Insertion and Deletion Correction in Polymer-Based Data Storage

Anisha Banerjee, *Student Member, IEEE*, Antonia Wachter-Zeh, *Senior Member, IEEE*, and Eitan Yaakobi, *Senior Member, IEEE*

***Abstract*— Synthetic polymer-based data storage seems to be a particularly promising candidate that could help to cope with the ever-increasing demand for archival storage requirements. It involves designing molecules of distinct masses to represent the respective bits $\{0, 1\}$, followed by the synthesis of a polymer of molecular units that reflects the order of bits in the information string. Reading out the stored data requires the use of a tandem mass spectrometer, that fragments the polymer into shorter substrings and provides their corresponding masses, from which the *composition*, i.e. the number of 1s and 0s in the concerned substring can be inferred. Prior works have dealt with the problem of unique string reconstruction from the set of all possible compositions, called *composition multiset*. This was accomplished either by determining which string lengths always allow unique reconstruction, or by formulating coding constraints to facilitate the same for all string lengths. Additionally, error-correcting schemes to deal with substitution errors caused by imprecise fragmentation during the readout process, have also been suggested. This work builds on this research by extending previously considered error models, mainly confined to substitution of compositions. To this end, we define new error models that consider insertions of spurious compositions and deletions of existing ones, thereby corrupting the composition multiset. We analyze if the reconstruction codebook proposed by Pattabiraman et al. is indeed robust to such errors, and if not, propose new coding constraints to remedy this.**

***Index Terms*— Polymer-based data storage, string reconstruction, composition errors, insertions, deletions.**

I. Introduction

As WE progress through this digital age, our rate of data generation continues to rise unhindered, and with it, so do our storage requirements. Since current data storage media are not particularly advantageous in regard to longevity or density, several molecular storage techniques [2], [3], [4], [5], [6], [7], [8], [9] have been proposed. The work in [2] involving synthetic polymer-based data storage systems appears to be especially favorable, given its promise of efficient synthesis, low read latency and cost. Under this paradigm, a string of information bits is encoded into a chain of molecules linked by means of phosphate bonds, such that the component molecules may only assume one of two significantly differing masses, which represent the bits 0 and 1 respectively. The stored data can be read out by employing a tandem mass (MS/MS) spectrometer, which essentially splits the synthesized polymer at the phosphate linkages and outputs the masses of the resulting fragments. In this manner, the user is given access to the masses of all substrings in the encoded string.

A previous work [10] dealt with the problem of reconstructing a binary string from such an MS/MS readout, under the following modeling assumptions:

Assumption 1: Masses of the component molecules are chosen such that one can always uniquely infer the *composition*, i.e., the number of 0s and 1s forming a certain fragment, from its mass.

Assumption 2: While fragmenting a polymer for the purpose of mass spectrometry analysis, the masses of all constituent substrings are observed with identical frequency.

This proposed setting simplifies the recovery of the original information string into the problem of binary string reconstruction from its composition multiset. More specifically, the reconstruction process now involves determining the binary string from a set of compositions of all of its substrings of each possible length. It is worth noting that this setup does not allow for differentiation between a string and its reversal, since their sets of substring compositions would be identical.

While the authors of [10] primarily focused on string lengths that ensured unique reconstruction from a composition multiset, subsequent works [11], [12], [13] extended this research by building a code that allows for unique reconstruction of each member codeword from its composition multiset alone, regardless of the string length. It was found that a redundancy proportional to the logarithm of the information length is sufficient to guarantee unique reconstruction. The authors also considered the problem of correcting potential errors in composition multisets. In particular, the error models involved substitution of one or more compositions in the composition multiset of a string. Such errors were treated under the ‘asymmetric’ and ‘symmetric’ settings. In the former case, the occurrence of a substitution error in one of the compositions of a certain

Manuscript received 15 March 2022; revised 9 February 2023; accepted 12 February 2023. Date of publication 3 March 2023; date of current version 16 June 2023. This work was supported by the European Research Council (ERC) under the European Union’s Horizon 2020 Research and Innovation Programme under Grant 801434. An earlier version of this paper was presented at the 2022 IEEE International Symposium on Information Theory (ISIT) [DOI: 10.1109/ISIT50566.2022.9834352]. *(Corresponding author: Anisha Banerjee.)*

Anisha Banerjee and Antonia Wachter-Zeh are with the Institute for Communications Engineering, Technical University of Munich, 80333 Munich, Germany (e-mail: anisha.banerjee@tum.de; antonia.wachter-zeh@tum.de).

Eitan Yaakobi is with the Computer Science Department, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: yaakobi@cs.technion.ac.il).

Communicated by X. Zhang, Associate Editor for Coding and Decoding.

Digital Object Identifier 10.1109/TIT.2023.3252045

length, say k, should not accompanied by another substitution of a composition of length $n-k+1$. The opposite is true for the symmetric case. A more general setting for composition substitutions was also considered. Suitable coding constraints to cope with such errors in the composition multiset were proposed. Most notably, it was found that a redundancy of $\mathcal{O}(\log n + t)$ is sufficient to correct t composition substitutions in a composition multiset. The work in [14] takes a step further by dealing with the recovery of multiple strings from the mass spectrometry readout of a mixture of synthesized polymers. In order to provide a brief overview of the best known constructions that allow unique string reconstruction from composition multisets, error-free or otherwise, we have mentioned a few constructions from [11], [12], and [13] in Table I.

Since the errors introduced during an MS/MS readout are often context-dependent, we devote this work to the extension of the error model considered in [11] and [12]. Specifically, we investigate the impact of inserting and deleting one or more compositions on the reconstructability of the encoded strings. This may be motivated by possible shortcomings of Assumption 2, due to which some compositions may escape the readout process. While this situation is equivalent to the deletion of a few compositions, we instead consider a more severe error model, i.e., deletions of complete multisets. This is done on account of the fact that all compositions in a multiset are not equally valuable to the reconstruction algorithm (more details in Section II-B), and most of them can be inferred directly from other multisets corresponding to greater substring lengths. As a consequence, the deletion of a few compositions from a single multiset is equivalent to the deletion of a complete multiset, under certain circumstances. Moreover, having a stronger code capable of correcting such a severe error allows for fast reconstruction.

For these reasons, we focus on error models that cause the deletions of complete multisets, under asymmetric and symmetric settings, similar to how substitution errors were categorized in [11]. We also propose new coding constraints to enable the correction of such errors. Specifically, we derive that redundancies of $\mathcal{O}(\log n + t)$ and $\mathcal{O}(t \log n)$ are sufficient to correct the deletion of t asymmetric multisets and t symmetric multisets, respectively.

We also establish an equivalence between codes that correct composition insertions and composition deletions. A special kind of substitution error, namely a *skewed substitution error* is also studied. This category of errors is motivated by imperfect fragmentations of a given polymer during the MS/MS readout process, as a result of which the observed molecular mass of a shorter monomer chain is lower than what the true mass of its perfectly fragmented version would have been. In this scenario, errors occur only in one direction, i.e., the measured mass can only be lower than the true mass, not higher. We prove that a code that corrects the asymmetric deletion of t multisets, can also correct t-asymmetric skewed composition substitutions. A summary of the constructions proposed in this work has been included in Table I.

The organization of this work is as follows. Section II introduces relevant terminology, notations and some preliminary results to be exploited subsequently. The error models considered previously, are also briefly discussed, while Section III describes the error models pertaining to insertions, deletions and skewed substitutions of one or multiple compositions and also briefly summarizes error-correcting codes to deal with the same. We demonstrate the equivalence between codes correcting deletions and insertions of multisets in Section IV. Sections V and VI delve deeper into the constructions capable of correcting deletions of multiple multisets. We also talk about skewed substitution errors and related coding constructions in Section VII. Finally, we conclude with Section VIII, where a few open problems are discussed.

II. Preliminaries

Let $\boldsymbol{s} = s_1 s_2 \ldots s_n$ denote a binary string of n bits. Any substring $s_i \ldots s_j$ where $i \leq j$, may be indicated by $\boldsymbol{s}_i^j$. The *composition* of this substring, denoted by $c(\boldsymbol{s}_i^j)$, is said to be $0^z 1^w$, where z and w refer to the number of 0s and 1s in $\boldsymbol{s}_i^j$ respectively, such that $z+w = j-i+1$. We also define $C_k(\boldsymbol{s})$ as the multiset of compositions of all length-k substrings in $\boldsymbol{s}$. Evidently, $C_k(\boldsymbol{s})$ should contain $n-k+1$ compositions.

Example 1: Consider $\boldsymbol{s} = 001010111$. Then, the multiset of compositions for substrings of length 7 is given by: $C_7(\boldsymbol{s}) = \{0^4 1^3, 0^3 1^4, 0^2 1^5\}$.

Upon combining the multisets for all $1 \leq k \leq n$, we obtain the *composition multiset* of $\boldsymbol{s}$:

$$C(\boldsymbol{s}) = \bigcup_{k \in [n]} C_k(\boldsymbol{s}).$$

where $[n] = \{1, \ldots, n\}$. As stated earlier, [10] determined string lengths for which unique reconstruction (up to reversal) from such sets is possible. For the remaining string lengths, the authors exploited a bivariate generating polynomial representation, to find strings that are equicomposable with a given string. Here, two distinct strings $\boldsymbol{s}, \boldsymbol{t} \in \{0,1\}^n$ are said to be *equicomposable* if a common composition multiset is shared, i.e., $C(\boldsymbol{s}) = C(\boldsymbol{t})$.

A code $\mathcal{C}$ is called a *composition-reconstructable code* if for all $\boldsymbol{s}, \boldsymbol{t} \in \mathcal{C}$, it holds that $C(\boldsymbol{s}) \neq C(\boldsymbol{t})$. For all n, denote by $A(n)$ the size of the largest composition reconstructable code. Since composition multisets are identical for a binary string and its reversal, it holds that

$$A(n) \leq 2^{\lceil \frac{n}{2} \rceil} + \frac{1}{2}(2^n - 2^{\lceil \frac{n}{2} \rceil}) = 2^{n-1} + 2^{\lceil \frac{n}{2} \rceil - 1},$$

where the term $2^{\lceil \frac{n}{2} \rceil}$ describes the number of palindromic strings of length n, and [10] determined string lengths n where it is possible to achieve this bound with equality. Specifically, it was shown that binary strings of length ≤ 7, one less than a prime, or one less than twice a prime, are uniquely reconstructable up to reversal. It is worth pointing out that a better bound for $A(n)$ is not known.

A. Unique Reconstruction Codes

For values of n where it is not possible to achieve the aforementioned bound, it is necessary to formulate a code, as done in [11] and [12].

The first major coding-theoretic problem concerning polymer-based data storage involved designing constraints in

TABLE I
SUMMARY OF CONSTRUCTIONS

Code	Symbol	Upper bound on redundancy	Proof
Composition-reconstructable code	$\mathcal{S}_R(n)$	$\frac{1}{2}\log_2 n + 5$	[11, 12]
Single composition substitution-correcting code		$\frac{1}{2}\log_2(n-2) + 8$	[11, 12]
t composition substitution-correcting code		$\mathcal{O}(\log n + t)$	[11, 13]
t-asymmetric multiset deletion code	$\mathcal{S}_{DA}^{(t)}(n)$	$\frac{1}{2}\log_2(n-2t-2) + 2t + 7$	Th. 2
2-symmetric multiset deletion code	$\mathcal{S}_{DS}^{(2)}(n)$	$\frac{1}{2}\log_2 n + 8$	Th. 4
t-symmetric multiset deletion code	$\mathcal{S}_{DS}^{(t)}(n)$	$3t\log_2 n + 4t + \frac{1}{2}\log_2(n-4t-2)$ $-\log_2(t+1) + 6$	Th. 5
t-asymmetric skewed composition code		$\frac{1}{2}\log_2(n-2t-2) + 2t + 7$	Th. 2 + Th. 6

order to guarantee unique reconstruction for codewords of a fixed length, i.e., to formulate a composition-reconstructable code. To this end, [12] introduced the following composition-reconstructable code for even codeword lengths.

Construction 1 [12]:

$$\begin{aligned}\mathcal{S}_R(n) = \{\boldsymbol{s} \in \{0,1\}^n : s_1 = 0, s_n = 1, \text{ and}\\ \exists I \subset \{2, \ldots, n-1\} \text{ such that}\\ \text{for all } i \in I, s_i \neq s_{n+1-i},\\ \text{for all } i \notin I, s_i = s_{n+1-i},\\ \boldsymbol{s}_{[n/2]\cap I} \text{ is a Catalan-Bertrand string.}\}\end{aligned} \quad (1)$$

In this context, a Catalan-Bertrand string refers to any binary vector wherein each prefix contains strictly more 0s than 1s. When n is odd, the codebook $\mathcal{S}_R(n)$ is defined as:

$$\mathcal{S}_R(n) = \bigcup_{\boldsymbol{s}\in\mathcal{S}_R(n-1)} \{\boldsymbol{s}_1^{(n-1)/2}\ 0\ \boldsymbol{s}_{(n+1)/2}^n,\ \ \boldsymbol{s}_1^{(n-1)/2}\ 1\ \boldsymbol{s}_{(n+1)/2}^n\}. \quad (2)$$

The number of redundant bits can thus be upper-bounded in terms of n as $1/2\log(n) + 5$ [11]. Alternatively, [12] also states the following.

Theorem 1: [12, pg. 3] There exist efficiently encodable and decodable reconstruction codes with k information bits and redundancy at most $\frac{1}{2}\log(k) + 6$.

From the definition of $A(n)$, we can also deduce that,

$$|\mathcal{S}_R(n)| \leq A(n).$$

For each $\boldsymbol{s} \in \mathcal{S}_R(n)$, this construction sets $s_1 = 0$ and $s_n = 1$ to avoid confusion among reversals, while the remaining bits are chosen such that the weights of a prefix and a suffix of equal length are unequal if the said prefix includes a Catalan-Bertrand string, i.e.,

$$\text{wt}(\boldsymbol{s}_2^i) \begin{cases} = \text{wt}(\boldsymbol{s}_{n-i+1}^{n-1}), & \text{if } [i] \cap I = \emptyset,\\ < \text{wt}(\boldsymbol{s}_{n-i+1}^{n-1}), & \text{otherwise,}\end{cases} \quad (3)$$

where $i < \lceil \frac{n}{2} \rceil$, $\text{wt}(\cdot)$ denotes the Hamming weight of the argument and I is defined as in (1). The latter inequality stems from the fact that if $\boldsymbol{s}_{[i]\cap I}$ has strictly more 0s than 1s, then $\boldsymbol{s}_{\{n-i+1,\ldots,n-1\}\cap I}$ contains strictly more 1s than 0s, thus causing a weight mismatch. Here, we note that the embedded Catalan-Bertrand string may begin from index 2 at the earliest.

B. Reconstruction from Error-Free Composition Multisets

The decoder of the composition-reconstructable code $\mathcal{S}_R(n)$ recovers a string from its composition multiset by employing the approach outlined in [10] and [11]. Since the underlying principles of this process help us in formulating coding constructions for the newer error models involving insertions and deletions, we briefly discuss it in this subsection. For further details, the reader is referred to [10] and [11].

The algorithm begins by deducing the following sequence that characterizes the string to be recovered, say $\boldsymbol{s} \in \mathcal{S}_R(n)$,

$$\boldsymbol{\sigma}_s = (\sigma_1, \ldots, \sigma_{\lceil n/2 \rceil}),$$

where $\sigma_i = \text{wt}(s_i s_{n-i+1})$ for $i \in \{1, \ldots, \lfloor n/2 \rfloor\}$. When n is odd, we set $\sigma_{\lceil \frac{n}{2} \rceil} = \text{wt}(s_{\lceil \frac{n}{2} \rceil})$, i.e., the weight of the central element.

Example 2: For $\boldsymbol{s} = 001010111$. the sequence of σ_i's is $\boldsymbol{\sigma}_s = (1, 1, 2, 0, 1)$.

These values can be computed by exploiting some inherent properties of composition multisets. In particular, we make use of *cumulative weights*, which are defined for each multiset $C_k(\boldsymbol{s})$ as:

$$w_k(\boldsymbol{s}) = \sum_{0^z1^w \in C_k(\boldsymbol{s})} w.$$

Example 3: For instance, the multiset $C_7(\boldsymbol{s}) = \{0^41^3, 0^31^4, 0^21^5\}$ has a cumulative weight $w_7(\boldsymbol{s}) = 12$.

It is easy to see that for all $k \leq \lceil \frac{n}{2} \rceil$, these weights obey the following relations.

$$w_1(\boldsymbol{s}) = \sum_{i=1}^{\lceil \frac{n}{2} \rceil} \sigma_i, \tag{4}$$

$$w_k(\boldsymbol{s}) = \sum_{i=1}^{k} i\sigma_i + k \sum_{i=k+1}^{\lceil n/2 \rceil} \sigma_i \tag{5}$$

$$= kw_1(\boldsymbol{s}) - \sum_{i=1}^{k-1} i\sigma_{k-i}. \tag{6}$$

We also observe a symmetry relation for any given set of cumulative weights:

$$w_k(\boldsymbol{s}) = w_{n-k+1}(\boldsymbol{s}), \quad \forall\, k \in [n]. \tag{7}$$

In light of this, the multisets C_i and C_{n-i+1} are henceforth said to be *symmetric*. For notational convenience, we also define

$$\widetilde{C}_i(\boldsymbol{s}) = C_i(\boldsymbol{s}) \cup C_{n-i+1}(\boldsymbol{s}).$$

Now to demonstrate how the reconstruction algorithm functions, we consider the following example.

Example 4: In this example, we reconstruct the string $\boldsymbol{s} = 001010111$ from its composition multiset $C(\boldsymbol{s})$, which is stated below:

$$\begin{aligned} C(\boldsymbol{s}) = \{&0, 0, 1, 0, 1, 0, 1, 1, 1, 0^2, 0^11^1, 0^11^1, 0^11^1, 0^11^1, \\ &0^11^1, 1^2, 1^2, 0^21^1, 0^21^1, 0^11^2, 0^21^1, 0^11^2, 0^11^2, \\ &1^3, 0^31^1, 0^21^2, 0^21^2, 0^21^2, 0^11^3, 0^11^3, 0^31^2, \\ &0^31^2, 0^21^3, 0^21^3, 0^11^4, 0^41^2, 0^31^3, 0^21^4, 0^21^4, \\ &0^41^3, 0^31^4, 0^21^5, 0^41^4, 0^31^5, 0^41^5\}. \end{aligned} \tag{8}$$

The reconstruction process involves the following steps.

1) Firstly, we deduce its $\boldsymbol{\sigma}_s$ sequence from (4) and (6):
$$\boldsymbol{\sigma}_s = (1, 1, 2, 0, 1).$$
2) We create a multiset $\mathcal{T}$ to include all compositions that can be determined from $\boldsymbol{\sigma}_s$. More explicitly, one can infer the compositions $c(s_5), c(\boldsymbol{s}_4^6), \ldots, c(\boldsymbol{s}_1^9)$ by noting that for any $i < \lceil n/2 \rceil$,
$$c(s_i s_{n-i+1}) = \begin{cases} 0^2, & \text{if } \sigma_i = 0. \\ 0^11^1, & \text{if } \sigma_i = 1. \\ 1^2, & \text{if } \sigma_i = 2. \end{cases}$$
$$\mathcal{T} = \{1, 0^21, 0^21^3, 0^31^4, 0^41^5\}.$$
3) The process now assigns the bits of $\boldsymbol{s}$ pairwise, in an inward manner, starting with bit pair (s_1, s_9). Since $\sigma_1 = 1$, we could set $s_1 = 0$ and $s_9 = 1$ or vice-versa. Due to (1), we opt for the former, i.e. $(s_1, s_9) = (0, 1)$.
4) Using the reconstructed prefix and suffix, we update $\mathcal{T}$:
$$\mathcal{T} = \{0, 1, 1, 0^21, 0^21^3, 0^31^4, 0^41^5, 0^31^5, 0^41^4\}.$$
5) The two longest compositions in the multiset $C(\boldsymbol{s}) \backslash \mathcal{T}$ are $\{0^41^3, 0^21^5\}$. These denote the compositions of substrings $\boldsymbol{s}_1^7$ and $\boldsymbol{s}_3^9$. Conversely, their complements $\{1^2, 0^2\}$ correspond to compositions $c(\boldsymbol{s}_1^2)$ and $c(\boldsymbol{s}_8^9)$. Combining this with the knowledge of bits s_1 and s_9, we reconstruct $\boldsymbol{s}$ up to its prefix-suffix pair of length 2, i.e. $(\boldsymbol{s}_1^2, \boldsymbol{s}_8^9) = (00, 11)$.
6) To recover the remaining bits, we simply repeat steps 4 and 5.

C. Previous Error Models

We now turn our attention to the problem of reconstruction from erroneous composition multisets. Substitution errors were considered in [11] under the asymmetric and symmetric setting. In this error model, some compositions in $C(\boldsymbol{s})$ are arbitrarily altered. If the errors occur such that each multiset $\widetilde{C}_i$ includes at most one substituted composition, then they are said to be *asymmetric*. On the contrary, a pair of *symmetric* substitution errors would occur in the multisets C_i and C_{n-i+1}, for any $i \in [n]$.

Definition 1: A composition multiset $C(\boldsymbol{s})$ of the string $\boldsymbol{s} \in \{0,1\}^n$ is said to have suffered an **asymmetric substitution error**, if for some $i \in [n]$, a single composition of the multiset $C_i(\boldsymbol{s})$ is modified, but its symmetric counterpart $C_{n-i+1}(\boldsymbol{s})$ remains unaffected.

Definition 2: If a composition multiset $C(\boldsymbol{s})$ is corrupted by having one composition substituted in each of the multisets $C_i(\boldsymbol{s})$ and $C_{n-i+1}(\boldsymbol{s})$ for some $i \in [n]$ such that $2i \neq n+1$, then **two symmetric substitution errors** are said to have occurred.

To exemplify this, we consider the following.

Example 5: Let $\boldsymbol{s} = 001010111$. The symmetric multiset pair $C_3(\boldsymbol{s})$ and $C_7(\boldsymbol{s})$ is given by

$$\begin{aligned} C_3(\boldsymbol{s}) &= \{0^21, 0^21, 01^2, 0^21, 01^2, 01^2, 1^3\}, \\ C_7(\boldsymbol{s}) &= \{0^41^3, 0^31^4, 0^21^5\}. \end{aligned}$$

For instance, an asymmetric substitution error is said to have occurred if $C_7(\boldsymbol{s})$ is corrupted to

$$C_7'(\boldsymbol{s}) = \{0^41^3, 0^31^4, 0^31^4\}.$$

On the contrary, if $C_3(\boldsymbol{s})$ is also corrupted in addition to $C_7(\boldsymbol{s})$ as follows,

$$C_3'(\boldsymbol{s}) = \{1^3, 0^21, 01^2, 0^21, 01^2, 01^2, 1^3\},$$

then two symmetric substitution errors are said to have occurred.

For multiple asymmetric and symmetric substitution errors, the aforementioned definitions are extended as follows.

Definition 3: A composition multiset $C(\boldsymbol{s})$ of the string $\boldsymbol{s} \in \{0,1\}^n$ is said to have suffered t **asymmetric substitution errors**, if for some $\mathcal{I} \in [n]$ where $|\mathcal{I}| = t$, a single composition in each of the multisets $C_i(\boldsymbol{s})$ for $i \in \mathcal{I}$, is modified, but the symmetric counterpart of this multiset, i.e., $C_{n-i+1}(\boldsymbol{s})$, remains unaffected.

Definition 4: If a composition multiset $C(\boldsymbol{s})$ is corrupted by having one composition substituted in each of the multisets $C_i(\boldsymbol{s})$ and $C_{n-i+1}(\boldsymbol{s})$ for some $i \in \mathcal{I} \subset \left[\left\lfloor \frac{n}{2} \right\rfloor\right]$, where $|\mathcal{I}| = t$, then t pairs of **symmetric substitution errors** are said to have occurred.

Besides introducing two distinct constructions aimed at correcting asymmetric and symmetric substitutions respectively, the authors of [11] and [13] also investigated a more natural error setting for composition substitutions and proposed a code capable of correcting t substitutions in a composition multiset. Formally, a code is said to be a t *composition substitution-correcting code*, if for any $\boldsymbol{s}, \boldsymbol{v}$ that belong to the code, it holds that $|C(\boldsymbol{s}) \backslash C(\boldsymbol{v})| > 2t$. A known instantiation is stated in Table I.

III. New Error Models

The subsequent sections explore error models that involve corrupting a valid composition multiset via the insertion or deletion of one or more multisets.

Definition 5: An **asymmetric multiset deletion** is said to have occurred in the composition multiset $C(\boldsymbol{s})$ of a string $\boldsymbol{s} \in \{0,1\}^n$, if for some $i \in [n]$, the multiset $C_i(\boldsymbol{s})$ is entirely missing, while $C_{n-i+1}(\boldsymbol{s})$ is uncorrupted.

Definition 6: A **pair of symmetric multiset deletions** is said to have occurred if the composition multiset $C(\boldsymbol{s})$ of a string $\boldsymbol{s} \in \{0,1\}^n$, if for some $i \in [n]$ such that $i \neq n-i+1$, the multisets $C_i(\boldsymbol{s})$ and $C_{n-i+1}(\boldsymbol{s})$ are entirely eliminated.

