Analysis of a reconstruction method based on multisets and backtracking algorithm – Midterm Report

1st Yixuan Zhang
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
120010058@link.cuhk.edu.cn

***Abstract*—Polymer-based data storage provides people with a landmark method in storing data with greater capacity, higher density, and longer life. Compared with current data storage techniques, like hard-disk or Redundant Array of Independent Disks(RAID), it shows outstanding storing functions. However, the accuracy of the binary data after recovering from polymers is sometimes relatively too low for data storage stability. The process of recovery will lead to mutation of data read. Transforming polymer information into binary data requires techniques in coding theory, supporting people to do reconstruction and error correction. The cases of error can be divided into three kinds: single error, multi errors under asymmetric case, and multi errors under symmetric case. In this article, a method based on polynomial designed to solve multiple error-correcting reconstruction in symmetric case is analyzed.**

I. Introduction

Current digital storage systems are facing numerous problems like conducting computations based on memory [1]. Scientists have come up with a number of molecular storage paradigms [2]- [4]. As a potential storing technique, polymer data storage has been tried to take the place of hardware data storage. However, there are still some obstacles of polymer data storage that prevent it from being in common usage. One critical obstacle is the decoding error during the process of decoding from polymer to binary strings. Among the earlier work, the problem of *binary string reconstruction from its substring composition multiset* is addressed in [5]. The method discussed here is for error-correcting reconstruction in symmetric case which leverages a polynomial formulation of the composition reconstruction problem first described in [5]. The method is improved in [6] to deal with t symmetric composition errors. Our main goal is to analyze the method based on polynomials and try to do some improvements on its efficiency. We take the basic logic of the method, indicating how it works to do error correction successfully. Theorem 7 from [6] is chosen to be proved, showing a key process of subsequent derivations of the method.

II. Problem Statement

Among all error correction codes and methods, polynomial can be an interesting and more vivid one. By transforming composition multisets of strings and their prefixes, one is able to correct multiple composition errors that are in symmetric case. One theorem useful for subsequent derivation needs to be proved and the process of the proof will be shown in final report.

Let $\mathbb{F}_q$ be a finite field of order q, where q is a odd prime. Let $\alpha \in \mathbb{F}_q$ be a primitive element of the field. For a polynomial $f(x) \in \mathbb{F}_q[x]$, let $\mathcal{R}(f)$ denote the set of its roots.

***Theorem** 1 (thm1): ([18, Ch. 5])Assume that E(x) $\in \mathbb{F}_q[x]$ has at most t nonzero coefficients. Then, E(x) can be uniquely determined in $\mathcal{O}(n^2)$ time given $(\alpha^t), E(\alpha^{t-1}), ..., E(\alpha^{(} - 1), E(\alpha^0, E(\alpha^1), ..., E(\alpha^{-t+1}), E(\alpha^{-t})$.*

Further, if there are some new idea to do some improvements in the polynomial error correcting method in [6].

III. Method and Algorithm

A. Polynomial-based error correction

In order to correct t symmetric composition errors while conducting binary strings reconstructions, we introduce a polynomial-based method for further usage. The method defines compositions of strings and their prefixes and extends to composition multisets which are constructed by compositions. Each component in the composition represents the weight of 0 and 1 of substring with exact length n.

For a string $\mathbf{s} \in \{0,1\}^n$, let $P_{\mathbf{s}}(x,y)$ be a bivariate polynomial of degree *n* with coefficients in $\{0,1\}$ such that contains exactly one term with total degree $i \in \{0,1,...,n\}$. If $\mathbf{s} = s_1..s_n$ and if $(P_s(x,y)_i$ denotes the unique term of total degree *i*, then $P_s(x,y)_0 = 1$, and

$$(P_s(x,y)_i) = \begin{cases} y\ (P_{\mathbf{s}}(x,y))_{i-1}, if\ s_i = 0 \\ x\ (P_{\mathbf{s}}(x,y))_{i-1}, if\ s_i = 1. \end{cases}$$

Here we use *x* to denote 1 and *y* to denote 0 in composition multisets of substrings. We summarize prefixes from length 1 to *n*-1 and construct composition, then transforming it into polynomials. For example, for $\mathbf{s} = 0010$ we have $P_{\mathbf{s}}(x,y) = 1+y+y^2+xy^2+xy^3$ where it generates all prefixes of string **s**. To prove the equation, we start with free coefficient 1 and expand to prefix with different lengths. By adding *y*, prefix with length 1 is expressed, adding y^2 to express prefix with length 2, adding xy^2 to express prefix with length 3, adding

xy^3 to express prefix with length 4. Then the relation between left and right hand side can be proved.

