circuit A over a set of variables X representing a polynomial $P(X)$.

Question: Does the polynomial $P(X)$ extended as a sum of monomials contain a multilinear monomial of degree k?

A naive expansion of A into $P(X)$ and evaluation of $P(X)$ is not optimal: an N-degree polynomial will have $2^{\Theta(N)}$ possible monomials, and most problems, that can use this algebraization, have been solved with faster algorithms (TODO: quick example?). This motivates the detection of multilinear monomials without fully expanding the polynomial into a sum of monomials.

In the next section, this thesis quickly overviews dynamic programming and color coding for k-path in the context of k-multilinear monomial detection. Although color coding is outperformed by the purely algebraic methods, color coding is given as valuable background information, since some ideas carry on to the algebraic methods. Moreover, it appears that most parameterized problems that are solved with color coding can be reduced to multilinear monomial detection and, thus, have faster algorithms [ref forgot :p].

In a later section, the thesis covers color coding with matrices as a purely algebraic method, which leads into the technique of algebraic fingerprinting. Then, a general framework for algebraic fingerprinting is discussed. In the last subsection under this section, the thesis discusses some limits and cons in general multilinear monomial detection.

3.1 Non-algebraic methods

Before discussing color coding, the thesis briefly overviews dynamic programming in the context of multilinear monomial detection, since dynamic programming is used in color coding. Moreover, the thesis introduces an algebraically interesting idea of setting squared variables to zero.

3.1.1 Dynamic programming for smart expansion of the polynomial

In multilinear monomial detection, only the multilinear terms are important. This implies that any non-linear term can be discarded as soon as they are formed in the arithmetic circuit, since the arithmetic circuit will never decrease the degree of a monomial. Therefore, a dynamic programming algorithm can be designed for the expansion with an additional rule: any squared variable can be instantly discarded.

In practise, a dynamic programming algorithm ... TODO: explain dynamic programming briefly and how it works with the arithmetic circuit

The rule of discarding squared variables can be written in algebraic form: a squared variable $x^2 \in X$ is set to the additive identity (zero) of the field $\mathbf{F}$, where $X \subset F$. With this additional rule for multilinear monomial detection, the following is deduced: if there are no solutions to the original problem, the polynomial will identically evaluate to zero.

Note, however, that the polynomial evaluating to zero does not imply that no solutions exist. Indeed, we will come across a problem where multilinear monomials cancel each other, and thus evaluate to zero (see Section X.X).

3.1.2 Non-deterministic color coding for faster evaluation

TODO: go over random assignment with colors, which results a polynomial of smaller domain (less variables), thus making the evaluation faster

In [AYZ95], Alon et al. introduces color coding, and applies it in a non-deterministic algorithm for the k-path problem. First, the thesis explains the idea behind the algorithm in [AYZ95]. After this, the idea is translated into multilinear monomial detection for easier relevance.

(DIRECTED OR UNDIRECTED) k-PATH (DECISION AND SEARCH)

Input: A (directed or undirected) graph $G = (V, E)$.

Question: Does G contain a simple path on k vertices? If so, give such a path.

TODO: go quickly over the algorithm, translate it in terms of multilinear monomial detection, give focus on the random assignment in order to reduce domain size (adds non-determinism, though)

TODO: end with hints toward matrix assignment (a non-zero matrix squared can equal zero)

3.2 Non-deterministic color coding with matrices

TODO: go through idea of matrix assignment and specifications for a suitable algebra, then lead to fingerprinting

3.3 Algebraic fingerprinting to prevent unwanted cancellation

TODO: explain how fingerprints prevent cancellation, give the general algebraic framework proposed by Williams and Koutis, (polynomial identity testing)

3.4 Limits of general multilinear monomial detection

TODO: give some cons wrt. color coding (difficult derandomization, are we able to add weights for optimization problems?)

TODO: explain the limit in general multilinear detection with this algebraic framework (impossible to find better algebra than what is used for the current fastest k-mld algorithm)

TODO: lead to problem specific implementations (we can use fingerprinting cleverly, when we understand the underlying problem well)

4 Problem specific implementations of algebraic fingerprints

TODO: add subsections for problem specific implementations with different utilizations of fingerprints

4.1 Fingerprinting for cancellation of non-solutions

TODO: go over Björklund et al. for k-path or Hamiltonicity, managed to design fingerprints such that non-solutions cancel

5 Conclusion

TODO: conclude

References

[Bel62] Richard Bellman. "Dynamic programming treatment of the travelling salesman problem". In: *Journal of the ACM (JACM)* 9.1 (1962), pp. 61–63.

[HK62] Michael Held and Richard M. Karp. "A Dynamic Programming Approach to Sequencing Problems". In: *Journal of the Society for Industrial and Applied Mathematics* 10.1 (1962), pp. 196–210. DOI: 10.1137/0110015. eprint: https://doi.org/10.1137/0110015. URL: https://doi.org/10.1137/0110015.

[AYZ95] Noga Alon, Raphael Yuster, and Uri Zwick. "Color-Coding". In: *J. ACM* 42.4 (July 1995), pp. 844–856. ISSN: 0004-5411. DOI: 10.1145/210332.210337. URL: https://doi.org/10.1145/210332.210337.

[Kou05] Ioannis Koutis. "A faster parameterized algorithm for set packing". In: *Information Processing Letters* 94.1 (2005), pp. 7–9. ISSN: 0020-0190. DOI: https://doi.org/10.1016/j.ipl.2004.12.005. URL: https://www.sciencedirect.com/science/article/pii/S0020019004003655.

[Che+07] Jianer Chen et al. "Improved Algorithms for Path, Matching, and Packing Problems". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 298–307. ISBN: 9780898716245.

[Kou08] Ioannis Koutis. "Faster Algebraic Algorithms for Path and Packing Problems". In: *Automata, Languages and Programming*. Ed. by Luca Aceto et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 575–586. ISBN: 978-3-540-70575-8.

[Wil09] Ryan Williams. "Finding paths of length k in O*(2k) time". In: *Information Processing Letters* 109.6 (2009), pp. 315–318. ISSN: 0020-0190. DOI: https://doi.org/10.1016/j.ipl.2008.11.004. URL: https://www.sciencedirect.com/science/article/pii/S0020019008003396.

[Bjö14] Andreas Björklund. "Determinant Sums for Undirected Hamiltonicity". In: *SIAM Journal on Computing* 43.1 (2014), pp. 280–299. DOI: 10.1137/110839229. eprint: https://doi.org/10.1137/110839229. URL: https://doi.org/10.1137/110839229.

[KW15] Ioannis Koutis and Ryan Williams. "Algebraic Fingerprints for Faster Algorithms". In: *Commun. ACM* 59.1 (Dec. 2015), pp. 98–105. ISSN: 0001-0782. DOI: 10.1145/2742544. URL: https://doi.org/10.1145/2742544.

[Bjö+17] Andreas Björklund et al. "Narrow sieves for parameterized paths and packings". English. In: *Journal of Computer and System Sciences* 87.C (2017), pp. 119–139. DOI: 10.1016/j.jcss.2017.03.003.