Example 6: Let $\boldsymbol{s} = 001010111$. If the composition multiset $C(\boldsymbol{s})$ is corrupted to

$$\begin{aligned}C'(\boldsymbol{s}) = &\bigcup_{i \in [n] \backslash \{3\}} C_i(\boldsymbol{s}),\\ = &\{0,0,1,0,1,0,1,1,1,0^2,0^1 1^1,0^1 1^1,0^1 1^1,0^1 1^1,\\ &0^1 1^1,1^2,1^2,0^3 1^1,0^2 1^2,0^2 1^2,0^2 1^2,0^1 1^3,0^1 1^3,\\ &0^3 1^2,0^3 1^2,0^2 1^3,0^2 1^3,0^1 1^4,0^4 1^2,0^3 1^3,0^2 1^4,\\ &0^2 1^4,0^4 1^3,0^3 1^4,0^2 1^5,0^4 1^4,0^3 1^5,0^4 1^5\}.\end{aligned}$$

then an asymmetric multiset deletion is said to have occurred. More specifically, the multiset $C_3(\boldsymbol{s}) = \{0^2 1^1, 0^2 1^1, 0^1 1^2, 0^2 1^1, 0^1 1^2, 0^1 1^2, 1^3\}$ has been deleted. On the other hand, if

$$\begin{aligned}C'(\boldsymbol{s}) = &\bigcup_{i \in [n] \backslash \{3,7\}} C_i(\boldsymbol{s}),\\ = &\{0,0,1,0,1,0,1,1,1,0^2,0^1 1^1,0^1 1^1,0^1 1^1,0^1 1^1,\\ &0^1 1^1,1^2,1^2,0^3 1^1,0^2 1^2,0^2 1^2,0^2 1^2,0^1 1^3,0^1 1^3,\\ &0^3 1^2,0^3 1^2,0^2 1^3,0^2 1^3,0^1 1^4,0^4 1^2,0^3 1^3,0^2 1^4,\\ &0^2 1^4,0^4 1^4,0^3 1^5,0^4 1^5\}.\end{aligned}$$

we say that a pair of symmetric multiset deletions has occurred. Here compared to $C(\boldsymbol{s})$, we are missing the multisets $C_3(\boldsymbol{s}) = \{0^2 1^1, 0^2 1^1, 0^1 1^2, 0^2 1^1, 0^1 1^2, 0^1 1^2, 1^3\}$ and $C_7(\boldsymbol{s}) = \{0^4 1^3, 0^3 1^4, 0^2 1^5\}$.

Definition 7: A composition multiset $C(\boldsymbol{s})$ of a string $\boldsymbol{s} \in \{0,1\}^n$ is said to have suffered a **composition insertion error**, if for some $i \in [n]$ the multiset $C_i(\boldsymbol{s})$ is replaced by $C_i'(\boldsymbol{s})$, such that $C_i(\boldsymbol{s}) \subset C_i'(\boldsymbol{s})$ and $|C_i'(\boldsymbol{s})| = n - i + 2$, i.e. an unknown and invalid composition has been registered.

Example 7: Once again, let $\boldsymbol{s} = 001010111$. If $C_7(\boldsymbol{s})$ has been altered as follows,

$$C_7'(\boldsymbol{s}) = \{0^4 1^3, 0^3 1^4, 0^2 1^5, 0^1 1^6\}.$$

we say that a composition insertion error has taken place.

The main contribution of this work consists of studying the aforementioned error models and proposing new coding constraints to combat the same. We also establish an equivalence between codes that correct composition insertions and composition deletions. Consequently, we restrict our attention to the latter for the remainder of this paper.

To this end, we first propose the following composition reconstruction code that allows for the correction of t asymmetric multiset deletions. Specifically, a code is termed as a *t-asymmetric multiset deletion code*, if for any $\boldsymbol{s}$, $\boldsymbol{v}$ that belong to this code, there exists no $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| \leq t$ such that,

$$\begin{aligned}C_i(\boldsymbol{s}) &\neq C_i(\boldsymbol{v}),\\ C_{n-i+1}(\boldsymbol{s}) &= C_{n-i+1}(\boldsymbol{v}) \quad \forall\, i \in \mathcal{I},\\ C_j(\boldsymbol{s}) &= C_j(\boldsymbol{v}) \quad \forall\, j \in [n] \setminus \mathcal{I}.\end{aligned}$$

We suggest the following instantiation for a t-asymmetric multiset deletion code.

Construction 2:

$$\begin{aligned}\mathcal{S}_{DA}^{(t)}(n) = &\{\boldsymbol{s} \in \{0,1\}^n : s_1 = 0, s_n = 1, \text{ and}\\ &\exists\, \mathcal{I} \subset \{2, \ldots, \frac{n}{2}\},\ |\mathcal{I}| \geq t, \text{ such that}\\ &\forall\, i \in \mathcal{I},\ s_i \neq s_{n+1-i}, \text{ and } \forall\, i \notin \mathcal{I},\ s_i = s_{n+1-i},\\ &(s_{\frac{n}{2}}, s_{\frac{n}{2}+1}) \neq (1,0),\\ &\boldsymbol{s}_{[n/2] \cap \mathcal{I}} \text{ is a string wherein each}\\ &\text{prefix has at least } t \text{ more 0s than 1s.}\} \qquad (9)\end{aligned}$$

The corresponding proof follows behind Theorem 2. Evidently, this construction is inspired from $\mathcal{S}_R^{(t)}(n)$ [11], in that it requires at least t 0s in $\boldsymbol{s}_1^{n/2}$ and at least t 1s in $\boldsymbol{s}_{n/2+1}^n$, however their locations are not necessarily restricted as in $\mathcal{S}_R^{(t)}(n)$. The extension to odd codeword lengths is similar to (2).

Following this, we investigate the case of symmetric multiset deletions, and discover that when two or more symmetric multiset pairs are missing, additional constraints are needed to bolster the code $S_R(n)$ so as to guarantee unique reconstructability. In this context, a code is termed as a *t-symmetric multiset deletion code*, if for any $\boldsymbol{s}$, $\boldsymbol{v}$ that belong to this code, there exists no $\mathcal{I} \subseteq \left[\left\lceil \frac{n}{2} \right\rceil\right]$ with $|\mathcal{I}| \leq t$ such that

$$\begin{aligned}\widetilde{C}_i(\boldsymbol{s}) &\neq \widetilde{C}_i(\boldsymbol{v}), \forall\, i \in \mathcal{I}\\ C_i(\boldsymbol{s}) &= C_i(\boldsymbol{v}) \quad \forall\, i \in \left[\left\lceil \frac{n}{2} \right\rceil\right] \setminus \mathcal{I}.\end{aligned}$$

For the elementary case of two deleted symmetric multiset pairs, we propose the following code.

Construction 3:

$$\begin{aligned}\mathcal{S}_{DS}^{(2)}(n) =&\{\boldsymbol{s} \in \mathcal{S}_R(n) : \\ &\sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(\boldsymbol{s}) \bmod 7 = a,\ a \in \{0, 1, \ldots, 6\}\}. \end{aligned} \tag{10}$$

Theorem 4 proves that this code can indeed correct the deletion of two symmetric multiset pairs. We also generalize this construction to accommodate for the deletion of any t symmetric multiset pairs, where $t \geq 2$.

Construction 4:

$$\begin{aligned}\mathcal{S}_{DS}^{(t)}(n) =&\{\boldsymbol{s} \in \{0,1\}^n : s_1 = 0, s_n = 1, \\ &\exists\, \mathcal{I} \subset \left\{2, \ldots, \frac{n}{2} - t - 1\right\},\ |\mathcal{I}| \geq t \text{ where} \\ &\forall i \in \mathcal{I}, s_i \neq s_{n-i+1},\ \text{and } \forall i \notin \mathcal{I}, s_i = s_{n-i+1}, \\ &\boldsymbol{s}_{[\frac{n}{2}]\cap\mathcal{I}} \text{is a string where each prefix} \\ &\text{has at least } t \text{ more 0s than 1s}, \\ &\boldsymbol{\sigma}_s \in 3t\text{-erasure-correcting code}, \\ &\forall i \in \left\{\frac{n}{2} - t, \ldots \frac{n}{2}\right\}, (s_i, s_{n-i+1}) \neq (1, 0).\} \end{aligned} \tag{11}$$

where $t \geq 2$, n is even and $n > 6t$. For odd values of n, a similar construction exists. Evidently, $\mathcal{S}_{DS}^{(t)}(n) \subset \mathcal{S}_{DA}^{(t)}(n)$.

Theorem 5 proves that $\mathcal{S}_{DS}^{(t)}(n)$ is capable of correcting the deletion of t symmetric multiset pairs.

Definition 8: A composition multiset $C(\boldsymbol{s})$ of the string $\boldsymbol{s} \in \{0,1\}^n$ is said to have suffered an **asymmetric skewed substitution error**, if for some $i \in [n]$, a single composition of multiset $C_i(\boldsymbol{s})$ is replaced with one of a lower Hamming weight, such that the symmetric counterpart $C_{n-i+1}(\boldsymbol{s})$ remains unaffected.

Example 8: For instance, if an erroneous measurement corrupts the composition $0^2 1^4$, the measured compositions could be $0^3 1^3$ or $0^4 1^2$, but not $0^1 1^5$.

To formally define a *t-asymmetric skewed composition code*, we first define the *t-asymmetric skewed error ball* of a string $\boldsymbol{s} \in \{0,1\}^n$ as

$$\begin{aligned}B_t(\boldsymbol{s}) =&\{C'(\boldsymbol{s}) = \bigcup_{i \in [n]} C'_i(\boldsymbol{s}) : \mathcal{I} \subseteq [n],\ |\mathcal{I}| \leq t, \\ &|C_i(\boldsymbol{s}) \backslash C'_i(\boldsymbol{s})| = 1, \sum_{\substack{0^{i-w}1^w \\ \in C'_i(\boldsymbol{s})}} w < \sum_{\substack{0^{i-w}1^w \\ \in C_i(\boldsymbol{s})}} w \quad \forall\, i \in \mathcal{I}, \\ &C'_j(\boldsymbol{s}) = C_j(\boldsymbol{s})\ \ \forall\, j \in [n] \backslash \mathcal{I}\}.\end{aligned}$$

We can now define a code as a *t-asymmetric skewed composition code*, if for any $\boldsymbol{s}, \boldsymbol{v}$ that belong to this code, it holds that $B_t(\boldsymbol{s}) \cap B_t(\boldsymbol{v}) = \emptyset$.

We subsequently prove in Lemma 6 of Section VII that the code $\mathcal{S}_{DA}^{(t)}(n)$ (Construction 2) can correct t skewed asymmetric substitution errors in its composition multiset.

These results, along with some of the earlier constructions proposed in [11], [12], and [13], have been summarized in Table I.

IV. Code Equivalence: Insertion and Deletion of Multisets

In this section, we demonstrate how codes which can correct the deletion of a group of t multisets, can also correct the occurrence of insertion errors in those t multisets.

Lemma 1: A code can correct the deletion of t composition multisets, if and only if it can correct any number of composition insertion errors in those t multisets.

Proof: We prove this by contradiction. Let there be two binary strings $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_R(n)$, such that

$$D_t(\boldsymbol{s}) \cap D_t(\boldsymbol{v}) \neq \emptyset. \tag{12}$$

where $D_t(\boldsymbol{s})$ is termed as the t-multiset deletion ball of $\boldsymbol{s}$, and constitutes all possible versions of the composition multiset of s, or $C(\boldsymbol{s})$, after corruption by the deletion of up to any t multisets. To put it more explicitly,

$$D_t(\boldsymbol{s}) = \{C'(\boldsymbol{s}) = \bigcup_{i \in [n] \backslash \mathcal{I}} C_i(\boldsymbol{s}) : \mathcal{I} \subseteq [n], |\mathcal{I}| \leq t\}. \tag{13}$$

Similarly, we also define a t-multiset insertion ball $I_t(\boldsymbol{s})$, as the set of all corrupted compositions multisets of $\boldsymbol{s}$, that are formed from $C(\boldsymbol{s})$ through insertion of compositions in up to t multisets, i.e.,

$$\begin{aligned}I_t(\boldsymbol{s}) =&\{C'(\boldsymbol{s}) = \bigcup_{i \in [n]} C'_i(\boldsymbol{s}) : \mathcal{I} \subseteq [n],\ |\mathcal{I}| \leq t, \\ &C_i(\boldsymbol{s}) \subset C'_i(\boldsymbol{s})\ \forall i \in \mathcal{I}, C'_i(\boldsymbol{s}) = C_i(\boldsymbol{s})\ \ \forall i \in [n] \backslash \mathcal{I}\}.\end{aligned}$$

Equation (12) implies that at least $n - t$ composition multisets of $\boldsymbol{s}$ and $\boldsymbol{v}$ are identical. In other words, when a specific group of t multisets disappears from the multiset information of $\boldsymbol{s}$ and $\boldsymbol{v}$, they become indistinguishable. Let these differing multisets correspond to substring lengths $i_1, i_2, \ldots i_t$. This allows us to write that:

$$\bigcup_{j \in [n] \backslash \{i_1, \ldots i_t\}} C_j(\boldsymbol{s}) = \bigcup_{j \in [n] \backslash \{i_1, \ldots i_t\}} C_j(\boldsymbol{v}).$$

If we perform a set union operation on both sides of the previous equation with $\bigcup_{i \in \{i_1, \ldots i_t\}} C_i(\boldsymbol{s}) \cup C_i(\boldsymbol{v})$, then we get:

$$\begin{aligned}&\bigcup_{i \in \{i_1, \ldots i_t\}} (C_j(\boldsymbol{v}) \backslash C_j(\boldsymbol{s})) \cup \bigcup_{j \in [n]} C_j(\boldsymbol{s}) \\ =&\bigcup_{i \in \{i_1, \ldots i_t\}} (C_j(\boldsymbol{s}) \backslash C_j(\boldsymbol{v})) \cup \bigcup_{j \in [n]} C_j(\boldsymbol{v}).\end{aligned}$$

This effectively means that if the multisets $C_{i_1}(\boldsymbol{s}), \ldots C_{i_t}(\boldsymbol{s})$ are corrupted by the insertion of some specific erroneous compositions, then the multiset information may correspond to both $\boldsymbol{s}$ and $\boldsymbol{v}$, and vice-versa. This lets us write that

$$I_t(\boldsymbol{s}) \cap I_t(\boldsymbol{v}) \neq \emptyset.$$

□

Owing to this result, we deem it sufficient to focus on multiset deletion-correcting codes. The subsequent sections examine how multiset deletions affect the reconstructability of an encoded string drawn from $\mathcal{S}_R(n)$. Similar to [11], we categorize such deletion errors into two major settings.

V. Asymmetric Multiset Deletion Codes

We begin by considering an error model where a complete multiset $C_k(\boldsymbol{s})$ can be deleted from the composition multiset $C(\boldsymbol{s})$. This is formally referred to as a single asymmetric multiset deletion [see Definition 5]. We investigate whether the composition-reconstructable code [see Construction 1] guarantees unique recoverability under this model. To proceed in this direction, we first introduce a definition that is relevant to the results that follow.

Definition 9: Consider any two binary strings $\boldsymbol{s}, \boldsymbol{v} \in \{0,1\}^n$. The length i-prefix-suffix pair of $\boldsymbol{s}$ is the vector $(\boldsymbol{s}_1^i, \boldsymbol{s}_{n-i+1}^n)$, where $1 \leq i \leq \lceil \frac{n}{2} \rceil$. The longest prefix-suffix pair shared by $\boldsymbol{s}$ and $\boldsymbol{v}$ is said to be of length l, where $1 \leq l < \lceil \frac{n}{2} \rceil$, if $(\boldsymbol{s}_1^l, \boldsymbol{s}_{n-l+1}^n) = (\boldsymbol{v}_1^l, \boldsymbol{v}_{n-l+1}^n)$, and $(s_{l+1}, s_{n-l}) \neq (v_{l+1}, v_{n-l})$.

Next, we consider the following lemma, which results from a specific case of [11, Lemma 4].

Lemma 2: Let $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_R(m)$ share the same $\boldsymbol{\sigma}$ sequence and satisfy $|C_j(\boldsymbol{s}) \backslash C_j(\boldsymbol{v})| \leq 2$ for all $j \in [m]$. If the longest prefix-suffix pair shared by $\boldsymbol{s}$ and $\boldsymbol{v}$ is of length i, then their corresponding composition multisets C_{m-i-1} and C_{m-i-2} each differ in at least 2 compositions.

Example 9: To shortly highlight the implications of this lemma, we consider the strings $\boldsymbol{s} = 001011101$ and $\boldsymbol{v} = 001110101$. Clearly, they are both specified by $\boldsymbol{\sigma} = (1,0,2,1,1)$. Since the longest prefix-suffix pair shared by them is $(001, 101)$, i.e., of length 3, their respective multisets C_4 and C_5 differ by at least 2 compositions.

Lemma 3: $\mathcal{S}_R(n)$ is a single asymmetric multiset deletion code.

Proof: To prove the statement of the lemma, we intend to establish that for any $\boldsymbol{s} \in \mathcal{S}_R(n)$, if one is given $C'(\boldsymbol{s}) = \bigcup_{i \in [n] \backslash \{k\}} C_i(\boldsymbol{s})$ where $k \in [n]$, then $\boldsymbol{s}$ can be fully recovered.

Case 1: n is even

From the steps of the reconstruction algorithm as described in Section II-B, it is evident that we only require the composition multisets $C_n(\boldsymbol{s}), \ldots, C_{\frac{n}{2}}(\boldsymbol{s})$. Hence, if $k < \frac{n}{2}$, the reconstruction of $\boldsymbol{s}$ is straightforward. On the contrary, if $k \geq \frac{n}{2}$, one can still infer the cumulative weight of the missing multiset $C_k(\boldsymbol{s})$ from (7). Consequently, $\boldsymbol{\sigma}_s$ can be obtained accurately.

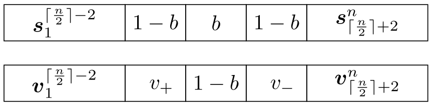
In the absence of $C_k(\boldsymbol{s})$, the prefix and suffix can be constructed up to $\boldsymbol{s}_1^{n-k-1}$ and $\boldsymbol{s}_{k+2}^n$. When $\sigma_{n-k+1} \in \{0,2\}$, there remains no ambiguity concerning the bits s_{n-k} and s_{k+1}. However, when $\sigma_{n-k+1} = 1$, one can either have $(s_{n-k}, s_{k+1}) = (0,1)$ or $(s_{n-k}, s_{k+1}) = (1,0)$ if both of these possibilities guarantee weight mismatch between $\boldsymbol{s}_1^{n-k}$ and $\boldsymbol{s}_{k+1}^n$, as mandated by the constraints in (1). Now since $\boldsymbol{s} \in \mathcal{S}_R(n)$, Lemma 2 tells us that choosing the bits s_{n-k} and s_{k+1} incorrectly, will lead to an incompatibility with the multiset $C_{k-1}(\boldsymbol{s})$. Thus, there exists only one valid choice for these bits, implying that $\boldsymbol{s}$ is uniquely recoverable.

Case 2: n is odd

Similar to the previous case, it can be argued that for any missing composition multiset $C_k(\boldsymbol{s})$, where $k \neq \lceil \frac{n}{2} \rceil$, $\boldsymbol{s}$ can be easily and uniquely determined. The more interesting case occurs when $k = \lceil \frac{n}{2} \rceil$, since the absence of $C_{\lceil \frac{n}{2} \rceil}(\boldsymbol{s})$, and thus $w_{\lceil \frac{n}{2} \rceil}(\boldsymbol{s})$, prevents us from computing $\sigma_{\lceil \frac{n}{2} \rceil - 1}$ and $\sigma_{\lceil \frac{n}{2} \rceil}$. However, their sum is known from (4), i.e.

$$\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} = w_1(\boldsymbol{s}) - \sum_{i=1}^{\lceil \frac{n}{2} \rceil - 2} \sigma_i. \tag{14}$$

Since $\sigma_{\lceil \frac{n}{2} \rceil - 1} = \text{wt}(s_{\lceil \frac{n}{2} \rceil - 1} s_{\lceil \frac{n}{2} \rceil + 1}) \in \{0,1,2\}$ and $\sigma_{\lceil \frac{n}{2} \rceil} = \text{wt}(s_{\lceil \frac{n}{2} \rceil}) \in \{0,1\}$, these values can be inferred directly when $\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} \in \{0,3\}$. However, an ambiguity arises when $\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} \in \{1,2\}$.



$\boldsymbol{s}_1^{\lceil \frac{n}{2} \rceil - 2}$	$1-b$	b	$1-b$	$\boldsymbol{s}_{\lceil \frac{n}{2} \rceil + 2}^n$
$\boldsymbol{v}_1^{\lceil \frac{n}{2} \rceil - 2}$	v_+	$1-b$	v_-	$\boldsymbol{v}_{\lceil \frac{n}{2} \rceil + 2}^n$

Fig. 1. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are such that $(\boldsymbol{s}_1^{\lceil \frac{n}{2} \rceil - 2}, \boldsymbol{s}_{\lceil \frac{n}{2} \rceil + 2}^n) = (\boldsymbol{v}_1^{\lceil \frac{n}{2} \rceil - 2}, \boldsymbol{v}_{\lceil \frac{n}{2} \rceil + 2}^n)$, where $v_+ = 1 - v_-$.

Let $\boldsymbol{v} \in \mathcal{S}_R(n)$ be a string with which $\boldsymbol{s}$ becomes equicomposable when the multiset $C_{\lceil n/2 \rceil}$ is deleted, i.e.,

$$\bigcup_{i \in [n] \backslash \{\lceil \frac{n}{2} \rceil\}} C_i(\boldsymbol{s}) = \bigcup_{i \in [n] \backslash \{\lceil \frac{n}{2} \rceil\}} C_i(\boldsymbol{v}). \tag{15}$$

Also, let $\boldsymbol{v}$ be specified by $\boldsymbol{\sigma}_{\boldsymbol{v}} = (\sigma'_1, \ldots, \sigma'_{\lceil n/2 \rceil})$. As a consequence of (15), we can write:

$$\begin{aligned} \sigma_i &= \sigma'_i, \quad \forall \ 1 \leq i \leq \left\lceil \frac{n}{2} \right\rceil - 2 \\ \sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} &= \sigma'_{\lceil \frac{n}{2} \rceil - 1} + \sigma'_{\lceil \frac{n}{2} \rceil}. \end{aligned} \tag{16}$$

To verify whether the reconstructability of $\boldsymbol{s}$ is affected, we simply check if there exists a suitable $\boldsymbol{v}$ that satisfies (15) and (16). We also note that (15) directly implies the equality of the prefix-suffix pairs $(\boldsymbol{s}_1^{\lceil \frac{n}{2} \rceil - 2}, \boldsymbol{s}_{\lceil \frac{n}{2} \rceil + 2}^n) = (\boldsymbol{v}_1^{\lceil \frac{n}{2} \rceil - 2}, \boldsymbol{v}_{\lceil \frac{n}{2} \rceil + 2}^n)$.

We jointly depict the specific subcases in Fig. 1, wherein we allow for $\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} \in \{1,2\}$ since for both $\boldsymbol{s}$ and $\boldsymbol{v}$, we have:

$$\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} = 2 - b.$$

where $b \in \{0,1\}$. To proceed with the proof, we try to determine the conditions under which $C_{\lceil \frac{n}{2} \rceil - 1}(\boldsymbol{s}) = C_{\lceil \frac{n}{2} \rceil - 1}(\boldsymbol{v})$ holds. This would require the following set equality:

$$\left\{ \begin{matrix} \{c(\boldsymbol{s}_1^{\lceil \frac{n}{2} \rceil - 2}), 1-b\}, \\ \{c(\boldsymbol{s}_2^{\lceil \frac{n}{2} \rceil - 2}), b, 1-b\}, \\ \{c(\boldsymbol{s}_{\lceil \frac{n}{2} \rceil + 2}^n), 1-b\}, \\ \{c(\boldsymbol{s}_{\lceil \frac{n}{2} \rceil + 2}^{n-1}), b, 1-b\} \end{matrix} \right\} = \left\{ \begin{matrix} \{c(\boldsymbol{s}_1^{\lceil \frac{n}{2} \rceil - 2}), v_+\}, \\ \{c(\boldsymbol{s}_2^{\lceil \frac{n}{2} \rceil - 2}), v_+, 1-b\}, \\ \{c(\boldsymbol{s}_{\lceil \frac{n}{2} \rceil + 2}^n), 1-v_+\}, \\ \{c(\boldsymbol{s}_{\lceil \frac{n}{2} \rceil + 2}^{n-1}), 1-v_+, 1-b\} \end{matrix} \right\}.$$

By checking the above relation exhaustively for all possibilities of $(b, v_+) \in \{0,1\}^2$, we conclude that the multisets $C_{\lceil n/2 \rceil - 1}(\boldsymbol{s})$ and $C_{\lceil n/2 \rceil - 1}(\boldsymbol{s})$ can never match. Therefore, $\boldsymbol{v}$ does not exist and $\boldsymbol{s}$ retains its unique reconstructability.