Except for prefixes, we also use $S_{\mathbf{s}}(x,y)$ to express composition multisets of the string **s**, with similar logic as above. As an example, for $\mathbf{s} = 0010$ we have

$$C(\mathbf{s}) = \{0, 0, 1, 0, 0^2, 01, 01, 0^21, 0^21, 0^31\}$$

and

$$S_{\mathbf{s}}(x,y) = x + 3y + 2xy + y^2 + 2xy^2 + xy^3,$$

where the first two terms of $S_{\mathbf{s}}(x,y)$ indicate that it contains one substring 1 and three substrings 0. The following three terms indicate that it contains two substrings constructed by one 0 and one 1, one substring constructed by two 0s, two substrings constructed by one 1 and two 0s. For longer strings and their composition multisets, similar logic could be used and following terms could be interpreted similarly.

B. Algorithm Induction

From [5], we introduce Equation 1

$$P_s(x,y)P_s(\frac{1}{x},\frac{1}{y}) = (n+1) + S_s(x,y) + S_s(\frac{1}{x},\frac{1}{y}) \quad (1)$$

Proof. For each component except 1 in $P_s(x,y)$ we call $x^i y^{n-i}$ for string **s** with length i, there is paired component in $P_s(\frac{1}{x},\frac{1}{y})$ represented as $x^{-i}y^{i-n}$. From $P_s(x,y)P_s(\frac{1}{x},\frac{1}{y})$, we can gain n 1s from n pairs of variables above if there is a string **s** with length n. Divide P_s into the n^{th} term $P_s(n)$ and remain components which we use $P_s(n-)$ to represent components prior to the n^{th} term and $P_s(n+)$ to represent components that are behind the n^{th} term. The multiplication would be $1 + S_s(x,y) + S_s(\frac{1}{x},\frac{1}{y})$. 1 from 1*1, $S_s(x,y)$ from $\sum P_s(x,y)(n) * P_s(\frac{1}{x},\frac{1}{y})(n-)$, $S_s(\frac{1}{x},\frac{1}{y})$ from $\sum P_s(x,y)(n) * P_s(\frac{1}{x},\frac{1}{y})(n+)$. Further we conclude that $\sum P_s(x,y)(n) * P_s(\frac{1}{x},\frac{1}{y})(n-) = \sum P_s(\frac{1}{x},\frac{1}{y})(n) * P_s(x,y)(n+)$. Then the equation holds.

We use a square to represent the pairing result of a string $s = 0010$, where we have $P_s(x,y) = 1 + y + y^2 + xy^2 + xy^3$ and $P_s(\frac{1}{x},\frac{1}{y}) = 1 + \frac{1}{y} + \frac{1}{y^2} + \frac{1}{xy^2} + \frac{1}{xy^3}$,

$$\mathbf{M}_5 = \begin{bmatrix} 1 & y & y^2 & xy^2 & xy^3 \\ \frac{1}{y} & 1 & y & xy & xy^2 \\ \frac{1}{y^2} & \frac{1}{y} & 1 & x & xy \\ \frac{1}{xy^2} & \frac{1}{xy} & \frac{1}{x} & 1 & y \\ \frac{1}{xy^3} & \frac{1}{xy^2} & \frac{1}{xy} & \frac{1}{y} & 1 \end{bmatrix} \quad (2)$$

In the matrix, we can find out that the matrix is divided into two parts by diagonal lines constructed by 1 from up-left to down-right. Since each string **s** with length n will have $P_s(x,y)$ and $P_s(\frac{1}{x},\frac{1}{y})$ with lengths equal to n+1, the $(n+1)*(n+1)$ matrix have diagonals with length n+1. The sum of components in the diagonal is n+1 which matches (n+1) at the right-hand side of the equation. The upper part of the rest of the matrix represents $P_s(x,y)$ and the lower part represents $P_s(\frac{1}{x},\frac{1}{y})$.