□

As a second step, $\mathcal{S}_R(n)$ is now generalized to $\mathcal{S}_{DA}^{(t)}(n)$ [see Construction 2] to allow correcting the deletion of t

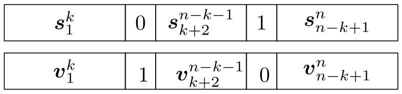


Fig. 2. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are related such that $(\boldsymbol{s}_1^k, \boldsymbol{s}_{n-k+1}^n) = (\boldsymbol{v}_1^k, \boldsymbol{v}_{n-k+1}^n)$ and $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$.

asymmetric multisets. To prove why this construction works, we first make the following observation.

Remark 1: Consider any $\boldsymbol{s} \in \{0,1\}^n$, whose composition multiset $C(\boldsymbol{s})$ suffers t asymmetric multiset deletions and is thereby modified to some $C'(\boldsymbol{s}) = \bigcup_{i \in [n] \setminus \mathcal{I}} C_i(\boldsymbol{s})$, where $\mathcal{I} = \{i_1, i_2, \dots i_{|\mathcal{I}|}\}$ such that for all $j \in [|\mathcal{I}|]$, it holds that $n - i_j + 1 \notin \mathcal{I}$. Since at least one multiset from each possible symmetric multiset pair is preserved, the cumulative weights $w_1(\boldsymbol{s}), w_2(\boldsymbol{s}), \dots, w_{\lceil \frac{n}{2} \rceil}(\boldsymbol{s})$ can be uniquely determined by virtue of (7), and in turn, $\boldsymbol{\sigma}_s$ can be uniquely recovered through (4) and (6).

Lemma 4: For even n, let $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_{DA}^{(t)}(n)$ be specified by an identical $\boldsymbol{\sigma}$ sequence, such that the longest prefix-suffix pair shared by them is of length k.[1] Then their corresponding multisets $C_{n-k-1}, \dots, C_{n-k-t-1}$ differ by at least two compositions.

Proof: Given that $(\boldsymbol{s}_1^k, \boldsymbol{s}_{n-k+1}^n) = (\boldsymbol{v}_1^k, \boldsymbol{v}_{n-k+1}^n)$, $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$ and $s_{k+1} \neq v_{k+1}$, it must hold that $\sigma_{k+1} = 1$. Additionally, as a consequence of $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$, we can infer that $c(\boldsymbol{s}_{k+2}^{n-k-1}) = c(\boldsymbol{v}_{k+2}^{n-k-1})$. Now, without loss of generality, we assume $(s_{k+1}, v_{k+1}) = (0, 1)$. This situation is depicted in Fig. 2.

The approach of this proof essentially centers around the constraints imposed by $\mathcal{S}_{DA}^{(t)}(n)$, to which $\boldsymbol{s}$ and $\boldsymbol{v}$ belong, and the aforementioned relations between $\boldsymbol{s}$ and $\boldsymbol{v}$. In the following, we seek to establish that when $\boldsymbol{s}$ and $\boldsymbol{v}$ are characterized as in Fig. 2, the mismatch between (s_{k+1}, s_{n-k}) and (v_{k+1}, v_{n-k}) will guarantee that the corresponding multisets $C_{n-k-1}, C_{n-k-2}, \dots, C_{n-k-t-1}$ cannot match for any realization of $\boldsymbol{s}$ and $\boldsymbol{v}$. Most crucially, this is a consequence of the fact that for any $\boldsymbol{u} \in \mathcal{S}_{DA}^{(t)}(n)$ that is characterized by the index set $\mathcal{I} = \{i : i \in [\frac{n}{2}], u_i \neq u_{n-i+1}\}$, each prefix of the subsequence $\boldsymbol{u}_{\mathcal{I}}$ (with length greater than t) contains at least t more 0s than 1s.

To proceed with the proof along this line of reasoning, consider the set of indices $\mathcal{I} = \{i_1, i_2, \dots i_{|\mathcal{I}|}\} \subset \{2, \dots, \frac{n}{2} - 1\}$ for $\boldsymbol{s} \in \mathcal{S}_{DA}^{(t)}(n)$, such that for any $i \in \mathcal{I}$, we have $s_i \neq s_{n-i+1}$, or equivalently, $\sigma_i = 1$, and $i_1 < i_2 < \dots < i_{|\mathcal{I}|}$. Owing to the construction of $\mathcal{S}_{DA}^{(t)}(n)$, it evidently holds that $|\mathcal{I}| \geq t$.

By virtue of the constraints of $\mathcal{S}_{DA}^{(t)}(n)$, we know that $\boldsymbol{v}_{\{2,\dots,k+1\} \cap \mathcal{I}}$ must have at least t more 0s than 1s. Also due to $(v_{k+1}, v_{n-k}) = (1, 0)$, we have $k+1 \in \mathcal{I}$. Thus, it can be deduced that $\boldsymbol{v}_{\{2,\dots,k\} \cap \mathcal{I}}$ must have at least $t+1$ more 0s than 1s. This lead us to

$$(s_{i_1}, s_{n-i_1+1}) = (s_{i_2}, s_{n-i_2+1}) = \dots = (s_{i_{t+1}}, s_{n-i_{t+1}+1}) \\ = (0, 1). \tag{17}$$

Evidently, $i_{t+1} \leq k$. Since $(\boldsymbol{s}_1^k, \boldsymbol{s}_{n-k+1}^n) = (\boldsymbol{v}_1^k, \boldsymbol{v}_{n-k+1}^n)$, we can also write

$$(v_{i_1}, v_{n-i_1+1}) = (v_{i_2}, v_{n-i_2+1}) = \dots = (v_{i_{t+1}}, v_{n-i_{t+1}+1}) \\ = (0, 1). \tag{18}$$

Also it must hold that $k + 1 > i_{t+1} \geq t + 2$, i.e., $k \geq t + 2$, since otherwise, $(v_{k+1}, v_{n-k}) = (1, 0)$ would not be a valid choice according to (18).

Furthermore, we have $k \leq \frac{n}{2} - 2$ since $\frac{n}{2}$ is the maximum length of a prefix-suffix pair of $\boldsymbol{s}$, and $k = \frac{n}{2} - 1$, would lead to $(v_{\frac{n}{2}-1}, v_{\frac{n}{2}+2}) = (1, 0)$, which is prohibited by the construction of $\mathcal{S}_{DA}^{(t)}(n)$.

It is evident that $|C_{n-k-1}(\boldsymbol{s}) \setminus C_{n-k-1}(\boldsymbol{v})| = 2$. For the remainder of this proof, we strive to find if $C_{n-k-j}(\boldsymbol{s}) = C_{n-k-j}(\boldsymbol{v})$[2] might hold for any $2 \leq j \leq t + 1 < k$. When $n - k - j \geq k + 1$, the satisfaction of $C_{n-k-j}(\boldsymbol{s}) = C_{n-k-j}(\boldsymbol{v})$ essentially requires that

$$\left\{ \begin{matrix} \{c(\boldsymbol{s}_1^k), 0, c(\boldsymbol{s}_{k+2}^{n-k-j})\}, \\ \{c(\boldsymbol{s}_2^k), 0, c(\boldsymbol{s}_{k+2}^{n-k-j+1})\}, \\ \vdots \\ \{c(\boldsymbol{s}_j^k), 0, c(\boldsymbol{s}_{k+2}^{n-k-1})\}, \\ \{c(\boldsymbol{s}_{n-k+1}^n), 1, c(\boldsymbol{s}_{k+j+1}^{n-k-1})\}, \\ \{c(\boldsymbol{s}_{n-k+1}^{n-1}), 1, c(\boldsymbol{s}_{k+j}^{n-k-1})\}, \\ \vdots \\ \{c(\boldsymbol{s}_{n-k+1}^{n-j+1}), 1, c(\boldsymbol{s}_{k+2}^{n-k-1})\} \end{matrix} \right\} \\ = \left\{ \begin{matrix} \{c(\boldsymbol{v}_1^k), 1, c(\boldsymbol{v}_{k+2}^{n-k-j})\}, \\ \{c(\boldsymbol{v}_2^k), 1, c(\boldsymbol{v}_{k+2}^{n-k-j+1})\}, \\ \vdots \\ \{c(\boldsymbol{v}_j^k), 1, c(\boldsymbol{v}_{k+2}^{n-k-1})\}, \\ \{c(\boldsymbol{v}_{n-k+1}^n), 0, c(\boldsymbol{v}_{k+j+1}^{n-k-1})\}, \\ \{c(\boldsymbol{v}_{n-k+1}^{n-1}), 0, c(\boldsymbol{v}_{k+j}^{n-k-1})\}, \\ \vdots \\ \{c(\boldsymbol{v}_{n-k+1}^{n-j+1}), 0, c(\boldsymbol{v}_{k+2}^{n-k-1})\} \end{matrix} \right\}, \tag{19}$$

since one can surmise from Fig. 2 that any composition of $\boldsymbol{s}$ in $C_{n-k-j}(\boldsymbol{s})$, say $c(\boldsymbol{s}_p^{n-k-j+p-1})$, such that $p \leq k + 1$ and $n - k - j + p - 1 \geq n - k$, will be identical to $c(\boldsymbol{v}_p^{n-k-j+p-1})$. The proof for the case when $n - k - j \leq k$ runs in a similar fashion, and is thus ignored in this analysis. By setting $a = \mathrm{wt}(\boldsymbol{s}_{j+1}^{n-k}) = \mathrm{wt}(\boldsymbol{v}_{j+1}^{n-k})$[3] and $b = \mathrm{wt}(\boldsymbol{s}_{k+1}^{n-j}) = \mathrm{wt}(\boldsymbol{v}_{k+1}^{n-j})$,[4] and additionally exploiting $(\boldsymbol{s}_1^k, \boldsymbol{s}_{n-k+1}^n) = (\boldsymbol{v}_1^k, \boldsymbol{v}_{n-k+1}^n)$, we can

[1] For details, refer to Example 9.

[2] $|C_{n-k-j}(\boldsymbol{s}) \setminus C_{n-k-j}(\boldsymbol{v})| = 1$ is not a possibility due to $w_{n-k-j}(\boldsymbol{s}) = w_{n-k-j}(\boldsymbol{v})$ which is a consequence of $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$ and (6). Therefore $C_{n-k-j}(\boldsymbol{s}) \neq C_{n-k-j}(\boldsymbol{v})$ automatically implies that $|C_{n-k-j}(\boldsymbol{s}) \setminus C_{n-k-j}(\boldsymbol{v})| \geq 2$.

[3] The expression $\mathrm{wt}(\boldsymbol{s}_{j+1}^{n-k}) = \mathrm{wt}(\boldsymbol{v}_{j+1}^{n-k})$ follows from $\boldsymbol{s}_1^k = \boldsymbol{v}_1^k$ and $c(\boldsymbol{s}_{k+1}^{n-k}) = c(\boldsymbol{v}_{k+1}^{n-k})$.

[4] The expression $\mathrm{wt}(\boldsymbol{s}_{k+1}^{n-j}) = \mathrm{wt}(\boldsymbol{v}_{k+1}^{n-j})$ follows from $\boldsymbol{s}_{n-k+1}^n = \boldsymbol{v}_{n-k+1}^n$ and $c(\boldsymbol{s}_{k+1}^{n-k}) = c(\boldsymbol{v}_{k+1}^{n-k})$.

transform (19) into an equivalent representation in terms of Hamming weights.

$$\left\{\begin{array}{l} a+\mathrm{wt}(\boldsymbol{s}_1^j)-\mathrm{wt}(\boldsymbol{s}_{n-k-j+1}^{n-k-1})-1, \\ a+\mathrm{wt}(\boldsymbol{s}_2^j)-\mathrm{wt}(\boldsymbol{s}_{n-k-j+2}^{n-k-1})-1, \\ \qquad\qquad\vdots \\ a+\mathrm{wt}(\boldsymbol{s}_{j-1}^j)-s_{n-k-1}-1, \\ a+s_j-1, \\ b+\mathrm{wt}(\boldsymbol{s}_{n-j+1}^{n})-\mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j}), \\ b+\mathrm{wt}(\boldsymbol{s}_{n-j+1}^{n-1})-\mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-1}), \\ \qquad\qquad\vdots \\ b+\mathrm{wt}(\boldsymbol{s}_{n-j+1}^{n-j+2})-s_{k+2}, \\ b+s_{n-j+1} \end{array}\right\}$$

$$= \left\{\begin{array}{l} a+\mathrm{wt}(\boldsymbol{s}_1^j)-\mathrm{wt}(\boldsymbol{v}_{n-k-j+1}^{n-k-1}), \\ a+\mathrm{wt}(\boldsymbol{s}_2^j)-\mathrm{wt}(\boldsymbol{v}_{n-k-j+2}^{n-k-1}), \\ \qquad\qquad\vdots \\ a+\mathrm{wt}(\boldsymbol{s}_{j-1}^j)-v_{n-k-1}, \\ a+s_j, \\ b+\mathrm{wt}(\boldsymbol{s}_{n-j+1}^{n})-\mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j})-1, \\ b+\mathrm{wt}(\boldsymbol{s}_{n-j+1}^{n-1})-\mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-1})-1, \\ \qquad\qquad\vdots \\ b+\mathrm{wt}(\boldsymbol{s}_{n-j+1}^{n-j+2})-v_{k+2}-1, \\ b+s_{n-j+1}-1 \end{array}\right\}, \quad (20)$$

Due to (17) and (18), we may deduce that for all $p \in [j]$, it should hold that $s_p \leq s_{n-p+1}$ and $v_p \leq v_{n-p+1}$. In particular, we define a set of indices $\mathcal{K}$ as

$$\mathcal{K} = \{2, \ldots, j\} \setminus \mathcal{I}.$$

This set essentially contains all $p \in [j]$ for which $\sigma_p \in \{0, 2\}$, i.e., $s_p = s_{n-p+1}$. Since $j < k$ and $(\boldsymbol{s}_1^k, \boldsymbol{s}_{n-k+1}^n) = (\boldsymbol{v}_1^k, \boldsymbol{v}_{n-k+1}^n)$, it also holds that for all $p \in \mathcal{K}$, $v_p = v_{n-p+1}$.

Finally, we require the following observations, which follow directly from the constraints of $\mathcal{S}_{DA}^{(t)}(n)$.

$$\begin{aligned} &\mathrm{wt}(\boldsymbol{v}_{n-k}^{n-1}) \geq \mathrm{wt}(\boldsymbol{v}_2^{k+1}) + t, \\ \implies &\mathrm{wt}(\boldsymbol{v}_{n-k+1}^{n}) \geq \mathrm{wt}(\boldsymbol{v}_1^{k}) + t + 2, \\ \implies &\mathrm{wt}(\boldsymbol{s}_{n-k+1}^{n}) \geq \mathrm{wt}(\boldsymbol{s}_1^{k}) + t + 2. \end{aligned}$$

This inequality also leads us to

$$\begin{aligned} b - a &= \mathrm{wt}(\boldsymbol{s}_{k+1}^{n-j}) - \mathrm{wt}(\boldsymbol{s}_{j+1}^{n-k}) \\ &= \mathrm{wt}(\boldsymbol{s}_{k+1}^{n}) - \mathrm{wt}(\boldsymbol{s}_{1}^{n-k}) - \mathrm{wt}(\boldsymbol{s}_{n-j+1}^{n}) + \mathrm{wt}(\boldsymbol{s}_{1}^{j}) \\ &= \mathrm{wt}(\boldsymbol{s}_{n-k+1}^{n}) - \mathrm{wt}(\boldsymbol{s}_{1}^{k}) - (j - |\mathcal{K}|) \\ &\geq t + 2 - j + |\mathcal{K}| \geq |\mathcal{K}| + 1, \end{aligned} \quad (21)$$

where the final inequality follows from $j \leq t + 1$. Next, we introduce the following notations for all $p \in [j]$ to simplify the exposition.

$$\begin{aligned} \gamma_p(\boldsymbol{s}) &= \mathrm{wt}(\boldsymbol{s}_p^{n-k-j+p-1}) \\ &= \begin{cases} a+\mathrm{wt}(\boldsymbol{s}_p^j)-\mathrm{wt}(\boldsymbol{s}_{n-k-j+p}^{n-k-1})-1, & \text{if } p \leq j-1 \\ a+s_j-1, & \text{if } p = j \end{cases} \\ \omega_{n-p+1}(\boldsymbol{s}) &= \mathrm{wt}(\boldsymbol{s}_{k+j+2-p}^{n+1-p}) \\ &= \begin{cases} b+\mathrm{wt}(\boldsymbol{s}_{n-j+1}^{n-p+1})-\mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j+1-p}), & \text{if } p \leq j-1 \\ b+s_{n-j+1}, & \text{if } p = j. \end{cases} \end{aligned}$$

and similarly for $\boldsymbol{v}$. Additionally, let $C^{(1)}(\boldsymbol{s}) = \{\gamma_1(\boldsymbol{s}), \ldots \gamma_j(\boldsymbol{s})\}$ and $C^{(2)}(\boldsymbol{s}) = \{\omega_n(\boldsymbol{s}), \ldots, \omega_{n-j+1}(\boldsymbol{s})\}$. We define $C^{(1)}(\boldsymbol{v})$ and $C^{(2)}(\boldsymbol{v})$ analogously. This lets us rewrite (20) in short, as

$$C^{(1)}(\boldsymbol{s}) \cup C^{(2)}(\boldsymbol{s}) = C^{(1)}(\boldsymbol{v}) \cup C^{(2)}(\boldsymbol{v}).$$

Now since for all $p \in [j] \setminus \mathcal{K}$ we have $s_p = v_p = 0$ and $s_{n-p+1} = v_{n-p+1} = 1$, it is possible to infer the following bounds for any $\alpha \in [j]$,

$$\begin{aligned} \omega_{n-\alpha+1}(\boldsymbol{s}) &\geq b + 1 - |\mathcal{K} \setminus [\alpha - 1]| + \mathrm{wt}(\boldsymbol{s}_{\mathcal{K}\setminus[\alpha-1]}) \\ \omega_{n-\alpha+1}(\boldsymbol{v}) &\geq b - |\mathcal{K} \setminus [\alpha - 1]| + \mathrm{wt}(\boldsymbol{s}_{\mathcal{K}\setminus[\alpha-1]}). \\ \gamma_\alpha(\boldsymbol{s}) &\leq a - 1 + \mathrm{wt}(\boldsymbol{s}_{\mathcal{K}\setminus[\alpha-1]}), \\ \gamma_\alpha(\boldsymbol{v}) &\leq a + \mathrm{wt}(\boldsymbol{s}_{\mathcal{K}\setminus[\alpha-1]}), \end{aligned}$$

From the preceding inequalities along with (21), it is possible to deduce that there exists no $\alpha_1, \alpha_2 \in [j]$, for which $\gamma_{\alpha_1}(\boldsymbol{v}) = \omega_{n-\alpha_2+1}(\boldsymbol{s})$. Similarly, there exists no $\alpha_1, \alpha_2 \in [j]$, for which $\gamma_{\alpha_1}(\boldsymbol{s}) = \omega_{n-\alpha_2+1}(\boldsymbol{v})$. Thus, in order for $C_{n-k-j}(\boldsymbol{s}) = C_{n-k-j}(\boldsymbol{v})$, or alternatively (20) to hold, we require

$$\{\gamma_1(\boldsymbol{s}), \ldots \gamma_j(\boldsymbol{s})\} = \{\gamma_1(\boldsymbol{v}), \ldots \gamma_j(\boldsymbol{v})\}, \quad (22)$$

$$\{\omega_n(\boldsymbol{s}), \ldots, \omega_{n-j+1}(\boldsymbol{s})\} = \{\omega_n(\boldsymbol{v}), \ldots, \omega_{n-j+1}(\boldsymbol{v})\}, \quad (23)$$

or more simply put, $C^{(1)}(\boldsymbol{s}) = C^{(1)}(\boldsymbol{v})$ and $C^{(2)}(\boldsymbol{s}) = C^{(2)}(\boldsymbol{v})$. In the following analysis, we strive to establish that any additional conditions on $\boldsymbol{s}$ and $\boldsymbol{v}$ that allow the satisfaction of (22), prohibit equality in (23).

Case 1: $\boldsymbol{s}_{k+2}^{k+j} = \boldsymbol{v}_{k+2}^{k+j}$.

Observe that on account of $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$, $\boldsymbol{s}_{k+2}^{k+j} = \boldsymbol{v}_{k+2}^{k+j}$ also implies that $\boldsymbol{s}_{n-k-j+1}^{n-k-1} = \boldsymbol{v}_{n-k-j+1}^{n-k-1}$.

We begin by attempting to find any conditions on $\boldsymbol{s}$ that allow (22). To this end, we reiterate (22) more explicitly.

$$C^{(1)}(\boldsymbol{s}) = \left\{\begin{array}{l} a+\mathrm{wt}(\boldsymbol{s}_1^j)-\mathrm{wt}(\boldsymbol{s}_{n-k-j+1}^{n-k-1})-1, \\ a+\mathrm{wt}(\boldsymbol{s}_2^j)-\mathrm{wt}(\boldsymbol{s}_{n-k-j+2}^{n-k-1})-1, \\ \qquad\qquad\vdots \\ a+\mathrm{wt}(\boldsymbol{s}_{j-1}^j)-s_{n-k-1}-1, \\ a+s_j-1 \end{array}\right\}$$

$$= \left\{\begin{array}{l} a+\mathrm{wt}(\boldsymbol{s}_1^j)-\mathrm{wt}(\boldsymbol{v}_{n-k-j+1}^{n-k-1}), \\ a+\mathrm{wt}(\boldsymbol{s}_2^j)-\mathrm{wt}(\boldsymbol{v}_{n-k-j+2}^{n-k-1}), \\ \qquad\qquad\vdots \\ a+\mathrm{wt}(\boldsymbol{s}_{j-1}^j)-v_{n-k-1}, \\ a+s_j \end{array}\right\} = C^{(1)}(\boldsymbol{v}),$$

Next, we try to match the composition of weight $a + s_j$ to any composition in $C^{(1)}(\boldsymbol{s})$. To facilitate this, we let

$$\begin{aligned} \gamma_\alpha(\boldsymbol{s}) &= a + s_j = \gamma_j(\boldsymbol{v}) \\ \implies \text{wt}(\boldsymbol{s}_\alpha^{j-1}) &- \text{wt}(\boldsymbol{s}_{n-k-j+\alpha}^{n-k-1}) = 1 \end{aligned} \tag{24}$$

Furthermore, we assume that there exists no $p \in \{\alpha + 1, \ldots, j-1\}$ for which $\gamma_p(\boldsymbol{s}) = a + s_j$. In other words, for all $p \in \{\alpha+1, \ldots, j-1\}$, we require $\gamma_p(\boldsymbol{s}) \leq a + s_j - 1$. This leads us to infer that

$$\text{wt}(\boldsymbol{s}_p^{j-1}) \leq \text{wt}(\boldsymbol{s}_{n-k-j+p}^{n-k-1}),$$

where $p \in \{\alpha+1, \ldots, j-1\}$. Additionally, due to the fact that we always have $|\gamma_{p+1}(\boldsymbol{s}) - \gamma_p(\boldsymbol{s})| \leq 1$ for any $p \in [j-1]$, we can deduce that $\gamma_{\alpha+1}(\boldsymbol{s}) = a + s_j - 1$. This observation, coupled with (24), gives us $\text{wt}(\boldsymbol{s}_{\alpha+1}^{j-1}) = \text{wt}(\boldsymbol{s}_{n-k-j+\alpha+1}^{n-k-1})$ and $(s_\alpha, s_{n-k-j+\alpha}) = (1, 0)$. These relations in turn cause $\gamma_\alpha(\boldsymbol{v}) = a + s_j + 1$. Trying to match this composition of weight $a + s_j + 1$ to any element in $C^{(1)}(\boldsymbol{s})$ will once again lead to some element in $C^{(2)}(\boldsymbol{s})$ to be equal to $a + s_j + 2$ and so on. Thus, in effect, (22) cannot hold. Alternatively, we say that $C_{n-k-j}(\boldsymbol{s}) \neq C_{n-k-j}(\boldsymbol{v})$ always holds.

Case 2: $\boldsymbol{s}_{k+2}^{k+j} \neq \boldsymbol{v}_{k+2}^{k+j}$.

Earlier, we had defined a set $\mathcal{K}$ as the set of all indices $p \in [j]$ for which $s_p = s_{n-p+1}$. We now denote the elements of this set as $\mathcal{K} = \{m_1, m_2, \ldots, m_{|\mathcal{K}|}\}$, where $m_1 > m_2 > \ldots > m_{|\mathcal{K}|}$. Recalling that for all $p \in [j] \setminus \mathcal{K}$, we have $(s_p, s_{n-p+1}) = (v_p, v_{n-p+1}) = (0, 1)$, we restate $C^{(1)}(\boldsymbol{s})$ more explicitly as follows.

$$\left\{\begin{array}{l}
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|}} - \text{wt}(\boldsymbol{s}_{n-k-j+1}^{n-k-1}) - 1, \\
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|}} - \text{wt}(\boldsymbol{s}_{n-k-j+2}^{n-k-1}) - 1, \\
\qquad\qquad \vdots \\
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|}} - \text{wt}(\boldsymbol{s}_{n-k-j+m_{|\mathcal{K}|}-1}^{n-k-1}) - 1, \\
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|}} - \text{wt}(\boldsymbol{s}_{n-k-j+m_{|\mathcal{K}|}}^{n-k-1}) - 1, \\
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|-1}} - \text{wt}(\boldsymbol{s}_{n-k-j+m_{|\mathcal{K}|}+1}^{n-k-1}) - 1, \\
\qquad\qquad \vdots \\
a + s_{m_1} + s_{m_2} - \text{wt}(\boldsymbol{s}_{n-k-j+m_2-1}^{n-k-1}) - 1, \\
a + s_{m_1} + s_{m_2} - \text{wt}(\boldsymbol{s}_{n-k-j+m_2}^{n-k-1}) - 1, \\
a + s_{m_1} - \text{wt}(\boldsymbol{s}_{n-k-j+m_2+1}^{n-k-1}) - 1, \\
\qquad\qquad \vdots \\
a + s_{m_1} - \text{wt}(\boldsymbol{s}_{n-k-j+m_1-1}^{n-k-1}) - 1, \\
a + s_{m_1} - \text{wt}(\boldsymbol{s}_{n-k-j+m_1}^{n-k-1}) - 1, \\
a - \text{wt}(\boldsymbol{s}_{n-k-j+m_1+1}^{n-k-1}) - 1, \\
\qquad\qquad \vdots \\
a - s_{n-k-1} - 1, \\
a - 1
\end{array}\right\}.$$

Similarly, $C^{(1)}(\boldsymbol{v})$ may be restated as

$$\left\{\begin{array}{l}
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|}} - \text{wt}(\boldsymbol{v}_{n-k-j+1}^{n-k-1}), \\
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|}} - \text{wt}(\boldsymbol{v}_{n-k-j+2}^{n-k-1}), \\
\qquad\qquad \vdots \\
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|}} - \text{wt}(\boldsymbol{v}_{n-k-j+m_{|\mathcal{K}|}-1}^{n-k-1}), \\
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|}} - \text{wt}(\boldsymbol{v}_{n-k-j+m_{|\mathcal{K}|}}^{n-k-1}), \\
a + s_{m_1} + \ldots + s_{m_{|\mathcal{K}|-1}} - \text{wt}(\boldsymbol{v}_{n-k-j+m_{|\mathcal{K}|}+1}^{n-k-1}), \\
\qquad\qquad \vdots \\
a + s_{m_1} + s_{m_2} - \text{wt}(\boldsymbol{v}_{n-k-j+m_2-1}^{n-k-1}), \\
a + s_{m_1} + s_{m_2} - \text{wt}(\boldsymbol{v}_{n-k-j+m_2}^{n-k-1}), \\
a + s_{m_1} - \text{wt}(\boldsymbol{v}_{n-k-j+m_2+1}^{n-k-1}), \\
\qquad\qquad \vdots \\
a + s_{m_1} - \text{wt}(\boldsymbol{v}_{n-k-j+m_1-1}^{n-k-1}), \\
a + s_{m_1} - \text{wt}(\boldsymbol{v}_{n-k-j+m_1}^{n-k-1}), \\
a - \text{wt}(\boldsymbol{v}_{n-k-j+m_1+1}^{n-k-1}), \\
\qquad\qquad \vdots \\
a - v_{n-k-1}, \\
a
\end{array}\right\}. \tag{25}$$

We wish to match the composition $\gamma_j(\boldsymbol{v})$ with weight a to any element in $C^{(1)}(\boldsymbol{s})$. To this end, we set $\gamma_{m_1}(\boldsymbol{s}) = a + s_{m_1} - \text{wt}(\boldsymbol{s}_{n-k-j+m_1}^{n-k-1}) - 1 = a$, while ensuring that for any $p \in \{m_1+1, \ldots, j\}$, we have $\gamma_p(\boldsymbol{s}) \leq a - 1$. Again, since for any $p \in [j-1]$ we have $|\gamma_{p+1}(\boldsymbol{s}) - \gamma_p(\boldsymbol{s})| \leq 1$, it must hold that $\gamma_{m_1+1}(\boldsymbol{s}) = a - 1$. These impositions essentially translate to

$$\begin{aligned} \boldsymbol{s}_{n-k-j+m_1}^{n-k-1} &= \boldsymbol{0}, \\ s_{m_1} &= 1, \\ \gamma_p(\boldsymbol{s}) &= a - 1 \quad \forall\, p \in \{m_1+1, \ldots, j\}. \end{aligned} \tag{26}$$

Under these conditions, $C^{(1)}(\boldsymbol{s}) = C^{(1)}(\boldsymbol{v})$ reduces to

$$\left\{\begin{array}{l}
a + \displaystyle\sum_{i=2}^{|\mathcal{K}|} s_{m_i} - \text{wt}(\boldsymbol{s}_{n-k-j+1}^{n-k-j+m_1-1}), \\
a + \displaystyle\sum_{i=2}^{|\mathcal{K}|} s_{m_i} - \text{wt}(\boldsymbol{s}_{n-k-j+2}^{n-k-j+m_1-1}), \\
\qquad\qquad \vdots \\
a + s_{m_2} - \text{wt}(\boldsymbol{s}_{n-k-j+m_2-1}^{n-k-j+m_1-1}), \\
a + s_{m_2} - \text{wt}(\boldsymbol{s}_{n-k-j+m_2}^{n-k-j+m_1-1}), \\
a - \text{wt}(\boldsymbol{s}_{n-k-j+m_2+1}^{n-k-j+m_1-1}), \\
\qquad\qquad \vdots \\
a - s_{n-k-j+m_1-1}, \\
a, \\
a - 1, \\
\qquad\qquad \vdots \\
a - 1, \\
a - 1
\end{array}\right\}$$

$$
= \left\{\begin{array}{l}
a+1+\sum_{i=2}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{n-k-j+1}^{n-k-1}),\\
a+1+\sum_{i=2}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{n-k-j+2}^{n-k-1}),\\
\vdots\\
a+1+s_{m_2} - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_2-1}^{n-k-1}),\\
a+1+s_{m_2} - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_2}^{n-k-1}),\\
a+1 - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_2+1}^{n-k-1}),\\
\vdots\\
a+1 - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_1-1}^{n-k-1}),\\
a+1 - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_1}^{n-k-1}),\\
a - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_1+1}^{n-k-1}),\\
\vdots\\
a - v_{n-k-1},\\
a
\end{array}\right\}. \tag{27}
$$

Since $\gamma_p(\boldsymbol{s}) = a-1$ for all $m_1+1 \le p \le j$, we need $j - m_1$ entries in $C^{(1)}(\boldsymbol{v})$ to be equal to $a-1$. We wish to achieve this by setting the elements $\gamma_{j-1}(\boldsymbol{v}), \gamma_{j-2}(\boldsymbol{v}), \ldots, \gamma_{m_1}(\boldsymbol{v})$ to $a-1$. We prefer to pick these entries due to their adjacency to $\gamma_j(\boldsymbol{v})$, as this implies fewer constraints.[5] Now in order to set $\gamma_{j-1}(\boldsymbol{v}) = \gamma_{j-2}(\boldsymbol{v}) = \ldots = \gamma_{m_1}(\boldsymbol{v}) = a-1$, we require

$$
\begin{aligned}
v_{n-k-1} &= 1,\\
v_{n-k-j+m_1+1}^{n-k-2} &= \boldsymbol{0},\\
v_{n-k-j+m_1} &= 1.
\end{aligned} \tag{28}
$$

Consequently, (27) becomes

$$
C^{(1)}(\boldsymbol{s}) = \left\{\begin{array}{l}
a+\sum_{i=2}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{n-k-j+1}^{n-k-j+m_1-1}),\\
a+\sum_{i=2}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{n-k-j+2}^{n-k-j+m_1-1}),\\
\vdots\\
a+s_{m_2} - \mathrm{wt}(\boldsymbol{s}_{n-k-j+m_2-1}^{n-k-j+m_1-1}),\\
a+s_{m_2} - \mathrm{wt}(\boldsymbol{s}_{n-k-j+m_2}^{n-k-j+m_1-1}),\\
a - \mathrm{wt}(\boldsymbol{s}_{n-k-j+m_2+1}^{n-k-j+m_1-1}),\\
\vdots\\
a - s_{n-k-j+m_1-1},\\
a,\\
a-1,\\
\vdots\\
a-1,\\
a-1
\end{array}\right\}
$$

$$
= \left\{\begin{array}{l}
a-1+\sum_{i=2}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{n-k-j+1}^{n-k-j+m_1-1}),\\
a-1+\sum_{i=2}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{n-k-j+2}^{n-k-j+m_1-1}),\\
\vdots\\
a-1+s_{m_2} - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_2-1}^{n-k-j+m_1-1}),\\
a-1+s_{m_2} - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_2}^{n-k-j+m_1-1}),\\
a-1 - \mathrm{wt}(\boldsymbol{v}_{n-k-j+m_2+1}^{n-k-j+m_1-1}),\\
\vdots\\
a-1 - v_{n-k-j+m_1-1},\\
a-1,\\
a-1,\\
\vdots\\
a-1,\\
a
\end{array}\right\} = C^{(1)}(\boldsymbol{v}).
$$

Now observe that, if we additionally set

$$
\begin{aligned}
(s_{n-k-j+m_1-1}, v_{n-k-j+m_1-1}) &= (1,0),\\
\boldsymbol{s}_{n-k-j+1}^{n-k-j+m_1-2} &= \boldsymbol{v}_{n-k-j+1}^{n-k-j+m_1-2}.
\end{aligned} \tag{29}
$$

then $C^{(1)}(\boldsymbol{s}) = C^{(1)}(\boldsymbol{v})$ definitely holds. Now to see how the conditions imposed on $\boldsymbol{s}$ impact the satisfaction of $C^{(2)}(\boldsymbol{s}) = C^{(2)}(\boldsymbol{v})$, we note that on account of $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$, the following relations are automatically implied by (26), (28) and (29).

$$
\begin{aligned}
(s_{k+2}, v_{k+2}) &= (1,0)\\
\boldsymbol{s}_{k+3}^{k+j-m_1} &= \boldsymbol{v}_{k+3}^{k+j-m_1},\\
(s_{k+j-m_1+1}, v_{k+j-m_1+1}) &= (1,0),\\
(s_{k+j-m_1+2}, v_{k+j-m_1+2}) &= (0,1),\\
\boldsymbol{s}_{k+j-m_1+3}^{k+j} &= \boldsymbol{v}_{k+j-m_1+3}^{k+j}.
\end{aligned} \tag{30}
$$

Now as done previously for $C^{(1)}(\boldsymbol{s})$ and $C^{(1)}(\boldsymbol{v})$, we first restate $C^{(2)}(\boldsymbol{s})$ and $C^{(2)}(\boldsymbol{v})$ to deduce the conditions that lead to $C^{(2)}(\boldsymbol{s}) = C^{(2)}(\boldsymbol{v})$. To this end, we first express $C^{(2)}(\boldsymbol{s})$ as

$$
\left\{\begin{array}{l}
b+j-|\mathcal{K}|+\sum_{i=1}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j}),\\
b+j-|\mathcal{K}|-1+\sum_{i=1}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-1}),\\
\vdots\\
b+j-m_{|\mathcal{K}|}-|\mathcal{K}|+2+\sum_{i=1}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_{|\mathcal{K}|}+2}),\\
b+j-m_{|\mathcal{K}|}-|\mathcal{K}|+1+\sum_{i=1}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_{|\mathcal{K}|}+1}),\\
b+j-m_{|\mathcal{K}|}-|\mathcal{K}|+1+\sum_{i=1}^{|\mathcal{K}|-1} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_{|\mathcal{K}|}}),\\
\vdots\\
b+j-m_2+\sum_{i=1}^{2} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_2+2}),\\
b+j-m_2-1+\sum_{i=1}^{2} s_{m_i} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_2+1}),\\
b+j-m_2-1+s_{m_1} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_2}),\\
\vdots\\
b+j-m_1+1+s_{m_1} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_1+2}),\\
b+j-m_1+s_{m_1} - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_1+1}),\\
b+j-m_1 - \mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_1}),\\
\vdots\\
b+2-s_{k+2},\\
b+1
\end{array}\right\}.
$$

[5] Any alternative assignment leads to the same conclusion.

Next, we restate $C^{(2)}(\boldsymbol{v})$ as

$$\begin{Bmatrix} b+j-|\mathcal{K}|-1+\sum_{i=1}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j}), \\ b+j-|\mathcal{K}|-2+\sum_{i=1}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-1}), \\ \vdots \\ b+j-m_{|\mathcal{K}|}-|\mathcal{K}|+1+\sum_{i=1}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_{|\mathcal{K}|}+2}), \\ b+j-m_{|\mathcal{K}|}-|\mathcal{K}|+\sum_{i=1}^{|\mathcal{K}|} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_{|\mathcal{K}|}+1}), \\ b+j-m_{|\mathcal{K}|}-|\mathcal{K}|+\sum_{i=1}^{|\mathcal{K}|-1} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_{|\mathcal{K}|}}), \\ \vdots \\ b+j-m_2-1+\sum_{i=1}^{2} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_2+2}), \\ b+j-m_2-2+\sum_{i=1}^{2} s_{m_i} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_2+1}), \\ b+j-m_2-2+s_{m_1} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_2}), \\ \vdots \\ b+j-m_1+s_{m_1} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_1+2}), \\ b+j-m_1-1+s_{m_1} - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_1+1}), \\ b+j-m_1-1 - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_1}), \\ \vdots \\ b+1-v_{k+2}, \\ b \end{Bmatrix}.$$

Note that (30) further suggests that $\mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_1+2}) - \mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_1+2}) = 1$. Combining this with $s_{m_1} = 1$ and $\boldsymbol{s}_{k+j-m_1+3}^{k+j} = \boldsymbol{v}_{k+j-m_1+3}^{k+j}$, we are able to simplify $C^{(2)}(\boldsymbol{s}) = C^{(2)}(\boldsymbol{v})$, using the preceding expressions, to

$$\begin{Bmatrix} b+j-m_1+1-\mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_1+1}), \\ b+j-m_1-\mathrm{wt}(\boldsymbol{s}_{k+2}^{k+j-m_1}), \\ \vdots \\ b+2-s_{k+2}, \\ b+1 \end{Bmatrix} = \begin{Bmatrix} b+j-m_1-\mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_1+1}), \\ b+j-m_1-1-\mathrm{wt}(\boldsymbol{v}_{k+2}^{k+j-m_1}), \\ \vdots \\ b+1-v_{k+2}, \\ b \end{Bmatrix}.$$

Now observe that the remaining element b in $C^{(2)}(\boldsymbol{v})$ cannot be matched to any of the remaining elements in $C^{(2)}(\boldsymbol{s})$. Thus, once again $C_{n-k-j}(\boldsymbol{s}) = C_{n-k-j}(\boldsymbol{v})$ does not hold, since we have $C^{(2)}(\boldsymbol{s}) \neq C^{(2)}(\boldsymbol{v})$.

□

Remark 2: Observe that whenever we have two codestrings $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_{DA}^{(t)}(n)$ such that $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$, $\boldsymbol{s}_1^{t+1} = \boldsymbol{0}$,$\boldsymbol{s}_{n-t}^n = \boldsymbol{1}$, $(\boldsymbol{s}_1^k, \boldsymbol{s}_{n-k+1}^n) = (\boldsymbol{v}_1^k, \boldsymbol{v}_{n-k+1}^n)$ and $s_{k+1} \neq v_{k+1}$ for any $t+2 \leq k \leq \lfloor \frac{n}{2} \rfloor$, we deduce that for all $1 \leq j \leq t+1$, $C_{n-k-j}(\boldsymbol{s}) = C_{n-k-j}(\boldsymbol{v})$ never holds, since $\boldsymbol{s}_1^j = \boldsymbol{v}_1^j = \boldsymbol{0}$, in turn implying that $C^{(1)}(\boldsymbol{s}) = C^{(1)}(\boldsymbol{v})$ is always untrue.[6] Since $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_R^{(t+1)}(n)$ [11], we conclude that $\mathcal{S}_R^{(t+1)}(n)$ is also a t-asymmetric multiset deletion code.

The preceding lemma now helps us establish that the code $\mathcal{S}_{DA}^{(t)}(n)$ is robust to the deletion of any t asymmetric multisets.

Theorem 2: $\mathcal{S}_{DA}^{(t)}(n)$ is a t-asymmetric multiset deletion code.

Proof: In the following, we wish to establish that for any $\boldsymbol{s} \in \mathcal{S}_{DA}^{(t)}(n)$, if one is given $C'(\boldsymbol{s}) = \bigcup_{i \in [n] \setminus \{i_1, \ldots i_t\}} C_i(\boldsymbol{s})$, i.e., the multisets $C_{i_1}(\boldsymbol{s}), \ldots, C_{i_t}(\boldsymbol{s})$ are deleted, such that no two of the deleted multisets are mutually symmetric, then $\boldsymbol{s}$ can be uniquely recovered. As pointed out previously in Remark 1, $\boldsymbol{\sigma}_s$ can be uniquely recovered from the corrupted composition multiset.

Case 1: The deleted multisets are consecutive.

This case is directly implied by Lemma 4.

Case 2: Not all of the deleted multisets are consecutive. Since the reconstruction algorithm functions in an outside-in manner, the missing multiset encountered first, corresponds to that of highest substring length. In the following analysis, we assume that $i_t > i_{t-1} > \ldots > i_1$.

If $i_t = n$, we can directly infer $C_n(\boldsymbol{s})$ from the cumulative weight of $C_1(\boldsymbol{s})$. Alternatively when $i_t < n$ and additionally $i_t, \ldots, i_{t-j+1}$ are consecutive, we have $(\boldsymbol{s}_1^{n-i_t-1}, \boldsymbol{s}_{i_t+2}^n) = (\boldsymbol{v}_1^{n-i_t-1}, \boldsymbol{v}_{i_t+2}^n)$ since for $j \in \{n, n-1, \ldots, i_t+1\}$, it holds that $C_j(\boldsymbol{s}) = C_j(\boldsymbol{v})$. Now, an incorrect assignment of the bit pair (s_{n-i_t}, s_{i_t+1}) will certainly cause an incompatibility with the multiset $C_{i_{t-j+1}-1}(\boldsymbol{s}) = C_{i_t-j}(\boldsymbol{s})$, as Lemma 4 suggests. Thus, the backtracking algorithm can detect the mistake and accurately reconstruct the string up to $(\boldsymbol{s}_1^{n-i_t+j}, \boldsymbol{s}_{i_t-j+1}^n)$. Absence of the other missing multisets $C_{i_{t-j}}, \ldots, C_{i_1}$ can be dealt with similarly, by successively applying the aforementioned argument.

□

We also bound the number of redundant bits required by $\mathcal{S}_{DA}^{(t)}(n)$ as follows.

Lemma 5: The code $\mathcal{S}_{DA}^{(t)}(n)$ requires at most $\frac{1}{2}\log(n-2t-2) + 2t + 7$ bits of redundancy.

Proof: We refer to (9) and additionally recount from [11] that $\frac{1}{2}\binom{2h}{h}$ indicates the number of all strings of length $2h$ wherein every prefix of which contains strictly more 0s than 1s. For odd lengths $2h+1$, this term serves as a lower bound. Similarly, to count all strings $\boldsymbol{s} \in \{0,1\}^p$ wherein each prefix (of length exceeding t) contains at least t more 0s than 1s, we simply note that such strings satisfy $\boldsymbol{s}_1^{t-1} = \boldsymbol{0}$ and

[6] As $w_j(\boldsymbol{v})$ cannot be matched to any element in $C^{(1)}(\boldsymbol{s})$.

$\boldsymbol{s}_t^p$ should be a standard Catalan-Bertrand string. By virtue of this, we derive a lower bound on dimension of the codebook for even values of n.

$$|\mathcal{S}_{DA}^{(t)}(n)| \geq 3 \sum_{i=t}^{\frac{n-2}{2}-1} 2^{\frac{n-2}{2}-2-i} \binom{\frac{n-2}{2}-1}{i} \binom{i-t+1}{\lfloor (i-t+1)/2 \rfloor}.$$

After some algebraic manipulation of this expression, we conclude that the maximum number of redundant bits necessary is $\frac{1}{2}\log(n-2t-2)+2t+7$. It is easy to see that this bound on the number of redundant bits also holds for odd values of n. □

VI. Symmetric Multiset Deletion Codes

As mentioned in Section III, errors under this category occur in such a way that the affected multisets occur in pairs. We begin directly with the case when two symmetric multisets are inaccessible.

Theorem 3: The code $\mathcal{S}_R(n)$ is a single symmetric multiset deletion code.

Proof: In the following, we wish to prove that for any $\boldsymbol{s} \in \mathcal{S}_R(n)$, if we are given $C'(\boldsymbol{s}) = \bigcup_{i \in [n] \setminus \{k, n-k+1\}} C_i(\boldsymbol{s})$ for some $1 \leq k \leq \lceil \frac{n-1}{2} \rceil$, then $\boldsymbol{s}$ can be fully recovered.

Case 1: n is odd.

Since the deleted multisets $C_k(\boldsymbol{s})$ and $C_{n-k+1}(\boldsymbol{s})$ can never be consecutive when n is odd, we can infer from [11, Lemma 4] that any attempt to substitute $C_{n-k+1}(\boldsymbol{s})$ with another multiset, say C'_{n-k+1}, that may or may not preserve the value of $\sigma_{k-1}(\boldsymbol{s})$, will surely cause a disagreement with $C_{n-k}(\boldsymbol{s})$. Hence, there exists no valid alternative choices for the multiset pair $\{C_k(\boldsymbol{s}), C_{n-k+1}(\boldsymbol{s})\}$, thus implying that $\boldsymbol{s}$ is uniquely reconstructable.

Case 2: n is even.

As in the previous case, we can argue that for any $k \neq \{\frac{n}{2}, \frac{n}{2}+1\}$, i.e., when the missing multisets are non-consecutive, $\boldsymbol{s}$ remains unique reconstructable by virtue of [11, Lemma 4]. The only case left to be analyzed is when the deleted multisets are adjacent, i.e., $C_{\frac{n}{2}}(\boldsymbol{s})$ and $C_{\frac{n}{2}+1}(\boldsymbol{s})$. More specifically, we examine the existence of any $\boldsymbol{v} \in \mathcal{S}_R(n)$, such that

$$\bigcup_{i \in [n] \setminus \{\frac{n}{2}, \frac{n}{2}+1\}} C_i(\boldsymbol{v}) = \bigcup_{i \in [n] \setminus \{\frac{n}{2}, \frac{n}{2}+1\}} C_i(\boldsymbol{s}).$$

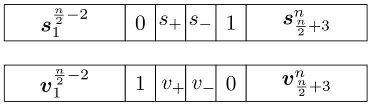
This directly leads to the following relations:

$$\begin{aligned} (\boldsymbol{s}_1^{n/2-2}, \boldsymbol{s}_{n/2+3}^n) &= (\boldsymbol{v}_1^{n/2-2}, \boldsymbol{v}_{n/2+3}^n), \\ \sigma_i = \sigma'_i, &\quad \forall\ 1 \leq i \leq \frac{n}{2}-2 \\ \sigma_{\frac{n}{2}-1} + \sigma_{\frac{n}{2}} &= \sigma'_{\frac{n}{2}-1} + \sigma'_{\frac{n}{2}}. \end{aligned}$$

where the sequence $\boldsymbol{\sigma_v} = (\sigma'_1, \ldots, \sigma'_{n/2})$ describes $\boldsymbol{v}$.

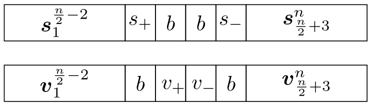
Subcase (i): $\boldsymbol{\sigma}_s = \boldsymbol{\sigma_v}$

We only study this subcase for when $\sigma_{\frac{n}{2}-1} = \sigma'_{\frac{n}{2}-1} = 1$ and $(s_{\frac{n}{2}-1}, s_{\frac{n}{2}+2}) \neq (v_{\frac{n}{2}-1}, v_{\frac{n}{2}+2})$, since the alternative involves $C_{n/2+1}(\boldsymbol{s}) = C_{n/2+1}(\boldsymbol{v})$ and as a result of this, Lemma 3 precludes the existence of $\boldsymbol{v}$, since $C(\boldsymbol{s})$ and $C(\boldsymbol{v})$ cannot differ by a single multiset alone. This situation is illustrated in Fig. 3.



$\boldsymbol{s}_1^{\frac{n}{2}-2}$
0
s_+
s_-
1
$\boldsymbol{s}_{\frac{n}{2}+3}^n$
$\boldsymbol{v}_1^{\frac{n}{2}-2}$
1
v_+
v_-
0
$\boldsymbol{v}_{\frac{n}{2}+3}^n$

Fig. 3. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are such that $(\boldsymbol{s}_1^{\frac{n}{2}-2}, \boldsymbol{s}_{\frac{n}{2}+3}^n) = (\boldsymbol{v}_1^{\frac{n}{2}-2}, \boldsymbol{v}_{\frac{n}{2}+3}^n)$, where $v_+ + v_- = s_+ + s_-$.



$\boldsymbol{s}_1^{\frac{n}{2}-2}$
s_+
b
b
s_-
$\boldsymbol{s}_{\frac{n}{2}+3}^n$
$\boldsymbol{v}_1^{\frac{n}{2}-2}$
b
v_+
v_-
b
$\boldsymbol{v}_{\frac{n}{2}+3}^n$

Fig. 4. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are such that $(\boldsymbol{s}_1^{\frac{n}{2}-2}, \boldsymbol{s}_{\frac{n}{2}+3}^n) = (\boldsymbol{v}_1^{\frac{n}{2}-2}, \boldsymbol{v}_{\frac{n}{2}+3}^n)$, where $s_+ + s_- = v_+ + v_- = 1$.