For any given bivariate polynomial $f(x,y)$, denote $f^*(x,y)$ to be its reciprocal polynomial which defined as

$$f^*(x,y) = x^{deg_x(f)}y^{deg_y(f)}f(\tfrac{1}{x},\tfrac{1}{y})$$

, where $deg_x(f)$ denotes the x-degree of $f(x,y)$ and $deg_y(f)$ denotes the y-degree of $f(x,y)$. Thus we can rewrite (1) as:

$$P_s(x,y)P_s^*(x,y) = x^{d_x}y^{d_y}(n+1+S_s(x,y)) + S_s^*(x,y). \quad (3)$$

Given $\widetilde{C}(s)$ representing composition multiset with t symmetric errors different from $C(s)$, we use $S_s(x,y)$ to denote the polynomial form of $C(s)$ and $\widetilde{S}_s(x,y)$ to denote $\widetilde{C}(s)$. We use $E(x,y)$ to denote the difference between $\widetilde{C}(s)$ and $S_s(x,y)$, represented as:

$$\widetilde{S}_s(x,y) = S_s(x,y) + E(x,y) \quad (4)$$

where the range of $E(x,y)$ is 2t symmetric errors with at most 2t nonzero coefficients. Here we represent the range as $\{-t, -t+1, -t+2, ..., -1, 0, 1, 2, ..., t-1, t\}$

The above are some basic concepts and equations which construct the logic of transforming composition multisets into polynomial forms and representing the difference when there are both multiset $C(s)$ and corresponding terms with at most t symmetric errors $\widetilde{C}(s)$. The knowledge might help prove Theorem 1.

IV. Preliminary Result

At the first stage, I understand the logic of how to construct composition multisets based on given strings in [6]. I analyze the backtracking algorithm utilized in solving single error case and how it works when there are multi asymmetric errors. Further I use matrix to prove (1) for subsequent derivation. The method is implemented to a single example, a string with length 4. It should be expanded to a general form instead of solving single example.

V. Future Work

The elemental stage of my future work is to gain the proof of Theorem **??**. It is worthy to discuss the root of finite field with specific order. The properties of the primitive elements contained in the finite field should contribute to how to denote *E(x)* based on given conditions. What should be the meaning of the exponents of α among the elements in the given condition. The exponents follows the range of d_x, recalling it as x-degree of $f(x,y)$, showing at most 2t nonzero coefficients of $E(x,y)$.

Acknowledgment

The work is based on a backtracking algorithm in [5] designed to solve single error case string reconstruction. However, the polynomial method used to solve symmetric multi errors is based on number theory, mostly Reed-Solomon Codes which is ubiquitous in practice.

REFERENCES

[1] V. Zhirov, R. M. Zadegan, G. S. Sandhu, G. M. Church, and W. L. Hughes, "Nucleic acid memory," Nature materials, vol. 15, no. 4, p. 366, 2016.

[2] A. Al Ouahabi, J.-A. Amalian, L. Charles, and J.-F. Lutz, "Mass spectrometry sequencing of long digital polymers facilitated by programmed inter-byte fragmentation," Nature communications, vol. 8, no. 1, p. 967, 2017.

[3] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized dna," Nature, vol. 494, no. 7435, p. 77, 2013.

[4] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," Angewandte Chemie International Edition, vol. 54, no. 8, pp. 2552–2555, 2015.

[5] J.Acharya,H.Das,O.Milenkovic,A.Orlitsky,andS.Pan,"String reconstruction from substring compositions," SIAM Journal on Discrete Mathematics, vol. 29, no. 3, pp. 1340–1371, 2015.

[6] S. Pattabiraman, R. Gabrys and O. Milenkovic, "Coding for Polymer-Based Data Storage," IEEE Transactions on Information Theory, vol. 69, no. 8, pp. 4812-4836, Aug. 2023.

[7] R. Roth, Introduction to coding theory. Cambridge University Press, 2006.