We now proceed to ascertain if there exists some $\boldsymbol{v}$ for which $C_{n/2-1}(\boldsymbol{s}) = C_{n/2-1}(\boldsymbol{v})$ holds. Alternatively, we need the following set equality relation to hold:

$$\left\{ \begin{array}{l} \{c(\boldsymbol{s}_1^{\frac{n}{2}-2}), 0\}, \\ \{c(\boldsymbol{s}_2^{\frac{n}{2}-2}), 0, s_+\}, \\ \{c(\boldsymbol{s}_3^{\frac{n}{2}-2}), 0, s_+, s_-\}, \\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^n), 1\}, \\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-1}), 1, s_-\}, \\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-2}), 1, s_+, s_-\} \end{array} \right\} = \left\{ \begin{array}{l} \{c(\boldsymbol{v}_1^{\frac{n}{2}-2}), 1\}, \\ \{c(\boldsymbol{v}_2^{\frac{n}{2}-2}), 1, v_+\}, \\ \{c(\boldsymbol{v}_3^{\frac{n}{2}-2}), 1, v_+, v_-\}, \\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^n), 0\}, \\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-1}), 0, v_-\}, \\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-2}), 0, v_+, v_-\} \end{array} \right\}. \tag{31}$$

Due to the weight mismatch property between prefix and suffix of equal lengths, we note from Fig. 3 that $\boldsymbol{v}$ must uphold:

$$\begin{aligned} \mathrm{wt}(\boldsymbol{s}_2^{n/2-2}) + 1 &< \mathrm{wt}(\boldsymbol{s}_{n/2+3}^{n-1}) \\ \implies \mathrm{wt}(\boldsymbol{s}_1^{n/2-2}) + 3 &\leq \mathrm{wt}(\boldsymbol{s}_{n/2+3}^n). \end{aligned} \tag{32}$$

Now to prove that (31) never holds, it suffices to show that the composition $\{c(\boldsymbol{s}_{\frac{n}{2}+3}^n), 1\}$ can never be matched to any two elements on the RHS in (31), even when (32) holds with equality. It is easy to see this when $v_+ + v_- < 2$. On the contrary, when $v_+ + v_- = 2$, the compositions $\{c(\boldsymbol{v}_1^{\frac{n}{2}-2}), 1\}$ and $\{c(\boldsymbol{v}_2^{\frac{n}{2}-2}), 1, v_+\}$ become identical, and cannot be matched simultaneously to the components of RHS in (31). Therefore, $\boldsymbol{v}$ does not exist.

Subcase (ii): $\boldsymbol{\sigma}_s \neq \boldsymbol{\sigma_v}$

All of the possible combinations of $(\sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}})$ and $(\sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}})$ that comprehensively cover this subcase are:

- $(\sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) = (1, 2b)$ and $(\sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) = (2b, 1)$.
- $(\sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) = (2, 0)$ and $(\sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) = (1, 1)$.
- $(\sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) = (0, 2)$ and $(\sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) = (1, 1)$.

where $b \in \mathbb{F}_2$. For the sake of brevity, we only prove the first instance. The remaining proofs run in a similar fashion.

To reiterate our objective, we check for the existence of a string $\boldsymbol{v} \in \mathcal{S}_R(n)$, for a given $\boldsymbol{s} \in \mathcal{S}_R(n)$, which are characterized as per the depiction in Fig. 4. Since $\boldsymbol{s}$ and $\boldsymbol{v}$ may only differ in their respective composition multisets of substring lengths $\frac{n}{2}$ and $\frac{n}{2}+1$ alone, we endeavor to find the conditions that allow for the set equality of $C_{\frac{n}{2}-1}(\boldsymbol{s})$ and $C_{\frac{n}{2}-1}(\boldsymbol{v})$. More explicitly, we require:

$$\left\{\begin{matrix}\{c(\boldsymbol{s}_1^{\frac{n}{2}-2}), s_+\}, \\ \{c(\boldsymbol{s}_2^{\frac{n}{2}-2}), s_+, b\}, \\ \{c(\boldsymbol{s}_3^{\frac{n}{2}-2}), s_+, b^2\}, \\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n}), 1-s_+\}, \\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-1}), 1-s_+, b\}, \\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-2}), 1-s_+, b^2\}\end{matrix}\right\} = \left\{\begin{matrix}\{c(\boldsymbol{v}_1^{\frac{n}{2}-2}), b\}, \\ \{c(\boldsymbol{v}_2^{\frac{n}{2}-2}), b, v_+\}, \\ \{c(\boldsymbol{v}_3^{\frac{n}{2}-2}), b, 01\}, \\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n}), b\}, \\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-1}), b, 1-v_+\}, \\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-2}), b, 01\}\end{matrix}\right\}.$$

When $s_+ = v_+ = 0$, we may proceed under the assumption that $\mathrm{wt}(\boldsymbol{s}_2^{n/2-2}) = \mathrm{wt}(\boldsymbol{s}_{n/2+3}^{n-1})$ to account for the worst case. In this situation, either $\{c(\boldsymbol{s}_1^{\frac{n}{2}-2}), s_+\}$ or $\{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n}), 1-s_+\}$ fails to be matched, depending on the chosen value of b. Else, when either s_+ or v_+ equals 1, we infer that (32) holds true. Again, we choose to proceed with the worst case, i.e., $\mathrm{wt}(\boldsymbol{s}_2^{n/2-2}) + 3 = \mathrm{wt}(\boldsymbol{s}_{n/2+3}^{n-1})$, and an exhaustive examination of each possibility reveals that the previous set equality cannot be satisfied. Thus, we conclude that $\boldsymbol{v}$ does not exist.

□

Consequently, if a single composition is substituted in $C(\boldsymbol{s})$ where $\boldsymbol{s} \in \mathcal{S}_R(n)$, then there occurs a mismatch between the cumulative weights of the specific multiset affected, say $C_i(\boldsymbol{s})$, and its symmetric counterpart $C_{n-i+1}(\boldsymbol{s})$. Now if both $C_i(\boldsymbol{s})$ and $C_{n-i+1}(\boldsymbol{s})$ are deleted, Lemma 3 tells us that $\boldsymbol{s}$ is still uniquely recoverable. Thus, we conclude that $\mathcal{S}_R(n)$ is capable of correcting a single composition error just like $S_C^{(1)}(n)$ [11], [12].

We now investigate further along this direction and seek to determine if the absence of multiple pairs of such multisets impacts reconstructability. The deletion of two or more pairs of symmetric multisets, as shown in Lemma 9 (Appendix), no longer guarantees unique reconstruction of codewords drawn from $\mathcal{S}_R(n)$. To remedy this, we propose the code $\mathcal{S}_{DS}^{(2)}(n)$ [see Construction 3], capable of correcting deletions of two pairs of symmetric sets.

Theorem 4: The code $\mathcal{S}_{DS}^{(2)}(n)$ is a 2-symmetric multiset deletion code.

Proof: To prove the statement of this theorem, we intend to show that for any $\boldsymbol{s} \in \mathcal{S}_{DS}^{(2)}(n)$, if we are only given the composition multisets $\bigcup_{i\in[n]\setminus\{k_1,k_2,n-k_1+1,n-k_2+1\}} C_i(\boldsymbol{s})$, where $1 \le k_1, k_2 \le \lfloor \frac{n}{2} \rfloor$, then $\boldsymbol{s}$ can be uniquely recovered. We prove this by contradiction, i.e., we attempt to find some $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_{DS}^{(2)}(n)$, such that

$$\bigcup_{i\in[n]\setminus\substack{\{k_1,k_2,n-k_1+1,\\ n-k_2+1\}}} C_i(\boldsymbol{s}) = \bigcup_{i\in[n]\setminus\substack{\{k_1,k_2,n-k_1+1,\\ n-k_2+1\}}} C_i(\boldsymbol{v}).$$

Additionally, we let $\boldsymbol{\sigma}_s = (\sigma_1, \ldots, \sigma_{n/2})$ and $\boldsymbol{\sigma}_v = (\sigma_1', \ldots, \sigma_{n/2}')$.

Case 1: n is even and the deleted multisets are neighboring, i.e., $\{C_{n/2-1}(\boldsymbol{s}), \ldots, C_{n/2+2}(\boldsymbol{s})\}$.

Since the cumulative weights $w_1(\boldsymbol{s}), \ldots, w_{\frac{n}{2}-2}(\boldsymbol{s})$ are still accessible, one can unambiguously deduce $(\sigma_1, \ldots, \sigma_{\frac{n}{2}-3})$ by applying (6). We thus have

$$\begin{aligned} \sigma_i &= \sigma_i', \qquad \forall \ \ 1 \le i \le \frac{n}{2} - 3 \\ \sigma_{\frac{n}{2}-2} + \sigma_{\frac{n}{2}-1} + \sigma_{\frac{n}{2}} &= \sigma_{\frac{n}{2}-2}' + \sigma_{\frac{n}{2}-1}' + \sigma_{\frac{n}{2}}', \end{aligned} \tag{33}$$

where the second equation follows from a combining its predecessor, $w_1(\boldsymbol{s}) = w_1(\boldsymbol{v})$ and (4).

The difference of the sum of their respective cumulative weights for composition multisets containing substrings of lengths from 1 to $\frac{n}{2}$, can be simplified to the following, by means of (5) and (33).

$$\begin{aligned} &\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) - \sum_{i=1}^{n/2} w_i(\boldsymbol{v}) \\ &= \sum_{i=n/2-1}^{n/2} w_i(\boldsymbol{s}) - \sum_{i=n/2-1}^{n/2} w_i(\boldsymbol{v}) \\ &= 3(\sigma_{n/2-2}' - \sigma_{n/2-2}) + (\sigma_{n/2-1}' - \sigma_{n/2-1}). \end{aligned} \tag{34}$$

The above difference is maximized when either:

$$\begin{aligned} (\sigma_{\frac{n}{2}-2}, \sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) &= (0,1,2), \\ (\sigma_{\frac{n}{2}-2}', \sigma_{\frac{n}{2}-1}', \sigma_{\frac{n}{2}}') &= (2,1,0). \end{aligned}$$

or:

$$\begin{aligned} (\sigma_{\frac{n}{2}-2}, \sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) &= (0,2,2), \\ (\sigma_{\frac{n}{2}-2}', \sigma_{\frac{n}{2}-1}', \sigma_{\frac{n}{2}}') &= (2,2,0). \end{aligned}$$

In either case, (33) is upheld. Hence, we can write that:

$$\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) - \sum_{i=1}^{n/2} w_i(\boldsymbol{v}) \le 6.$$

Thus, an additional constraint of $\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) \bmod 7$ will be sufficient to fully recover $\boldsymbol{\sigma}_s$. This implies that $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$.

Now since $C_i(\boldsymbol{s}) = C_i((\boldsymbol{v})$ for $n/2 + 3 \le i \le n$, it immediately follows that

$$(\boldsymbol{s}_1^{n/2-3}, \boldsymbol{s}_{n/2+4}^{n}) = (\boldsymbol{v}_1^{n/2-3}, \boldsymbol{v}_{n/2+4}^{n}), \tag{35}$$

by virtue of how the reconstruction algorithm works and the fact that $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_{DS}^{(2)}(n) \subset \mathcal{S}_R(n)$. Following this, we seek to determine whether $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ can hold.

Subcase (i): $s_{n/2-2} \neq v_{n/2-2}$

Without loss of generality, we assume $(s_{n/2-2}, v_{n/2-2}) = (0,1)$. Since $\sigma_{n/2-2} = \sigma_{n/2-2}'$, it must hold that $(s_{n/2+3}, v_{n/2+3}) = (0,1)$. This has been illustrated in Fig. 5.

Moreover, on account of the Catalan-Bertrand constraint imposed by $\mathcal{S}_{DS}^{(2)}(n)$, we note that for some $\mathcal{I} \subseteq \{2, \ldots, \frac{n}{2} - 2\}$, such that for any $i \in \mathcal{I}$, $v_i \neq v_{n-i+1}$, we can write

$$\begin{aligned} |\mathcal{I}| - \mathrm{wt}(\boldsymbol{v}_{\mathcal{I}}) &\ge \mathrm{wt}(\boldsymbol{v}_{\mathcal{I}}) + 1, \\ \implies \mathrm{wt}(\boldsymbol{v}_{\widetilde{\mathcal{I}}}) &\ge \mathrm{wt}(\boldsymbol{v}_{\mathcal{I}}) + 1, \end{aligned}$$

where $\widetilde{\mathcal{I}} = \{n - i + 1 : i \in \mathcal{I}\}$. Since for any $i \in [\frac{n}{2} - 2]\backslash\mathcal{I}$, $v_i = v_{n-i+1}$, the preceding equation can be restated to

$$\begin{aligned} \mathrm{wt}(\boldsymbol{v}_{n/2+3}^{n-1}) &\ge \mathrm{wt}(\boldsymbol{v}_2^{n/2-2}) + 1, \\ \implies \mathrm{wt}(\boldsymbol{v}_{n/2+4}^{n}) &\ge \mathrm{wt}(\boldsymbol{v}_1^{n/2-3}) + 3, \\ \implies \mathrm{wt}(\boldsymbol{s}_{n/2+4}^{n}) &\ge \mathrm{wt}(\boldsymbol{s}_1^{n/2-3}) + 3, \end{aligned} \tag{36}$$

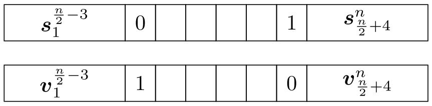

$\boldsymbol{s}_1^{\frac{n}{2}-3}$
0
1
$\boldsymbol{s}_{\frac{n}{2}+4}^{n}$
$\boldsymbol{v}_1^{\frac{n}{2}-3}$
1
0
$\boldsymbol{v}_{\frac{n}{2}+4}^{n}$

Fig. 5. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are such that $(\boldsymbol{s}_1^{\frac{n}{2}-3},\boldsymbol{s}_{\frac{n}{2}+4}^{n}) = (\boldsymbol{v}_1^{\frac{n}{2}-3},\boldsymbol{v}_{\frac{n}{2}+4}^{n})$, and $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$.

where the second inequality follows from $(v_1, v_n) = (v_{n/2+3}, v_{n/2-2}) = (0,1)$, while the final expression is implied by (35).

Next, we observe that for any $6 \leq i \leq n/2-2$, $c(\boldsymbol{s}_i^{i+n/2-3}) = c(\boldsymbol{v}_i^{i+n/2-3})$. Hence, for $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ to be upheld, we simply require that

$$\left\{\begin{array}{l} \{c(\boldsymbol{s}_1^{\frac{n}{2}-3}),0\},\\ \{c(\boldsymbol{s}_2^{\frac{n}{2}-3}),0,s_{\frac{n}{2}-1}\},\\ \{c(\boldsymbol{s}_3^{\frac{n}{2}-3}),0,c(\boldsymbol{s}_{\frac{n}{2}-1}^{\frac{n}{2}})\},\\ \{c(\boldsymbol{s}_4^{\frac{n}{2}-3}),0,c(\boldsymbol{s}_{\frac{n}{2}-1}^{\frac{n}{2}+1})\},\\ \{c(\boldsymbol{s}_5^{\frac{n}{2}-3}),0,c(\boldsymbol{s}_{\frac{n}{2}-1}^{\frac{n}{2}+2})\},\\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n}),1\},\\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n}),s_{\frac{n}{2}+2}\},\\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-2}),1,c(\boldsymbol{s}_{\frac{n}{2}+1}^{\frac{n}{2}+2})\},\\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-3}),1,c(\boldsymbol{s}_{\frac{n}{2}}^{\frac{n}{2}+2})\},\\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-4}),1,c(\boldsymbol{s}_{\frac{n}{2}-1}^{\frac{n}{2}+2})\} \end{array}\right\} = \left\{\begin{array}{l} \{c(\boldsymbol{v}_1^{\frac{n}{2}-3}),1\},\\ \{c(\boldsymbol{v}_2^{\frac{n}{2}-3}),1,v_{\frac{n}{2}-1}\},\\ \{c(\boldsymbol{v}_3^{\frac{n}{2}-3}),1,c(\boldsymbol{v}_{\frac{n}{2}-1}^{\frac{n}{2}})\},\\ \{c(\boldsymbol{v}_4^{\frac{n}{2}-3}),1,c(\boldsymbol{v}_{\frac{n}{2}-1}^{\frac{n}{2}+1})\},\\ \{c(\boldsymbol{v}_5^{\frac{n}{2}-3}),1,c(\boldsymbol{v}_{\frac{n}{2}-1}^{\frac{n}{2}+2})\},\\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n}),0\},\\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-1}),0,v_{\frac{n}{2}+2}\},\\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-2}),0,c(\boldsymbol{v}_{\frac{n}{2}+1}^{\frac{n}{2}+2})\},\\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-3}),0,c(\boldsymbol{v}_{\frac{n}{2}}^{\frac{n}{2}+2})\},\\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-4}),0,c(\boldsymbol{v}_{\frac{n}{2}-1}^{\frac{n}{2}+2})\} \end{array}\right\}.$$

To proceed with the proof, we assume that (36) holds with equality. When this is not the case, the proof will follow similarly. We let $\alpha = \mathrm{wt}(\boldsymbol{s}_1^{\frac{n}{2}-3})$, and rewrite the preceding set equality in terms of Hamming weights, i.e.,

$$\left\{\begin{array}{l} \alpha,\\ \alpha + s_{\frac{n}{2}-1},\\ \alpha + \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}-1}^{\frac{n}{2}}) - s_2,\\ \alpha + \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}-1}^{\frac{n}{2}+1}) - \mathrm{wt}(\boldsymbol{s}_2^3),\\ \alpha + \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}-1}^{\frac{n}{2}+2}) - \mathrm{wt}(\boldsymbol{s}_2^4),\\ \alpha + 4,\\ \alpha + 3 + s_{\frac{n}{2}+2},\\ \alpha + 3 + \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}+1}^{\frac{n}{2}+2}) - s_{n-1},\\ \alpha + 3 + \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}}^{\frac{n}{2}+2}) - \mathrm{wt}(\boldsymbol{s}_{n-2}^{n-1}),\\ \alpha + 3 + \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}-1}^{\frac{n}{2}+2}) - \mathrm{wt}(\boldsymbol{s}_{n-3}^{n-1}) \end{array}\right\} = \left\{\begin{array}{l} \alpha+1,\\ \alpha + 1 + v_{\frac{n}{2}-1},\\ \alpha + 1 + \mathrm{wt}(\boldsymbol{v}_{\frac{n}{2}-1}^{\frac{n}{2}}) - v_2,\\ \alpha + 1 + \mathrm{wt}(\boldsymbol{v}_{\frac{n}{2}-1}^{\frac{n}{2}+1}) - \mathrm{wt}(\boldsymbol{v}_2^3),\\ \alpha + 1 + \mathrm{wt}(\boldsymbol{v}_{\frac{n}{2}-1}^{\frac{n}{2}+2}) - \mathrm{wt}(\boldsymbol{v}_2^4),\\ \alpha + 3,\\ \alpha + 2 + v_{\frac{n}{2}+2},\\ \alpha + 2 + \mathrm{wt}(\boldsymbol{v}_{\frac{n}{2}+1}^{\frac{n}{2}+2}) - v_{n-1},\\ \alpha + 2 + \mathrm{wt}(\boldsymbol{v}_{\frac{n}{2}}^{\frac{n}{2}+2}) - \mathrm{wt}(\boldsymbol{v}_{n-2}^{n-1}),\\ \alpha + 2 + \mathrm{wt}(\boldsymbol{v}_{\frac{n}{2}-1}^{\frac{n}{2}+2}) - \mathrm{wt}(\boldsymbol{v}_{n-3}^{n-1}) \end{array}\right\}. \tag{37}$$

Naturally, the variable α can be removed from every element in both sets. Upon exhaustively searching for a suitable vector $(\boldsymbol{s}_2^4, \boldsymbol{s}_{n-3}^{n-1}, \boldsymbol{s}_{n/2-1}^{n/2+2}, \boldsymbol{v}_{n/2-1}^{n/2+2}) = (\boldsymbol{v}_2^4, \boldsymbol{v}_{n-3}^{n-1}, \boldsymbol{s}_{n/2-1}^{n/2+2}, \boldsymbol{v}_{n/2-1}^{n/2+2})$ in the space $\{0,1\}^{14}$, we find no solution that satisfies (37) while also upholding the Catalan-Bertrand constraint. Thus, $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ cannot hold.

Subcase (ii): $s_{n/2-2} = v_{n/2-2}$, $s_{n/2-1} \neq v_{n/2-1}$

We assume without loss of generality that $(s_{n/2-1}, v_{n/2-1}) = (0,1)$. As done previously, we investigate whether $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ can hold, while

$$\begin{aligned} (\boldsymbol{s}_1^{n/2-3}, \boldsymbol{s}_{n/2+4}^{n}) &= (\boldsymbol{v}_1^{n/2-3}, \boldsymbol{v}_{n/2+4}^{n}),\\ (s_{n/2-1}, v_{n/2-1}) &= (0,1),\\ (s_{n/2+2}, v_{n/2+2}) &= (1,0),\\ s_{n/2} + s_{n/2+1} &= v_{n/2} + v_{n/2+1}. \end{aligned}$$

These equations have been summarized in Fig. 6. Again, since for any $5 \leq i \leq n/2-1$ as well as for $i \in \{1, n\}$, $c(\boldsymbol{s}_i^{i+n/2-3}) = c(\boldsymbol{v}_i^{i+n/2-3})$, $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ is equivalent to

$$\left\{\begin{array}{l} \{c(\boldsymbol{s}_2^{\frac{n}{2}-2}),0\},\\ \{c(\boldsymbol{s}_3^{\frac{n}{2}-2}),0,s_{\frac{n}{2}}\},\\ \{c(\boldsymbol{s}_4^{\frac{n}{2}-2}),0,c(\boldsymbol{s}_{\frac{n}{2}}^{\frac{n}{2}+1})\},\\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-1}),1\},\\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-2}),1,s_{\frac{n}{2}+1}\},\\ \{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-3}),1,c(\boldsymbol{s}_{\frac{n}{2}}^{\frac{n}{2}+1})\} \end{array}\right\} = \left\{\begin{array}{l} \{c(\boldsymbol{v}_2^{\frac{n}{2}-2}),1\},\\ \{c(\boldsymbol{v}_3^{\frac{n}{2}-2}),1,v_{\frac{n}{2}}\},\\ \{c(\boldsymbol{v}_4^{\frac{n}{2}-2}),1,c(\boldsymbol{v}_{\frac{n}{2}}^{\frac{n}{2}+1})\},\\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-1}),0\},\\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-2}),0,v_{\frac{n}{2}+1}\},\\ \{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-3}),0,c(\boldsymbol{v}_{\frac{n}{2}}^{\frac{n}{2}+1})\} \end{array}\right\}. \tag{38}$$

Akin to (36), we can use similar arguments to reach the following inequality.

$$\mathrm{wt}(\boldsymbol{s}_{n/2+3}^{n}) \geq \mathrm{wt}(\boldsymbol{s}_1^{n/2-2}) + 3.$$

Once again, we assume that the above expression holds with equality, and let $\alpha = \mathrm{wt}(\boldsymbol{s}_1^{n/2-2})$. Now transforming (38) into

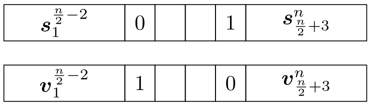

$\boldsymbol{s}_1^{\frac{n}{2}-2}$
0
1
$\boldsymbol{s}_{\frac{n}{2}+3}^{n}$
$\boldsymbol{v}_1^{\frac{n}{2}-2}$
1
0
$\boldsymbol{v}_{\frac{n}{2}+3}^{n}$

Fig. 6. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are such that $(\boldsymbol{s}_1^{\frac{n}{2}-2}, \boldsymbol{s}_{\frac{n}{2}+3}^{n}) = (\boldsymbol{v}_1^{\frac{n}{2}-2}, \boldsymbol{v}_{\frac{n}{2}+3}^{n})$, and $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$.

its corresponding Hamming weights representation, we arrive at

$$\begin{Bmatrix} \alpha, \\ \alpha + s_{\frac{n}{2}} - s_2, \\ \alpha + \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}}^{\frac{n}{2}+1}) - \mathrm{wt}(\boldsymbol{s}_2^3), \\ \alpha + 3, \\ \alpha + 3 + s_{n/2+1} - s_{n-1}, \\ \alpha + 3 + \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}}^{\frac{n}{2}+1}) - \mathrm{wt}(\boldsymbol{s}_{n-2}^{n-1}) \end{Bmatrix}$$
$$= \begin{Bmatrix} \alpha + 1, \\ \alpha + 1 + v_{\frac{n}{2}} - v_2, \\ \alpha + 1 + \mathrm{wt}(\boldsymbol{v}_{\frac{n}{2}}^{\frac{n}{2}+1}) - \mathrm{wt}(\boldsymbol{v}_2^3), \\ \alpha + 2, \\ \alpha + 2 + v_{n/2+1} - v_{n-1}, \\ \alpha + 2 + \mathrm{wt}(\boldsymbol{v}_{\frac{n}{2}}^{\frac{n}{2}+1}) - \mathrm{wt}(\boldsymbol{v}_{n-2}^{n-1}) \end{Bmatrix}.$$

Upon searching for a suitable vector $(\boldsymbol{s}_2^3, \boldsymbol{s}_{n-2}^{n-1}, \boldsymbol{s}_{n/2}^{n/2+1}, \boldsymbol{v}_{n/2}^{n/2+1})$ in the space $\{0,1\}^8$ that satisfies the preceding set equality as well as $s_{\frac{n}{2}} + s_{\frac{n}{2}+1} = v_{\frac{n}{2}} + v_{\frac{n}{2}+1}$ along with the Catalan-Bertrand restriction, we find no viable solutions. Hence, no suitable $\boldsymbol{v}$ exists.

Subcase (iii): $\boldsymbol{s}_{n/2-2}^{n/2-1} = \boldsymbol{v}_{n/2-2}^{n/2-1}$, $s_{n/2} \neq v_{n/2}$

By adopting the approach employed in earlier subcases, we obtain

$$\mathrm{wt}(\boldsymbol{s}_{n/2+2}^{n}) \geq \mathrm{wt}(\boldsymbol{s}_1^{n/2-1}) + 3. \tag{39}$$

We also observe that $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ is equivalent to

$$\begin{Bmatrix} \{c(\boldsymbol{s}_3^{\frac{n}{2}-1}), 0\}, \\ \{c(\boldsymbol{s}_{\frac{n}{2}+2}^{n-2}), 1\} \end{Bmatrix} = \begin{Bmatrix} \{c(\boldsymbol{v}_3^{\frac{n}{2}-1}), 1\}, \\ \{c(\boldsymbol{v}_{\frac{n}{2}+2}^{n-2}), 0\} \end{Bmatrix}, \tag{40}$$

since $c(\boldsymbol{s}_i^{n/2+i-3}) = c(\boldsymbol{v}_i^{n/2+i-3})$ for any $i \in \left[\frac{n}{2}+3\right] \backslash \{3, \frac{n}{2}+2\}$. Evidently, (40) is upheld only when

$$\begin{aligned} &\mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}+2}^{n-2}) = \mathrm{wt}(\boldsymbol{v}_3^{\frac{n}{2}-1}) = \mathrm{wt}(\boldsymbol{s}_3^{\frac{n}{2}-1}) \\ \implies &\mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}+2}^{n-2}) - \mathrm{wt}(\boldsymbol{s}_3^{\frac{n}{2}-1}) = 0 \\ \implies &\mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}+2}^{n}) - \mathrm{wt}(\boldsymbol{s}_1^{\frac{n}{2}-1}) = s_{n-1} + s_n - s_1 - s_2 < 3, \end{aligned}$$

which clearly contradicts (39). Consequently, $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ cannot hold.

Case 2: n may be odd/even and the deleted multisets are not all consecutive, i.e., $k + 1 < n - k + 1$.

In the following, for $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_{DS}^{(2)}(n)$, we define σ_i and σ_i' as

$$\begin{aligned} \sigma_i &= s_i + s_{n-i+1}, \\ \sigma_i' &= v_i + v_{n-i+1}, \end{aligned} \tag{41}$$

for any $1 \leq i \leq \lfloor n/2 \rfloor$. When n is odd, we define $\sigma_{\lceil n/2 \rceil} = s_{\lceil n/2 \rceil}$ and $\sigma'_{\lceil n/2 \rceil} = v_{\lceil n/2 \rceil}$.

Subcase (i): $k = 2$

When the multisets $C_1(\boldsymbol{s}), C_2(\boldsymbol{s}), C_{n-1}(\boldsymbol{s})$ and $C_n(\boldsymbol{s})$ are missing, $\boldsymbol{s}$ can be uniquely recovered only if there exists no other $\boldsymbol{v}$ such that

$$\bigcup_{i \in \{3, \ldots, n-2\}} C_i(\boldsymbol{s}) = \bigcup_{i \in \{3, \ldots, n-2\}} C_i(\boldsymbol{v}). \tag{42}$$

For the purposes of this proof, we assume that such a $\boldsymbol{v}$ indeed exists.

Firstly, since $\mathcal{S}_{DS}^{(2)}(n) \subset \mathcal{S}_R(n)$, we immediately know that $(s_1, s_n) = (v_1, v_n) = (0, 1)$. Next, since $w_3(\boldsymbol{s}) = w_3(\boldsymbol{v})$,

$$\begin{aligned} w_3(\boldsymbol{s}) &= 3w_1(\boldsymbol{s}) - \sigma_2 - 2\sigma_1 \\ &= 3w_1(\boldsymbol{s}) - \sigma_2 - 2 = 3w_1(\boldsymbol{v}) - \sigma_2' - 2 \\ \implies 3(w_1(\boldsymbol{s}) - w_1(\boldsymbol{v})) &= \sigma_2 - \sigma_2'. \end{aligned} \tag{43}$$

Now since $\sigma_i, \sigma_i' \in \{0, 1, 2\}$, it can only be that $w_1(\boldsymbol{s}) = \boldsymbol{v}$ and $\sigma_2 = \sigma_2'$. As a result, we have $C_1(\boldsymbol{s}) = C_1(\boldsymbol{v})$ and $C_n(\boldsymbol{s}) = C_n(\boldsymbol{v})$.

Once again, we may exploit $(s_1, s_n) = (v_1, v_n) = (0, 1)$ in order to further determine $C_{n-1}(\boldsymbol{s}) = C_{n-1}(\boldsymbol{v})$ from the knowledge of $C_n(\boldsymbol{s})$. Thus, (42) can be rewritten to

$$\bigcup_{i \in \{1, 3, \ldots, n\}} C_i(\boldsymbol{s}) = \bigcup_{i \in \{1, 3, \ldots, n\}} C_i(\boldsymbol{v}).$$

Since this expression reflects the case of a single multiset deletion, we conclude from Lemma 3 that $C(\boldsymbol{s}) = C(\boldsymbol{v})$. We infer this directly from the fact that $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_{DS}^{(2)}(n) \subset \mathcal{S}_R(n)$. Thus, $\boldsymbol{s} = \boldsymbol{v}$.

Subcase (ii): $k = 3$

When $k = 3$, multisets $C_2(\boldsymbol{s}), C_3(\boldsymbol{s}), C_{n-2}(\boldsymbol{s})$ and $C_{n-1}(\boldsymbol{s})$ are deleted. Again, since $(s_1, s_n) = (0, 1)$, we can promptly determine $C_{n-1}(\boldsymbol{s})$ from $C_n(\boldsymbol{s})$.

If there exists some $\boldsymbol{v}$ such that

$$\bigcup_{i \in \{1, 4, \ldots, n-3, n-1, n\}} C_i(\boldsymbol{s}) = \bigcup_{i \in \{1, 4, \ldots, n-3, n-1, n\}} C_i(\boldsymbol{v}), \tag{44}$$

then clearly $w_4(\boldsymbol{s}) = w_4(\boldsymbol{v})$ and $\sigma_1 = \sigma_1'$. We can thus deduce from (6) that

$$\begin{aligned} 2\sigma_2 + \sigma_3 &= 2\sigma_2' + \sigma_3', \\ 2(\sigma_2 - \sigma_2') &= \sigma_3' - \sigma_3. \end{aligned} \tag{45}$$

Now using the following expression for any $6 \leq i \leq \lceil \frac{n}{2} \rceil$,

$$2w_{i-1}(\boldsymbol{s}) - w_{i-2}(\boldsymbol{s}) - w_i(\boldsymbol{s}) = \sigma_{i-1}, \tag{46}$$

we can uniquely deduce $(\sigma_5, \ldots, \sigma_{\lceil \frac{n}{2} \rceil - 1})$. We can also recover $\sigma_{\lceil n/2 \rceil}$ by exploiting (5) as follows.

$$\begin{aligned} w_{\lceil \frac{n}{2} \rceil}(\boldsymbol{s}) - w_{\lceil \frac{n}{2} \rceil - 1}(\boldsymbol{s}) &= \sum_{i=1}^{\lceil \frac{n}{2} \rceil - 1} i\sigma_i + \left\lceil \frac{n}{2} \right\rceil \sigma_{\lceil \frac{n}{2} \rceil} \\ &\quad - \sum_{i=1}^{\lceil \frac{n}{2} \rceil - 1} i\sigma_i - \left(\left\lceil \frac{n}{2} \right\rceil - 1 \right) \sigma_{\lceil \frac{n}{2} \rceil} \\ &= \sigma_{\lceil \frac{n}{2} \rceil}. \end{aligned}$$

Since for $4 \leq i \leq \lceil \frac{n}{2} \rceil$, $w_i(\boldsymbol{s}) = w_i(\boldsymbol{v})$, it naturally follows that

$$(\sigma_5, \ldots, \sigma_{\lceil \frac{n}{2} \rceil}) = (\sigma'_5, \ldots, \sigma'_{\lceil \frac{n}{2} \rceil}).$$

Now since $w_1(\boldsymbol{s}) = w_1(\boldsymbol{v})$ and the vector $(\sigma_5, \ldots, \sigma_{\lceil \frac{n}{2} \rceil})$ is known, it must be due to (4) that

$$\sigma_2 + \sigma_3 + \sigma_4 = \sigma'_2 + \sigma'_3 + \sigma'_4. \tag{47}$$

Next, to compute the difference between the sum of cumulative weights of $\boldsymbol{s}$ and $\boldsymbol{v}$,

$$\begin{aligned}\sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{s}) - \sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{v}) &= w_2(\boldsymbol{s}) + w_3(\boldsymbol{s}) - w_2(\boldsymbol{v}) - w_3(\boldsymbol{v}) \\ &= w_3(\boldsymbol{s}) - w_3(\boldsymbol{v}) = \sigma'_2 - \sigma_2,\end{aligned}$$

where the second equality follows from $w_{n-1}(\boldsymbol{s}) = w_{n-1}(\boldsymbol{v})$ and (7), while the final equality is a consequence of (6) and $\sigma_1 = \sigma'_1$. Due to the constraint on the sum of cumulative weights in $\mathcal{S}^{(2)}_{DS}(n)$, it must hold that $\sigma_2 = \sigma'_2$.

Equations (45) and (47) further lead to $(\sigma_2, \sigma_3, \sigma_4) = (\sigma'_2, \sigma'_3, \sigma'_4)$. Now if $\sigma_2 = \sigma'_2 = 1$ and $s_2 \neq v_2$, then Lemma 2 allows us to infer that $C_{n-3}(\boldsymbol{s}) \neq C_{n-3}(\boldsymbol{v})$, which is not possible according to (44). Thus, no suitable $\boldsymbol{v}$ exists.

Subcase (iii): $k \geq 4$

From the proof of Lemma 9, we note that when the multisets $\{C_{k-1}(\boldsymbol{s}), C_k(\boldsymbol{s}), C_{n-k+1}(\boldsymbol{s}), C_{n-k+2}(\boldsymbol{s})\}$ are deleted, there may exist an alternate $\boldsymbol{v}$ such that:

$$\begin{aligned}(\boldsymbol{s}_1^{k-3}, \boldsymbol{s}_{n-k+4}^n) &= (\boldsymbol{v}_1^{k-3}, \boldsymbol{v}_{n-k+4}^n) \\ \sigma_i &= \sigma'_i, \qquad \forall\, i \in I \\ \sigma_k + 2\sigma_{k-1} + 3\sigma_{k-2} &= \sigma'_k + 2\sigma'_{k-1} + 3\sigma'_{k-2}, \\ \sigma_{k+1} + \sigma_k + \sigma_{k-1} + \sigma_{k-2} &= \sigma'_{k+1} + \sigma'_k + \sigma'_{k-1} + \sigma'_{k-2}.\end{aligned}$$

where $I = [\lceil \frac{n}{2} \rceil] \backslash \{k-2, \ldots, k+1\}$. As before, we bound the difference of the sum of cumulative weights of $\boldsymbol{s}$ and $\boldsymbol{v}$.

$$\begin{aligned}\sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{s}) - \sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{v}) &= \sum_{i=k-1}^{k} w_i(\boldsymbol{s}) - \sum_{i=k-1}^{k} w_i(\boldsymbol{v}) \\ &= (\sigma'_{k-1} - \sigma_{k-1}) \\ &\quad +3(\sigma'_{k-2} - \sigma_{k-2}).\end{aligned} \tag{48}$$

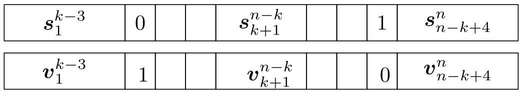
We find through numerical verification that this quantity cannot exceed 5, and it precisely occurs when:

$$\begin{aligned}(\sigma_{k-2}, \sigma_{k-1}, \sigma_k, \sigma_{k+1}) &= (0, 2, 0, 0), \\ (\sigma'_{k-2}, \sigma'_{k-1}, \sigma'_k, \sigma'_{k+1}) &= (1, 0, 1, 0).\end{aligned}$$

Therefore, when $\boldsymbol{s} \in \mathcal{S}^{(2)}_{DS}(n)$, the constraint on $\sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{s})$ helps us recover $\boldsymbol{\sigma}_s$ completely. The next logical step is to investigate whether there exists some $\boldsymbol{v} \in \mathcal{S}^{(2)}_{DS}(n)$, for such an $\boldsymbol{s}$, such that for any $i \in [n] \backslash \{k-1, k, n-k+1, n-k+2\}$, where $k \geq 3$,

$$\begin{gathered} C_i(\boldsymbol{s}) = C_i(\boldsymbol{v}), \\ \boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v, \\ (\boldsymbol{s}_1^{k-3}, \boldsymbol{s}_{n-k+4}^n) = (\boldsymbol{v}_1^{k-3}, \boldsymbol{v}_{n-k+4}^n). \end{gathered}$$

Note that since $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$, we can infer that $c(\boldsymbol{s}_{k+1}^{n-k}) = c(\boldsymbol{v}_{k+1}^{n-k})$. In the following, we denote $c(\boldsymbol{s}_{k+1}^{n-k})$ as c.



$\boldsymbol{s}_1^{k-3}$	0			$\boldsymbol{s}_{k+1}^{n-k}$			1	$\boldsymbol{s}_{n-k+4}^n$
$\boldsymbol{v}_1^{k-3}$	1			$\boldsymbol{v}_{k+1}^{n-k}$			0	$\boldsymbol{v}_{n-k+4}^n$

Fig. 7. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are related such that $(\boldsymbol{s}_1^{k-3}, \boldsymbol{s}_{n-k+4}^n) = (\boldsymbol{v}_1^{k-3}, \boldsymbol{v}_{n-k+4}^n)$ and $c(\boldsymbol{s}_{k+2}^{n-k-1}) = c(\boldsymbol{v}_{k+2}^{n-k-1})$.

Let $s_{k-2} \neq v_{k-2}$. Without loss of generality, we assume $(s_{k-2}, v_{k-2}) = (0, 1)$. We illustrate this in Fig. 7. As done previously in (36), we infer that

$$\mathrm{wt}(\boldsymbol{s}_{n-k+4}^n) \geq \mathrm{wt}(\boldsymbol{s}_1^{k-3}) + 3.$$

It is assumed that this expression holds with equality, since in other cases, the proof will follow similarly. For notational convenience, we let $\alpha = \mathrm{wt}(\boldsymbol{s}_1^{k-3}) = \mathrm{wt}(\boldsymbol{v}_1^{k-3})$.

Now for $C_{n-k}(\boldsymbol{s}) = C_{n-k}(\boldsymbol{v})$ to hold, we essentially require that

$$\begin{aligned}&\left\{\begin{aligned}&\{c(\boldsymbol{s}_1^{k-3}), 0, c(\boldsymbol{s}_{k-1}^k), c\}, \\ &\{c(\boldsymbol{s}_2^{k-3}), 0, c(\boldsymbol{s}_{k-1}^k), c, s_{n-k+1}\}, \\ &\{c(\boldsymbol{s}_3^{k-3}), 0, c(\boldsymbol{s}_{k-1}^k), c, c(\boldsymbol{s}_{n-k+1}^{n-k+2})\}, \\ &\{c(\boldsymbol{s}_{n-k+4}^n), 1, c(\boldsymbol{s}_{n-k+1}^{n-k+2}), c\}, \\ &\{c(\boldsymbol{s}_{n-k+4}^{n-1}), 1, c(\boldsymbol{s}_{n-k+1}^{n-k+2}), c, s_k\}, \\ &\{c(\boldsymbol{s}_{n-k+4}^{n-1}), 1, c(\boldsymbol{s}_{n-k+1}^{n-k+2}), c, c(\boldsymbol{s}_{k-1}^k)\}\end{aligned}\right\} \\ &= \left\{\begin{aligned}&\{c(\boldsymbol{v}_1^{k-3}), 1, c(\boldsymbol{v}_{k-1}^k), c\}, \\ &\{c(\boldsymbol{v}_2^{k-3}), 1, c(\boldsymbol{v}_{k-1}^k), c, v_{n-k+1}\}, \\ &\{c(\boldsymbol{v}_3^{k-3}), 1, c(\boldsymbol{v}_{k-1}^k), c, c(\boldsymbol{v}_{n-k+1}^{n-k+2})\}, \\ &\{c(\boldsymbol{v}_{n-k+4}^n), 0, c(\boldsymbol{v}_{n-k+1}^{n-k+2}), c\}, \\ &\{c(\boldsymbol{v}_{n-k+4}^{n-1}), 0, c(\boldsymbol{v}_{n-k+1}^{n-k+2}), c, v_k\}, \\ &\{c(\boldsymbol{v}_{n-k+4}^{n-1}), 0, c(\boldsymbol{v}_{n-k+1}^{n-k+2}), c, c(\boldsymbol{v}_{k-1}^k)\}\end{aligned}\right\},\end{aligned} \tag{49}$$

since $c(\boldsymbol{s}_i^{n-k+i-1}) = c(\boldsymbol{v}_i^{n-k+i-1})$ for any $i \in \{4, \ldots, k-2\}$. By recalling that $s_1 = v_1 = 0$ and $s_n = v_n = 1$, eliminating c from all elements in both sets and transforming the preceding set equality into its corresponding Hamming weights representation, we acquire

$$\begin{aligned}&\left\{\begin{aligned}&\alpha + s_{k-1} + s_k, \\ &\alpha + s_{k-1} + s_k + s_{n-k+1}, \\ &\alpha + s_{k-1} + s_k + s_{n-k+1} + s_{n-k+2} - s_2, \\ &\alpha + 4 + s_{n-k+1} + s_{n-k+2}, \\ &\alpha + 3 + s_{n-k+1} + s_{n-k+2} + s_k, \\ &\alpha + 3 + s_{n-k+1} + s_{n-k+2} + s_k + s_{k-1} - s_{n-1}\end{aligned}\right\} \\ &= \left\{\begin{aligned}&\alpha + 1 + v_{k-1} + v_k, \\ &\alpha + 1 + v_{k-1} + v_k + v_{n-k+1}, \\ &\alpha + 1 + v_{k-1} + v_k + v_{n-k+1} + v_{n-k+2} - v_2, \\ &\alpha + 3 + v_{n-k+1} + v_{n-k+2}, \\ &\alpha + 2 + v_{n-k+1} + v_{n-k+2} + v_k, \\ &\alpha + 2 + v_{n-k+1} + v_{n-k+2} + v_k + v_{k-1} - v_{n-1}\end{aligned}\right\}.\end{aligned}$$

Next, we search for a possible solution for the vector $(s_2 = v_2, s_{n-1} = v_{n-1}, \boldsymbol{s}_{k-1}^k, \boldsymbol{v}_{k-1}^k, \boldsymbol{s}_{n-k+1}^{n-k+2}, \boldsymbol{v}_{n-k+1}^{n-k+2})$ in the space

$\{0,1\}^{10}$, such that the preceding set equality as well as the following constraints, as mandated by $\boldsymbol{\sigma}_s = \boldsymbol{\sigma}_v$, are satisfied.

$$s_{k-1} + s_{n-k+2} = v_{k-1} + v_{n-k+2},$$
$$s_k + s_{n-k+1} = v_k + v_{n-k+1}.$$

We find only one possible solution, which requires $(s_2, s_{n-1}) = (1, 0)$. Evidently, this is prohibited by the Catalan-Bertrand constraint in $\mathcal{S}_{DS}^{(2)}(n)$.

When $s_{k-2} = v_{k-2}$ and $s_{k-1} \neq v_{k-1}$, Lemma 2 implies that $C_{n-k}(\boldsymbol{s}) \neq C_{n-k}(\boldsymbol{v})$. The same applies for the case when $\boldsymbol{s}_{k-2}^{k-1} = \boldsymbol{v}_{k-2}^{k-1}$ and $s_k \neq v_k$. Thus, an appropriate $\boldsymbol{v}$ does not exist, and $\boldsymbol{s}$ can be recovered uniquely.

Case 3: n may be odd/even and deleted multisets are $C_{k_1}, C_{k_2}, C_{n-k_1+1}, C_{n-k_2+1}$, where $k_2 > k_1 + 1$.

In this case, every pair of deleted multisets is separated by one or more multisets which are uncorrupted. Without loss of generality, assume that $k_2 < n - k_2 + 1$. Since the reconstruction process has access to the multisets $C_n(\boldsymbol{s}), \ldots, C_{n-k_1+2}(\boldsymbol{s})$, the prefix-suffix pair $(\boldsymbol{s}_1^{k_1-2}, \boldsymbol{s}_{n-k_1+3}^{n})$ can be uniquely determined. Now, there remains an ambiguity regarding the assignment of bits $(s_{k_1-1}, s_{n-k-1+2})$. To verify if an incorrect assignment of these bits leads to a conflict with the multiset $C_{n-k_1}(\boldsymbol{s})$, we assume that there exists some $\boldsymbol{v} \in \mathcal{S}_{DS}^{(2)}(n)$ such that for any $i \in \{n, n-1, \ldots, n-k_1+2, n-k_1\}$,

$$C_i(\boldsymbol{s}) = C_i(\boldsymbol{v}),$$
$$(\boldsymbol{s}_1^{k_1-2}, \boldsymbol{s}_{n-k_1+3}^{n}) = (\boldsymbol{v}_1^{k_1-2}, \boldsymbol{v}_{n-k_1+3}^{n}).$$

Due to the second relation, it is easy to see that $(\sigma_1, \ldots, \sigma_{k_1-2}) = (\sigma_1', \ldots, \sigma_{k_1-2}')$, where σ_i and σ_i' obey their previous definitions in (41). Now since $w_{k_1+1}(\boldsymbol{s}) = w_{k_1+1}(\boldsymbol{v})$, we can infer the following as we did earlier in (45) using (6).

$$2(\sigma_{k_1-1} - \sigma_{k_1-1}') = \sigma_{k_1}' - \sigma_{k_1}.$$

This leads to two possibilities for $(\sigma_{k_1-1}, \sigma_{k_1-1}', \sigma_{k_1}, \sigma_{k_1}')$.

- $(\sigma_{k_1-1}, \sigma_{k_1}) = (1, 0)$ and $(\sigma_{k_1-1}', \sigma_{k_1}') = (0, 2)$.
- $(\sigma_{k_1-1}, \sigma_{k_1}) = (2, 0)$ and $(\sigma_{k_1-1}', \sigma_{k_1}') = (1, 2)$.

For both of these potential solutions, the following holds due to (4) and $w_1(\boldsymbol{s}) = w_1(\boldsymbol{v})$.

$$\begin{aligned} \mathrm{wt}(\boldsymbol{s}_{k_1+1}^{n-k_1}) - \mathrm{wt}(\boldsymbol{v}_{k_1+1}^{n-k_1}) &= \sum_{i=k_1+1}^{\lceil \frac{n}{2} \rceil} (\sigma_i - \sigma_i') \\ &= w_1(\boldsymbol{s}) - w_1(\boldsymbol{v}) + \sum_{i=1}^{k_1} (\sigma_i' - \sigma_i) \\ &= \sum_{i=k_1-1}^{k_1} (\sigma_i' - \sigma_i) = 1. \end{aligned}$$

Additionally, since we always have $\sigma_{k_1} = 0$ and $\sigma_{k_1}' = 2$, we may set $s_{k_1} = s_{n-k_1+1} = 0$ and $v_{k_1} = v_{n-k_1+1} = 1$. Now similar to the approach in prior cases, we note that $C_{n-k_1}(\boldsymbol{s}) = C_{n-k_1}(\boldsymbol{v})$ is only upheld when

$$\begin{aligned} &\{c(\boldsymbol{s}_1^{n-k_1}), c(\boldsymbol{s}_2^{n-k_1+1}), c(\boldsymbol{s}_{k_1+1}^{n}), c(\boldsymbol{s}_{k_1}^{n-1})\} \\ &= \{c(\boldsymbol{v}_1^{n-k_1}), c(\boldsymbol{v}_2^{n-k_1+1}), c(\boldsymbol{v}_{k_1+1}^{n}), c(\boldsymbol{v}_{k_1}^{n-1})\}. \end{aligned}$$

After plugging in the values of $(s_{k_1}, s_{n-k_1+1}, v_{k_1}, v_{n-k_1+1})$ and transforming this set equality into its equivalent representation in terms of Hamming weights, we obtain

$$\begin{Bmatrix} \mathrm{wt}(\boldsymbol{s}_1^{k_1-2}) + s_{k_1-1} + \mathrm{wt}(\boldsymbol{s}_{k_1-1}^{n-k_1}), \\ \mathrm{wt}(\boldsymbol{s}_2^{k_1-2}) + s_{k_1-1} + \mathrm{wt}(\boldsymbol{s}_{k_1-1}^{n-k_1}), \\ \mathrm{wt}(\boldsymbol{s}_{n-k_1+3}^{n}) + s_{n-k_1+2} + \mathrm{wt}(\boldsymbol{s}_{k_1-1}^{n-k_1}), \\ \mathrm{wt}(\boldsymbol{s}_{n-k_1+3}^{n-1}) + s_{n-k_1+2} + \mathrm{wt}(\boldsymbol{s}_{k_1-1}^{n-k_1}) \end{Bmatrix}$$
$$= \begin{Bmatrix} \mathrm{wt}(\boldsymbol{v}_1^{k_1-2}) + v_{k_1-1} + \mathrm{wt}(\boldsymbol{v}_{k_1-1}^{n-k_1}), \\ \mathrm{wt}(\boldsymbol{v}_2^{k_1-2}) + v_{k_1-1} + \mathrm{wt}(\boldsymbol{v}_{k_1-1}^{n-k_1}), \\ \mathrm{wt}(\boldsymbol{v}_{n-k_1+3}^{n}) + v_{n-k_1+2} + \mathrm{wt}(\boldsymbol{v}_{k_1-1}^{n-k_1}), \\ \mathrm{wt}(\boldsymbol{v}_{n-k_1+3}^{n-1}) + v_{n-k_1+2} + \mathrm{wt}(\boldsymbol{v}_{k_1-1}^{n-k_1}) \end{Bmatrix},$$

Finally, we attempt to find a possible solution for the vector $(s_{k_1-1}, s_{n-k_1+2}, v_{k_1-1}, v_{n-k_1+2})$ in the space $\{0,1\}^4$ that satisfies the preceding expression along with the fact that $\sigma_{k_1-1} - \sigma_{k_1-1}' = 1$. None of the feasible solutions agree with the Catalan-Bertrand constraint imposed by $\mathcal{S}_R(n)$.

Thus, an incorrect assignment of the bits $(s_{k_1-1}, s_{k_1}, s_{n-k_1+1}, s_{n-k_1+2})$ leads to a mismatch with multiset $C_{n-k_1}(\boldsymbol{s})$. Consequently, we can recover these bits, as well as $C_{n-k_1+1}(\boldsymbol{s})$ uniquely. The absence of multisets $C_{n-k_2+1}(\boldsymbol{s})$ and $C_{k_2}(\boldsymbol{s})$ can also be corrected due to Lemma 3.

□

We now seek to generalize the coding constraints in $\mathcal{S}_{DS}^{(2)}(m)$ in (10) by examining how the required redundancy scales as more multiset pairs go missing. This is accomplished by $\mathcal{S}_{DS}^{(t)}(n)$ [see Construction 4]. Theorem 5 demonstrates that $\mathcal{S}_{DS}^{(t)}(n)$ is a t-symmetric consecutive multiset deletion code. The proof commences with the following lemma.

Lemma 6: For an even n, consider any $\boldsymbol{s} \in \mathcal{S}_{DS}^{(t)}(n)$, the composition multiset of which suffers the deletion of up to t symmetric multisets, i.e., for some $\mathcal{I} \subset [\frac{n}{2}]$ where $|\mathcal{I}| \leq t$, $C(\boldsymbol{s})$ is modified to $C'(\boldsymbol{s}) = \bigcup_{i \in [\frac{n}{2}] \setminus \mathcal{I}} \widetilde{C}_i(\boldsymbol{s})$. From $C'(\boldsymbol{s})$, $\boldsymbol{\sigma}_s$ can be recovered uniquely.

Proof: Let $\mathcal{I} = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$, such that $i_1 < i_2 < \ldots < i_{|I|}$.

Case 1: $i_1 + 3 < i_2, \ldots i_{|\mathcal{I}|-1} + 3 < i_{|\mathcal{I}|}$.

Since the reconstruction algorithm has access to the multisets $C_n(\boldsymbol{s}), \ldots, C_{n-i_1+2}(\boldsymbol{s})$, we can unambiguously determine $(\sigma_1, \ldots, \sigma_{i_1-2})$. Now since $i_1 + 3 < i_2$, we have access to the cumulative weights $w_{i_1+1}(\boldsymbol{s}), w_{i_1+2}(\boldsymbol{s})$ and $w_{i_1+3}(\boldsymbol{s})$, which help us evaluate σ_{i_1+2} by virtue of (46). Additionally, we can infer the values of $2\sigma_{i_1-1} + \sigma_{i_1}$ and $\sigma_{i_1-1} + \sigma_{i_1} + \sigma_{i_1+1}$ from the knowledge of $w_{i_1+1}(\boldsymbol{s})$, $w_{i_1+2}(\boldsymbol{s})$ and $(\sigma_1, \ldots, \sigma_{i_1-2})$, by exploiting (6).

Subcase (i): $i_{|\mathcal{I}|} \leq \frac{n}{2} - 3$

In a similar fashion, we can also determine $(\sigma_{i_1+3}, \ldots, \sigma_{i_2-2})$. Repeating this process for each $i \in \mathcal{I}$, we conclude that the only unknown values in $\boldsymbol{\sigma}_s$ are $(\sigma_{i_1-1}, \sigma_{i_1}, \sigma_{i_1+1}, \ldots, i_{|\mathcal{I}|-1}, i_{|\mathcal{I}|}, i_{|\mathcal{I}|+1})$, i.e. $3|\mathcal{I}|$ elements in $\boldsymbol{\sigma}_s$ have been erased. Since $|\mathcal{I}| \leq t$ and $\boldsymbol{\sigma}_s$ belongs to a code that corrects up to $3t$ erasures, we can uniquely recover $\boldsymbol{\sigma}_s$.

Subcase (ii): $i_{|\mathcal{I}|} = \frac{n}{2} - 2$

Proceeding similarly to the previous subcase, we can determine $(\sigma_1, \ldots, \sigma_{i_1-2}, \sigma_{i_1+2}, \ldots, \sigma_{i_{|\mathcal{I}|-1}-2}, \sigma_{i_{|\mathcal{I}|-1}+2}, \ldots, \sigma_{i_{|\mathcal{I}|}-2})$, where $\sigma_{i_{|\mathcal{I}|}-2} = \sigma_{\frac{n}{2}-4}$. In addition, we can also determine $\sigma_{\frac{n}{2}}$ from the difference between $w_{\frac{n}{2}}(\boldsymbol{s})$ and $w_{\frac{n}{2}-1}(\boldsymbol{s})$. However, $\sigma_{\frac{n}{2}-3}$, $\sigma_{\frac{n}{2}-2}$ and $\sigma_{\frac{n}{2}-1}$ remain unknown. Hence, once again $3|\mathcal{I}|$ erasures in $\boldsymbol{\sigma}_s$ need to be corrected, which is permitted by the construction of $\mathcal{S}_{DS}^{(t)}(n)$.

Subcase (iii): $\frac{n}{2} - 1 \leq i_{|\mathcal{I}|} \leq \frac{n}{2}$.

By employing arguments used earlier, we can infer that when $i_{|\mathcal{I}|} = \frac{n}{2} - 1$, we are unable to deduce $\sigma_{i_{|\mathcal{I}|}-1}$, $\sigma_{i_{|\mathcal{I}|}}$ and $\sigma_{i_{|\mathcal{I}|}+1}$, in addition to the subvectors $(\sigma_{i_1-1}, \sigma_{i_1}, \sigma_{i_1+1})$, $(\sigma_{i_2-1}, \sigma_{i_2}, \sigma_{i_2+1})$ and so on, up to $(\sigma_{i_{|\mathcal{I}|-1}-1}, \sigma_{i_{|\mathcal{I}|-1}}, \sigma_{i_{|\mathcal{I}|-1}+1})$. Similarly, when $i_{|\mathcal{I}|} = \frac{n}{2}$, the complete list of erasures in $\boldsymbol{\sigma}_s$ is given by $\sigma_{i_1-1}, \sigma_{i_1}, \sigma_{i_1+1}, \ldots, \sigma_{i_{|\mathcal{I}|-1}-1}, \sigma_{i_{|\mathcal{I}|-1}}, \sigma_{i_{|\mathcal{I}|-1}+1}, \sigma_{i_{|\mathcal{I}|}-1}$ and $\sigma_{i_{|\mathcal{I}|}}$. For both situations, the total number of erasures does not exceed $3|\mathcal{I}|$. Hence once again, $\boldsymbol{\sigma}_s$ can be uniquely recovered.

Case 2: $i_j + 3 \geq i_{j+1}$, for at least one $j \leq |\mathcal{I}| - 1$.

Assume there exists only one such value of j. When multiple such indices exist, the proof follows similarly.

Consider $i_j + 3 = i_{j+1}$. Since for $2 \leq k \leq j$, we have $i_{k-1} + 3 < i_k$, we can easily deduce that none of the elements in $(\sigma_{i_1-1}, \sigma_{i_1}, \sigma_{i_1+1}, \ldots, \sigma_{i_2-1}, \sigma_{i_2}, \sigma_{i_2+1}, \ldots, \sigma_{i_j-1}, \sigma_{i_j}, \sigma_{i_j+1})$ can be recovered.

Additionally, since $w_{i_j+3}(\boldsymbol{s})$ is also inaccessible, we are unable to compute $\sigma_{i_j+2} = \sigma_{i_{j+1}-1}$, along with σ_{i_j+3} and σ_{i_j+4}. However, due to $i_{j+1} + 3 = i_j + 6 < i_{j+2}$, we can determine the values $(\sigma_{i_j+5} = \sigma_{i_{j+1}+2}, \ldots, \sigma_{i_{j+2}-2})$. For the remaining deleted multiset pairs $\widetilde{C}_{i_{j+2}}(\boldsymbol{s}), \ldots, \widetilde{C}_{i_{|\mathcal{I}|}}(\boldsymbol{s})$, the unrecoverable elements of $\boldsymbol{\sigma}_s$ are $(\sigma_{i_{j+2}-1}, \sigma_{i_{j+2}}, \sigma_{i_{j+2}+1}, \ldots, \sigma_{i_{|\mathcal{I}|}-1}, \sigma_{i_{|\mathcal{I}|}}, \sigma_{i_{|\mathcal{I}|}+1})$. Therefore, the total number of erasures in $\boldsymbol{\sigma}_s$ equals $3j + 3 + 3(|\mathcal{I}| - j - 1) = 3|\mathcal{I}| \leq 3t$.

When $i_j + 1 \leq i_{j+1} \leq i_j + 2$, we can similarly establish that the number of erasures in $\boldsymbol{\sigma}_s$ does not exceed $3t$.

Consequently, for any configuration of $\mathcal{I}$, at least $\frac{n}{2} - 3t$ elements of $\boldsymbol{\sigma}_s$ can always be recovered, and owing to the constraint on $\boldsymbol{\sigma}_s$ imposed by $\mathcal{S}_{DS}^{(t)}(n)$, we can uniquely determine each element in $\boldsymbol{\sigma}_s$. □

Theorem 5: $\mathcal{S}_{DS}^{(t)}(n)$ is a t-symmetric multiset deletion code.

Proof: In the following, we show that the statement of the theorem holds for even values of n. The proof for odd n will run in a similar fashion. In particular, we wish to show that for any $\boldsymbol{s} \in \mathcal{S}_{DS}^{(t)}(n)$, where $t \geq 2$ and $n > 6t$, if one is given a corrupted composition multiset $C'(\boldsymbol{s}) = \bigcup_{i \in [\frac{n}{2}] \setminus \mathcal{I}} \widetilde{C}_i(\boldsymbol{s})$ where $\mathcal{I} \subset [\frac{n}{2}]$ and $|\mathcal{I}| \leq t$, then $\boldsymbol{s}$ can be uniquely reconstructed.

As suggested by Lemma 6, $\boldsymbol{\sigma}_s$ can be fully and unambiguously recovered, despite the deletion of up to t symmetric multiset pairs. Let $\mathcal{I} = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$, such that $i_1 < i_2 < \ldots < i_{|\mathcal{I}|}$, and $i_{|\mathcal{I}|} \leq \frac{n}{2}$.

Note that whenever $\mathcal{I} \cap \{\frac{n}{2} - t, \ldots, \frac{n}{2}\} \neq \emptyset$, then for any $j \in \mathcal{I} \cap \{\frac{n}{2} - t, \ldots, \frac{n}{2}\}$, we can assign the bits (s_j, s_{n-j+1}) without any ambiguity since the construction of $\mathcal{S}_{DS}^{(t)}(n)$ ensures that σ_j is known and corresponds to a single possibility of (s_j, s_{n-j+1}).

It is also worth pointing out that since $\boldsymbol{\sigma}_s$ is always recoverable, we can also deduce $w_1(\boldsymbol{s})$ from it using (4), implying that $C_1(\boldsymbol{s})$ and $C_n(\boldsymbol{s})$ can also be inferred from the knowledge of $\boldsymbol{\sigma}_s$. Additionally, since $(s_1, s_n) = (0, 1)$, $C_{n-1}(\boldsymbol{s})$ and in turn $w_{n-1}(\boldsymbol{s}) = w_2(\boldsymbol{s})$ can be directly determined from $C_n(\boldsymbol{s})$.[7] Thus, in the following analysis, we ignore the cases where $1 \leq i_1 \leq 2$.

Case 1: $i_1 + 1 = i_2, \ldots, i_{|\mathcal{I}|-1} + 1 = i_{|\mathcal{I}|}$, $i_1 \geq 3$.

This case corresponds to a burst of symmetric multiset deletions. More specifically, the multiset pairs $\widetilde{C}_{i_1}(\boldsymbol{s}), \widetilde{C}_{i_1+1}(\boldsymbol{s}), \ldots, \widetilde{C}_{i_{|\mathcal{I}|}}(\boldsymbol{s})$ are removed.

With the observation that $\boldsymbol{\sigma}_s$ is known and that only the multisets $C_n(\boldsymbol{s}), C_{n-1}(\boldsymbol{s}), \ldots, C_{\frac{n}{2}}(\boldsymbol{s})$ are relevant to the reconstruction algorithm, we recognize that the prefix-suffix pair $(\boldsymbol{s}_1^{i_1-2}, \boldsymbol{s}_{n-i_1+3}^n)$ can be uniquely deduced from $C_n(\boldsymbol{s}), \ldots, C_{n-i_1+2}(\boldsymbol{s})$, which are unaffected by deletion errors.

The next available multiset that the reconstruction algorithm will be able to access, is $C_{n-i_{|\mathcal{I}|}}(\boldsymbol{s}) = C_{n-i_1-|\mathcal{I}|+1}(\boldsymbol{s})$. Recall from Example 4 that for any $\boldsymbol{v} \in \mathcal{S}_R(n)$, the bits (v_j, v_{n-j+1}) can be inferred from the knowledge of prefix-suffix pair $(\boldsymbol{v}_1^{j-1}, \boldsymbol{v}_{n-j+2}^n)$, together with the compositions $c(\boldsymbol{s}_1^j)$ and $c(\boldsymbol{s}_{n-j+1}^n)$, which are extracted from multiset $C_{n-j}(\boldsymbol{s})$.

Subcase (i): $\mathcal{I} \cap \{\frac{n}{2} - t, \ldots, \frac{n}{2}\} = \emptyset$.

The absence of multisets $C_{n-i_1+1}(\boldsymbol{s}), \ldots, C_{n-i_{|\mathcal{I}|}+1}(\boldsymbol{s})$ hinders our ability to immediately determine the bits $(\boldsymbol{s}_{i_1-1}^{i_{|\mathcal{I}|}-1}, \boldsymbol{s}_{n-i_1+2}^{n-i_{|\mathcal{I}|}+2})$. Now, if $(\sigma_{i_1-1}, \ldots, \sigma_{i_{|\mathcal{I}|}-1}) \in \{0, 2\}^{|\mathcal{I}|}$, there exists no ambiguity in the assignment of $(\boldsymbol{s}_{i_1-1}^{i_{|\mathcal{I}|}-1}, \boldsymbol{s}_{n-i_1+2}^{n-i_{|\mathcal{I}|}+2})$. But when this is not the case, i.e., there is at least one $j \in [|\mathcal{I}|]$ such that $\sigma_{i_j-1} = 1$, then on account of Lemma 4 and the fact that $\mathcal{S}_{DS}^{(t)}(n) \subset \mathcal{S}_{DA}^{(t)}(n)$, we conclude that an incorrect assignment of the bits (s_{i_j-1}, s_{n-i_j+2}) will lead to an inconsistency with multiset $C_{n-i_{|\mathcal{I}|}}(\boldsymbol{s})$. Thus, the bits $(\boldsymbol{s}_{i_1-1}^{i_{|\mathcal{I}|}-1}, \boldsymbol{s}_{n-i_1+2}^{n-i_{|\mathcal{I}|}+2})$ as well as the multisets $C_{n-i_1+1}(\boldsymbol{s}), \ldots, C_{n-i_{|\mathcal{I}|}+1}(\boldsymbol{s})$ can be recovered uniquely. Following this, the reconstruction algorithm may proceed normally and $\boldsymbol{s}$ can be fully reconstructed.

Subcase (ii): $i_1 = \frac{n}{2} - t + 1$.

Under this subcase, the multisets $\widetilde{C}_{\frac{n}{2}-t+1}, \ldots, \widetilde{C}_{\frac{n}{2}-t+|\mathcal{I}|}$ are deleted. As argued previously, the prefix-suffix pair $(\boldsymbol{s}_1^{\frac{n}{2}-t-1}, \boldsymbol{s}_{\frac{n}{2}+t+2}^n)$ can be deduced from the multisets $C_n(\boldsymbol{s}), \ldots, C_{\frac{n}{2}+t+1}(\boldsymbol{s})$. Now since $\sigma_{n-t}, \ldots \sigma_{\frac{n}{2}}$ are known, we can directly assign the bits $(\boldsymbol{s}_{\frac{n}{2}-t}^{\frac{n}{2}}, \boldsymbol{s}_{\frac{n}{2}+1}^n)$, thereby completing the reconstruction of $\boldsymbol{s}$.

When $i_1 > \frac{n}{2} - t + 1$ or in more general cases of $\mathcal{I} \cap \{\frac{n}{2} - t, \ldots, \frac{n}{2}\} \neq \emptyset$, the proof runs along similar lines.

Case 2: $i_1 + 1 = i_2, \ldots, i_{j-1} + 1 = i_j$, $i_j + 1 < i_{j+1}$, where $j \geq 1$ and $i_1 \geq 3$.

The reconstruction algorithm is able to infer the prefix-suffix pair $(\boldsymbol{s}_1^{i_1-2}, \boldsymbol{s}_{n-i_1+3}^n)$ from the multisets $C_n(\boldsymbol{s}), \ldots, C_{n-i_1+2}(\boldsymbol{s})$, and halts due to the absence of multisets $C_{n-i_1+1}(\boldsymbol{s}), \ldots, C_{n-i_j+1}(\boldsymbol{s}) = C_{n-i_1-j+2}(\boldsymbol{s})$,

[7] $C_2(\boldsymbol{s})$ may still be unknown, but is irrelevant to the reconstruction algorithm.

thus interrupting the recovery of bits $(\boldsymbol{s}_{i_1-1}^{i_1+j-2}, \boldsymbol{s}_{n-i_1-j+3}^{n-i_1+2})$. Now if $(\sigma_{i_1-1}, \ldots, \sigma_{i_j+j-2}) \in \{0,2\}^j$, then we can assign these bits directly and without any uncertainty. On the contrary, if for at least one $i_1 - 1 \leq p \leq i_j + j - 2$ we have $\sigma_p = 1$ and the bits (s_p, s_{n-p+1}) are assigned incorrectly, then an incompatibility with the multiset $C_{n-i_1-j+1}(\boldsymbol{s})$ occurs, as implied by Lemma 2. Thus, the prefix-suffix pair $(\boldsymbol{s}_1^{i_1+j-2}, \boldsymbol{s}_{n-i_1-j+3}^{n})$ can be uniquely recovered.

By applying this line of reasoning repeatedly for the deleted multisets $\widetilde{C}_{i_{j+1}}(\boldsymbol{s}), \ldots, \widetilde{C}_{i_{|\mathcal{I}|}}(\boldsymbol{s})$, we can arrive at the statement of the lemma.

□

Lemma 7: The code $\mathcal{S}_{DS}^{(t)}$ requires at most $3t\log_2 n + 4t + \frac{1}{2}\log_2(n - 4t - 2) - \log_2(t+1) + 6$ bits of redundancy.

Proof: From the specification of $\mathcal{S}_{DS}^{(t)}$ in (11), we observe that $(\boldsymbol{s}_1^{\frac{n}{2}-t-1}, \boldsymbol{s}_{\frac{n}{2}+t+2}^{n}) \in \mathcal{S}_{DA}^{(t)}$, while the substring $\boldsymbol{s}_{\frac{n}{2}-t}^{\frac{n}{2}+t+1}$ can assume up to $3(t+1)$ possible binary vectors. Furthermore, we require $3t\log n$ bits of redundancy to ensure that for any $\boldsymbol{s} \in \mathcal{S}_{DS}^{(t)}$, $\boldsymbol{\sigma}_s$ belongs to a code capable of correcting up to $3t$ erasures.

After some algebraic manipulation of this expression, we find that the required number of redundant bits does not exceed $3t\log_2 n + 4t + \frac{1}{2}\log_2(n - 4t - 2) - \log_2(t+1) + 6$. □

Remark 3: It is possible to correct a combination of symmetric and asymmetric multiset deletions by suitably combining the constructions $\mathcal{S}_{DA}^{(t)}(n)$ and $\mathcal{S}_{DS}^{(t)}(n)$. More specifically, one can correct t_1 asymmetric and t_2 symmetric multiset deletions with the code

$$\begin{aligned}
\mathcal{S}_{DAS}^{(t_1,t_2)}(n) =& \mathcal{S}_{DA}^{(t_1+t_2)}(n) \cap \mathcal{S}_{DS}^{(t_2)}(n)\\
=& \{\boldsymbol{s} \in \{0,1\}^n : s_1 = 0, s_n = 1,\\
& \exists\, \mathcal{I} \subset \left\{2, \ldots, \frac{n}{2} - t_1 - t_2 - 1\right\},\ |\mathcal{I}| \geq t_1 + t_2,\\
& \text{where } \forall i \in \mathcal{I}, s_i \neq s_{n-i+1}, \text{and } \forall i \notin \mathcal{I}, s_i = s_{n-i+1},\\
& \boldsymbol{s}_{[\frac{n}{2}]\cap\mathcal{I}} \text{is a string where each prefix has at least}\\
& t_1 + t_2 \text{ more 0s than 1s},\\
& \boldsymbol{\sigma}_s \in 3t_2\text{-erasure-correcting code},\\
& \forall i \in \left\{\frac{n}{2} - t_1 - t_2, \ldots, \frac{n}{2}\right\}, (s_i, s_{n-i+1}) \neq (1,0)\}.
\end{aligned}$$

The proof will follow similarly to that of Theorem 2 and Theorem 4. Intuitively, the aforementioned code can correct a combination t_1 asymmetric and t_2 symmetric multiset deletions since the constraint of $\boldsymbol{\sigma}$ belonging to a $3t_2$-erasure correcting code allows the complete recovery of $\boldsymbol{\sigma}$, and thereby effectively transforms the reconstruction problem into that of correcting $t_1 + t_2$ asymmetric multiset deletions, which can be corrected due to the constraints of $\mathcal{S}_{DA}^{(t_1+t_2)}(n)$. The redundancy of this construction will be at most $3t_2\log_2 n + \frac{1}{2}\log_2(n - 2t_1 - 4t_2 - 2) + 2t_1 + 4t_2 - \log_2(t_2+1) + \mathcal{O}(1)$ bits.

However, $\mathcal{S}_{DAS}^{(t_1,t_2)}(n)$ does not necessarily correct any $(t_1 + t_2)$ multiset deletions.

VII. Skewed Substitution-Correcting Codes

In this section, we confine our focus to the correction of skewed substitution errors [see Definition 8].

Lemma 8: Consider any $\boldsymbol{s} \in \mathcal{S}_R(n)$. Given that there occurs a single skewed substitution error in its composition set, one can uniquely recover $\boldsymbol{s}$.

Proof: In the following, we let the corrupted composition set be denoted by $C'(\boldsymbol{s}) = \bigcup_{i\in[n]} C'_i(\boldsymbol{s})$.

Case 1: n is even.

Given $C'(\boldsymbol{s})$, it is easy to identify the corrupted composition multiset $C'_k(\boldsymbol{s})$, since the following relation only holds for k:

$$w'_k < w'_{n-k+1}. \tag{50}$$

If we now delete all elements of $C'_k(\boldsymbol{s})$ from $C'(\boldsymbol{s})$, Lemma 3 tells us that $\boldsymbol{s}$ is still uniquely recoverable.

Case 2: n is odd.

Using the arguments of the preceding case, we can reach the same conclusion for an odd n, when the affected multiset is $C'_k(\boldsymbol{s})$, where $\lceil n/2 \rceil < k \leq n$, because in these cases, there exists an uncorrupted distinct symmetric multiset $C'_{n-k+1}(\boldsymbol{s})$, which gives us the true cumulative weight and thus allows us to accurately recover $\boldsymbol{\sigma}_s$.

If $k = \lceil n/2 \rceil$, this is no longer true since the multiset $C_{\lceil n/2 \rceil}(\boldsymbol{s})$ is its own symmetric counterpart. Noting that this normally helps us determine the bits $(s_{\lceil n/2 \rceil - 1}, s_{\lceil n/2 \rceil + 1})$, we recall from Lemma 2 that when these bits are assigned incorrectly, inconsistencies with the multiset $C_{\lceil n/2 \rceil - 1}$ would arise, which are not permitted under the considered error model. Hence, we conclude that $\boldsymbol{s}$ can be recovered uniquely.

□

We now consider a more general error model involving multiple asymmetric skewed substitution errors, wherein each multiset pair $\widetilde{C}_i$, for any $i \in [n]$, may contain at most one skewed substitution and the total number of errors does not exceed t. It is found that any asymmetric t-multiset deletion code is also robust to t asymmetric skewed substitutions. It is worth observing that in general, it cannot be said that a t-(asymmetric) multiset deletion code is also a t-(asymmetric) composition substitution-correcting code.

Theorem 6: A t-asymmetric multiset deletion code is a t-asymmetric skewed composition code.

Proof: We aim to show that if t skewed asymmetric substitution errors occur in $C(\boldsymbol{s})$, where $\boldsymbol{s} \in \mathcal{S}_{DA}^{(t)}(n)$, such that for all $1 \leq i \leq n$, $\widetilde{C}_i(\boldsymbol{s})$ contains at most one skewed substitution error, then one can uniquely recover $\boldsymbol{s}$.

Since the error model only allows at most one skewed substitution in a pair of symmetric multisets, the cumulative weights of all sets can be determined accurately. This is due to the fact that if multiset $C_k(\boldsymbol{s})$ has been corrupted, we may write:

$$w_k < w_{n-k+1}. \tag{51}$$

As a consequence, all cumulative weights can be correctly re-assigned and in turn the $\boldsymbol{\sigma}_s$ sequence can be recovered. The preceding inequality also allows us to identify the affected multisets, the deletion of which would transform our problem of correcting t asymmetric skewed substitutions into reconstruction under the absence of t multisets. According to Theorem 2, unique reconstruction of $\boldsymbol{s}$ is perfectly possible, thus concluding our proof. □

Consequently, $\mathcal{S}_{DA}^{(t)}(n)$ is a t-asymmetric skewed composition code.

VIII. Conclusion

In this work, we propose and investigate error models involving insertion and deletion of substring compositions in the context of polymer-based data storage. In particular, we examine the robustness of the composition-reconstructable code introduced in [11] and [12], and identify the situations which do not guarantee unique reconstruction of codewords from this construction. For these cases, new codes are proposed. Notably, an equivalence between codes correcting multiset deletions and insertions is established. We also examine a special asymmetric variant of substitution errors, namely skewed substitution errors, which manifest in polymer-based storage.

Several problems pertaining to string reconstruction under this data storage paradigm still remain open:

- The error model involving skewed substitutions under a symmetric setting is yet to be investigated. It would be interesting to know if there exists a suitable codebook offering a lower redundancy than that designed to correct standard substitution errors under the symmetric setting, as stated in [11].
- The extension of the problem of string reconstruction from composition multisets, error-free or otherwise, to larger alphabets, is also a promising direction.
- Though some bounds on the maximum number of mutually equicomposable strings were stated in [10], it is still not known if the existing code constructions are optimal.
- This research may be extended to a setting wherein bits are arranged in a circular fashion, on a ring.
- As pointed out in [10], a polynomial-time algorithm for the string reconstruction problem is yet to be found.
- Lemma 3 establishes that $S_R(n)$ can correct a single multiset deletion, while Theorem 3 implies that $S_R(n)$ can also correct a single composition substitution. However, more generally, it seems unclear if any t-multiset deletion code is also a t-composition substitution code. A counter-example could not be found. Hence, finding a proof for the same is certainly an open problem. It is however easy to show that any $2t$-multiset deletion code is also a t-composition substitution code.

Appendix

Lemma 9: Consider a string $\boldsymbol{s} \in \mathcal{S}_R(n)$. Given $C'(\boldsymbol{s}) = \bigcup_{i\in[n]\setminus\{k-1,k,n-k+1,n-k+2\}} C_i(\boldsymbol{s})$ for any $1 \le k < \lceil \frac{n-1}{2} \rceil$, $\boldsymbol{s}$ may no longer be uniquely determined.

Proof:

Case 1: n is even and deleted sets are: $\{C_{\frac{n}{2}-1}(\boldsymbol{s}), \dots, C_{\frac{n}{2}+2}(\boldsymbol{s})\}$.

To demonstrate that $\mathcal{S}_R(n)$ does not necessarily preserve unique reconstructability when the multisets $\{C_{\frac{n}{2}-1}, \dots, C_{\frac{n}{2}+2}\}$ go missing, we consider two codewords $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_R(n)$, such that:

$$\bigcup_{i\in\{n,\dots,\frac{n}{2}+3\}} C_i(\boldsymbol{s}) = \bigcup_{i\in\{n,\dots,\frac{n}{2}+3\}} C_i(\boldsymbol{v}). \tag{52}$$

$\boldsymbol{s}_1^{\frac{n}{2}-3}$	0	0	1	0	0	0	$\boldsymbol{s}_{\frac{n}{2}+4}^{n}$
$\boldsymbol{v}_1^{\frac{n}{2}-3}$	1	0	0	0	0	0	$\boldsymbol{v}_{\frac{n}{2}+4}^{n}$

Fig. 8. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are specified by (53) and (54).

From our knowledge of the reconstruction algorithm [Section II], we can also infer the following:

$$\begin{aligned}(\boldsymbol{s}_1^{n/2-3}, \boldsymbol{s}_{n/2+4}^{n}) &= (\boldsymbol{v}_1^{n/2-3}, \boldsymbol{v}_{n/2+4}^{n}),\\ \sigma_i &= \sigma'_i. \qquad 1 \le i \le \frac{n}{2}-3,\\ \sigma_{\frac{n}{2}-2} + \sigma_{\frac{n}{2}-1} + \sigma_{\frac{n}{2}} &= \sigma'_{\frac{n}{2}-2} + \sigma'_{\frac{n}{2}-1} + \sigma'_{\frac{n}{2}}.\end{aligned} \tag{53}$$

where $\boldsymbol{\sigma_s} = (\sigma_1, \dots, \sigma_{n/2})$ and $\boldsymbol{\sigma_v} = (\sigma'_1, \dots, \sigma'_{n/2})$ correspond to $\boldsymbol{s}$ and $\boldsymbol{v}$ respectively. Additionally, we set:

$$\begin{aligned}(\sigma_{\frac{n}{2}-2}, \sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) &= (0,0,1),\\ (\sigma'_{\frac{n}{2}-2}, \sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) &= (1,0,0),\\ v_{n/2-2} &= 1,\\ s_{n/2} &= 1,\\ s_{n-3} &= 0,\\ \mathrm{wt}(\boldsymbol{s}_2^{n/2-3}) &= \mathrm{wt}(\boldsymbol{s}_{n/2+4}^{n-4}).\end{aligned} \tag{54}$$

The relations between $\boldsymbol{s}$ and $\boldsymbol{v}$ as described by (53) and (54) are depicted in Fig. 8. Evidently, $\boldsymbol{s}$ and $\boldsymbol{v}$ differ in their respective multisets $C_{n/2+2}$ and $C_{n/2+1}$ according to Lemma 2. Additionally, since their cumulative weights $w_{n/2+2}$ and $w_{n/2}$ also differ, as one may verify from (6) and (54), we deduce that the multisets $C_{n/2}$ and $C_{n/2-1}$ also do not match for $\boldsymbol{s}$ and $\boldsymbol{v}$. We now proceed to examine if $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ holds:

$$\left\{\begin{matrix} \{c(\boldsymbol{s}_1^{\frac{n}{2}-3}), 0\} \\ \{c(\boldsymbol{s}_2^{\frac{n}{2}-3}), 0^2\} \\ \{c(\boldsymbol{s}_3^{\frac{n}{2}-3}), 0^2 1\} \\ \{c(\boldsymbol{s}_4^{\frac{n}{2}-3}), 0^3 1\} \\ \{c(\boldsymbol{s}_5^{\frac{n}{2}-3}), 0^4 1\} \\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n}), 0\} \\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-1}), 0^2\} \\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-2}), 0^3\} \\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-3}), 0^3 1\} \\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-4}), 0^4 1\} \end{matrix}\right\} = \left\{\begin{matrix} \{c(\boldsymbol{v}_1^{\frac{n}{2}-3}), 1\} \\ \{c(\boldsymbol{v}_2^{\frac{n}{2}-3}), 01\} \\ \{c(\boldsymbol{v}_3^{\frac{n}{2}-3}), 0^2 1\} \\ \{c(\boldsymbol{v}_4^{\frac{n}{2}-3}), 0^3 1\} \\ \{c(\boldsymbol{v}_5^{\frac{n}{2}-3}), 0^4 1\} \\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n}), 0\} \\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-1}), 0^2\} \\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-2}), 0^3\} \\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-3}), 0^4\} \\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-4}), 0^5\} \end{matrix}\right\}. \tag{55}$$

Using (54) to simplify this set equality relation, we arrive at:

$$\left\{\begin{matrix} \{c(\boldsymbol{s}_1^{\frac{n}{2}-3}), 0\} \\ \{c(\boldsymbol{s}_2^{\frac{n}{2}-3}), 0^2\} \\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-3}), 0^3 1\} \\ \{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-4}), 0^4 1\} \end{matrix}\right\} = \left\{\begin{matrix} \{c(\boldsymbol{v}_1^{\frac{n}{2}-3}), 1\} \\ \{c(\boldsymbol{v}_2^{\frac{n}{2}-3}), 01\} \\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-3}), 0^4\} \\ \{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-4}), 0^5\} \end{matrix}\right\}. \tag{56}$$

Since the construction of $\mathcal{S}_R(n)$ requires $s_1 = 0$ and (54) mandates that $s_{n-3} = 0$ and $\mathrm{wt}(\boldsymbol{s}_2^{n/2-3}) = \mathrm{wt}(\boldsymbol{s}_{n/2+4}^{n-4})$, we are led to the following relation:

$$\mathrm{wt}(\boldsymbol{s}_1^{n/2-3}) = \mathrm{wt}(\boldsymbol{s}_2^{n/2-3}) = \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}+4}^{n-3}) = \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}+4}^{n-4}). \quad (57)$$

This allows us to conclude that (55) indeed holds, and further bit specifications in $\boldsymbol{s}$ and $\boldsymbol{v}$ can lead us to similar set equality relations for the multisets $C_{n/2-3}, \ldots, C_1$. Hence, $\boldsymbol{s}$ and $\boldsymbol{v}$ become confusable under the deletion of multisets $\{C_{\frac{n}{2}-1}(\boldsymbol{s}), \ldots, C_{\frac{n}{2}+2}(\boldsymbol{s})\}$.

Case 2: n may be odd/even and the four deleted sets are not consecutive: $\{C_{k-1}(\boldsymbol{s}), C_k(\boldsymbol{s}), C_{n-k+1}(\boldsymbol{s}), C_{n-k+2}(\boldsymbol{s})\}$, where $k+1 < n-k+1$.

In the following, we once again proceed by checking if $\boldsymbol{s}$ is uniquely recoverable, by probing the existence of some $\boldsymbol{v} \in \mathcal{S}_R(n)$, characterized by $\sigma'_1, \ldots, \sigma'_{\lceil \frac{n}{2} \rceil}$ such that for all $i \in [n] \backslash \{k-1, k-n-k+1, n-k+2\}$:

$$C_i(\boldsymbol{s}) = C_i(\boldsymbol{v}). \quad (58)$$

Subcase (i): $k = 2$

This situation corresponds to the deletion of multisets $C_1(\boldsymbol{s})$, $C_2(\boldsymbol{s})$, $C_{n-1}(\boldsymbol{s})$ and $C_n(\boldsymbol{s})$. When this happens, for any $3 \le i \le \lceil n/2 \rceil - 1$, the following values are recoverable:

$$w_{i+1}(\boldsymbol{s}) - w_i(\boldsymbol{s}) = \sigma_{i+1} + \ldots + \sigma_{\lceil n/2 \rceil}.$$

This can be used to recover the values of $\sigma_4, \ldots, \sigma_{\lceil n/2 \rceil}$. In other words,

$$\sigma_i = \sigma'_i. \quad \forall \;\; 4 \le i \le \lceil n/2 \rceil \quad (59)$$

Furthermore, since $w_3(\boldsymbol{s}) = w_3(\boldsymbol{v})$, we can infer from (5) and (59) that:

$$\begin{aligned} \sigma_1 + 2\sigma_2 + 3\sigma_3 &= \sigma'_1 + 2\sigma'_2 + 3\sigma'_3 \\ \Longrightarrow \; 2\sigma_2 + 3\sigma_3 &= 2\sigma'_2 + 3\sigma'_3. \end{aligned}$$

The second equality follows from the construction of $\mathcal{S}_R(n)$. Given the above relation, we conclude that (59) also holds for $i \in \{2, 3\}$. Moreover, we cannot have $(s_2, s_{n-1}) \neq (v_2, v_{n-1})$ even when $\sigma_2 = \sigma'_2 = 1$, since the Catalan-Bertrand structure would automatically imply that $(s_2, s_{n-1}) = (v_2, v_{n-1}) = (0, 1)$. This inference, combined with Lemma 2, lead us to the conclusion that no suitable $\boldsymbol{v}$ exists.

Subcase (ii): $k = 3$

When multisets $C_2(\boldsymbol{s}), C_3(\boldsymbol{s}), C_{n-2}(\boldsymbol{s})$ and $C_{n-1}(\boldsymbol{s})$ have been deleted, the availability of cumulative weights $w_1, w_4, \ldots, w_{\lceil n/2 \rceil}$ allow us to retrieve $\sigma_1, \sigma_5, \ldots, \sigma_{\lceil n/2 \rceil}$ as in the previous subcase, i.e.

$$\sigma_i = \sigma'_i. \quad \forall \;\; i \in [\lceil n/2 \rceil] \backslash \{2, 3, 4\} \quad (60)$$

We also observe from (6) and (58) that:

$$\begin{aligned} w_4(\boldsymbol{s}) - w_1(\boldsymbol{s}) &= w_4(\boldsymbol{v}) - w_1(\boldsymbol{v}) \\ &= 3w_1(\boldsymbol{s}) - \sigma_3 - 2\sigma_2 - 3\sigma_1, \\ \Longrightarrow \; \sigma_2 + 2\sigma_3 &= \sigma'_2 + 2\sigma'_3. \end{aligned} \quad (61)$$

Similarly, since $w_5(\boldsymbol{s}) = w_5(\boldsymbol{v})$, we obtain:

$$\sigma_2 + 2\sigma_3 + 3\sigma_4 = \sigma'_2 + 2\sigma'_3 + 3\sigma_4.$$

As a consequence, (60) also holds for $i = 4$. This, along with (4) hints that:

$$\sigma_2 + \sigma_3 = \sigma'_2 + \sigma'_3. \quad (62)$$

$\boldsymbol{s}_1^{k-3}$	0	0	0	1	$\boldsymbol{s}_{k+2}^{n-k-1}$	0	1	1	0	$\boldsymbol{s}_{n-k+4}^{n}$
$\boldsymbol{v}_1^{k-3}$	0	0	0	1	$\boldsymbol{v}_{k+2}^{n-k-1}$	1	0	0	1	$\boldsymbol{v}_{n-k+4}^{n}$

Fig. 9. Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are related such that $(\boldsymbol{s}_1^{k-3}, \boldsymbol{s}_{n-k+4}^{n}) = (\boldsymbol{v}_1^{k-3}, \boldsymbol{v}_{n-k+4}^{n})$ and $c(\boldsymbol{s}_{k+2}^{n-k-1}) = c(\boldsymbol{v}_{k+2}^{n-k-1})$.

Equations (61) and (62) together insinuate that $(\sigma_2, \sigma_3) = (\sigma'_2, \sigma'_3)$. Hence, we may argue as before, that no suitable $\boldsymbol{v}$ distinct from $\boldsymbol{s}$ actually exists.

Subcase (iii): $k \ge 4$

Similar to the approach used in Case 1, we attempt to show that there exist two codewords $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_R(n)$, such that for all $i \in [n] \backslash \{k-1, k, n-k+1, n-k+2\}$:

$$C_i(\boldsymbol{s}) = C_i(\boldsymbol{v}). \quad (63)$$

To this end, we construct a specific pair of strings $\boldsymbol{s}$ and $\boldsymbol{v}$ as follows:

$$\begin{aligned} (\boldsymbol{s}_1^{k-3}, \boldsymbol{s}_{n-k+4}^{n}) &= (\boldsymbol{v}_1^{k-3}, \boldsymbol{v}_{n-k+4}^{n}), \\ (\sigma_{k-2}, \sigma_{k-1}, \sigma_k, \sigma_{k+1}) &= (1, 1, 1, 0), \\ (\sigma'_{k-2}, \sigma'_{k-1}, \sigma'_k, \sigma'_{k+1}) &= (2, 0, 0, 1), \\ \sigma_i &= \sigma'_i, \qquad \forall \; k+2 \le i \le \lceil \frac{n}{2} \rceil \\ (s_{k-1}, s_k, s_{k+1}, s_{k+2}) &= (0, 0, 1), \\ s_2 &= 1, \\ v_{k-2} &= 0. \end{aligned} \quad (64)$$

These relations have been illustrated in Fig. 9. The preceding equalities also imply that:

$$\begin{aligned} \sigma_i &= \sigma'_i, \qquad \forall \; 1 \le i \le k-3 \\ \sum_{i=k-2}^{k+1} \sigma_i &= \sum_{i=k-2}^{k+1} \sigma'_i, \\ \sigma_k + 2\sigma_{k-1} + 3\sigma_{k-2} &= \sigma'_k + 2\sigma'_{k-1} + 3\sigma'_{k-2}, \\ c(\boldsymbol{s}_{k+2}^{n-k-1}) &= c(\boldsymbol{v}_{k+2}^{n-k-1}). \end{aligned}$$

In turn, these relations help ensure that:

$$\begin{aligned} w_i(\boldsymbol{s}) &= w_i(\boldsymbol{v}), \qquad \forall \; 1 \le i \le k-2 \\ w_{k+1}(\boldsymbol{s}) - w_{k-2}(\boldsymbol{s}) &= w_{k+1}(\boldsymbol{v}) - w_{k-2}(\boldsymbol{v}), \\ w_{k+i+1}(\boldsymbol{s}) - w_{k+i}(\boldsymbol{s}) &= w_{k+i+1}(\boldsymbol{v}) - w_{k+i}(\boldsymbol{v}). \end{aligned}$$

for $1 \le i \le n-k-1$. One may verify this with the assistance of (4) and (6).

From Fig. 9, it is fairly evident that $\boldsymbol{s}$ and $\boldsymbol{v}$ do not match in their corresponding multisets C_{n-k+2} and C_{n-k+1}. Now, as done in Case 1, we check if multisets $C_{n-k}(\boldsymbol{s})$ and $C_{n-k}(\boldsymbol{v})$ match:

$$\left\{\begin{matrix} \{c(\boldsymbol{s}_1^{k-3}), 0^4 1, c\} \\ \{c(\boldsymbol{s}_2^{k-3}), 0^4 1^2, c\} \\ \{c(\boldsymbol{s}_3^{k-3}), 0^4 1^3, c\} \\ \{c(\boldsymbol{s}_{n-k+4}^{n}), 0^2 1^3, c\} \\ \{c(\boldsymbol{s}_{n-k+4}^{n-1}), 0^3 1^3, c\} \\ \{c(\boldsymbol{s}_{n-k+4}^{n-2}), 0^4 1^3, c\} \end{matrix}\right\} = \left\{\begin{matrix} \{c(\boldsymbol{v}_1^{k-3}), 0^3 1^2, c\} \\ \{c(\boldsymbol{v}_2^{k-3}), 0^4 1^2, c\} \\ \{c(\boldsymbol{v}_3^{k-3}), 0^5 1^2, c\} \\ \{c(\boldsymbol{v}_{n-k+4}^{n}), 0^2 1^3, c\} \\ \{c(\boldsymbol{v}_{n-k+4}^{n-1}), 0^3 1^3, c\} \\ \{c(\boldsymbol{v}_{n-k+4}^{n-2}), 0^4 1^3, c\} \end{matrix}\right\}.$$

where $c = c(\boldsymbol{s}_{k+2}^{n-k-1}) = c(\boldsymbol{v}_{k+2}^{n-k-1})$. By applying (64) to this, we deduce that this equality is indeed upheld, thus implying that $\boldsymbol{s}$ and $\boldsymbol{v}$ are confusable under the absence of multisets $C_{k-1}, C_k, C_{n-k+1}, C_{n-k+2}$.

□

Acknowledgment

The authors would like to thank the reviewers and the editors for their insightful comments and helpful suggestions.

References

[1] A. Banerjee, A. Wachter-Zeh, and E. Yaakobi, "Insertion and deletion correction in polymer-based data storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 802–807.

[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 Commun.*, vol. 8, no. 1, p. 967, Oct. 2017.

[3] N. Goldman et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, pp. 77–80, Jan. 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," *Angew. Chem. Int. Ed.*, vol. 54, no. 8, pp. 2552–2555, 2015, doi: 10.1002/anie.201411378.

[5] R. Heckel, G. Mikutis, and R. N. Grass, "A characterization of the DNA data storage channel," *Sci. Rep.*, vol. 9, no. 1, p. 9663, Jul. 2019.

[6] C. N. Takahashi, B. H. Nguyen, K. Strauss, and L. Ceze, "Demonstration of end-to-end automation of DNA data storage," *Sci. Rep.*, vol. 9, no. 1, p. 4998, Mar. 2019.

[7] S. M. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," *Sci. Rep.*, vol. 5, no. 1, p. 14138, Sep. 2015.

[8] S. M. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Rep.*, vol. 7, p. 5011, Jul. 2017.

[9] S. K. Tabatabaei et al., "DNA punch cards for storing data on native DNA sequences via enzymatic nicking," *Nature Commun.*, vol. 11, no. 1, p. 1742, Apr. 2020.

[10] J. Acharya, H. Das, O. Milenkovic, A. Orlitsky, and S. Pan, "String reconstruction from substring compositions," *SIAM J. Discrete Math.*, vol. 29, no. 3, pp. 1340–1371, 2015, doi: 10.1137/140962486.

[11] S. Pattabiraman, R. Gabrys, and O. Milenkovic, "Coding for polymer-based data storage," 2020, *arXiv:2003.02121*.

[12] S. Pattabiraman, R. Gabrys, and O. Milenkovic, "Reconstruction and error-correction codes for polymer-based data storage," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Aug. 2019, pp. 1–5.

[13] R. Gabrys, S. Pattabiraman, and O. Milenkovic, "Mass error-correction codes for polymer-based data storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 25–30.

[14] R. Gabrys, S. Pattabiraman, and O. Milenkovic, "Reconstructing mixtures of coded strings from prefix and suffix compositions," 2020, *arXiv:2010.11116*.

Anisha Banerjee (Student Member, IEEE) received the B.E. degree in electrical engineering from Jadavpur University, India, in 2018, and the M.Sc. degree (Hons.) in communications engineering from the Technical University of Munich (TUM), Munich, Germany, in 2021, where she is currently pursuing the Ph.D. degree with the Coding and Cryptography Group, Institute of Communications Engineering, under the supervision of Prof. Wachter-Zeh. Her research interests include coding theory and information theory and their applications to storage, with a special focus on insertion and deletion errors.

Antonia Wachter-Zeh (Senior Member, IEEE) received the M.Sc. degree in communications technology from Ulm University, Germany, in 2009, and the joint Ph.D. degree from Ulm University and Universite de Rennes 1, Rennes, France, in 2013. From 2013 to 2016, she was a Post-Doctoral Researcher with the Technion—Israel Institute of Technology, Haifa, Israel, and from 2016 to 2020, a tenure Track Assistant Professor with the Technical University of Munich (TUM), Munich, Germany, where she is currently an Associate Professor with the School of Computation, Information and Technology. Her research interests are coding theory, cryptography and information theory and their application to storage, communications, privacy, security, and machine learning. She was a recipient of the DFG Heinz Maier-Leibnitz-Preis and of an ERC Starting Grant. She is currently an Associate Editor of the IEEE Transactions on Information Theory.

Eitan Yaakobi (Senior Member, IEEE) received the B.A. degree in computer science and mathematics and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California at San Diego, San Diego, in 2011. He is currently an Associate Professor with the Computer Science Department, Technion—Israel Institute of Technology. Between 2011 and 2013, he was a Post-Doctoral Researcher with the Department of Electrical Engineering, California Institute of Technology, and the Center for Memory and Recording Research, University of California at San Diego, San Diego. His research interests include information and coding theory with applications to non-volatile memories, associative memories, DNA storage, data storage and retrieval, and private information retrieval. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010 and 2011. Since 2020, he serves as an Associate Editor for Coding and Decoding for the IEEE Transactions on Information Theory.