Bull. Aust. Math. Soc. **114** (2026), 197–209
doi:10.1017/S000497272610104X

EXPECTED LENGTH OF THE LONGEST COMMON SUBSEQUENCE OF MULTIPLE STRINGS

RAY LI[✉], **WILLIAM REN** and **YIRAN WEN**

(Received 14 October 2025; accepted 3 January 2026; first published online 2 March 2026)

Abstract

We study the generalised Chvátal–Sankoff constant $\gamma_{k,d}$, which represents the normalised expected length of the longest common subsequence of d independent uniformly random strings over an alphabet of size k. We derive asymptotically tight bounds for $\gamma_{2,d}$, establishing that $\gamma_{2,d} = \frac{1}{2} + \Theta(1/\sqrt{d})$. We also derive asymptotically near-optimal bounds on $\gamma_{k,d}$ for $d \geq \Omega(\log k)$.

2020 *Mathematics subject classification*: primary 68R15.

Keywords and phrases: longest common subsequence, Chvátal–Sankoff constant, expected value, multiple strings, greedy algorithm.

1. Introduction

The longest common subsequence (LCS) is a fundamental measure of the similarity of two or more strings that is important in theory and practice. A *subsequence* of a string is obtained by removing zero or more characters, and the LCS of d strings $X^1, \ldots, X^d$ is the longest subsequence that occurs in all of $X^1, \ldots, X^d$. For d strings $X^1, \ldots, X^d$, we let $\mathrm{LCS}(X^1, \ldots, X^d)$ denote the length of their LCS. For example, $\mathrm{LCS}(0011, 0101) = 3$. Computing the LCS is a textbook application of dynamic programming in computer science [24], and the algorithm has many applications from text processing, to linguistics and computational biology. As one example, the Linux `diff` tool uses a variation of the LCS algorithm.

Chvátal and Sankoff [5] showed that as n approaches infinity, the normalised expected length of the LCS of two independent uniformly random binary strings converges to a constant. This limit is known as the Chvátal–Sankoff constant,

$$\gamma \stackrel{\text{def}}{=} \lim_{n\to\infty} \frac{\mathbf{E}_{X^1,X^2\sim\{0,1\}^n}[\mathrm{LCS}(X^1, X^2)]}{n},$$

RL, WR and YW are supported by NSF grant CCF-2347371.

© The Author(s), 2026. Published by Cambridge University Press on behalf of Australian Mathematical Publishing Association Inc. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution, and reproduction in any medium, provided the original work is properly cited.

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

TABLE 1. History of bounds and estimates for the Chvátal–Sankoff constant, γ.

Authors	Year	Proven bounds	Estimates
Chvátal and Sankoff [5]	1975	$0.697\,844 \le \gamma \le 0.866\,595$	$\gamma \approx 0.8082$
Deken [9]	1979	$0.7615 \le \gamma \le 0.8575$	
Steele [23]	1986		$\gamma \approx 0.8284$
Dančík and Paterson [8]	1995	$0.773\,91 \le \gamma \le 0.837\,63$	
Boutet de Monvel [2]	1999		$\gamma \approx 0.812\,282$
Baeza-Yates *et al.* [1]	1999		$\gamma \approx 0.8118$
Bundschuh [4]	2001		$\gamma \approx 0.812\,653$
Lueker [19]	2009	$0.788\,071 \le \gamma \le 0.826\,28$	
Bukh and Cox [3]	2022		$\gamma \approx 0.8122$
Soiffer *et al.* [22]	2024	$0.792\,665 \le \gamma$	

where the expectation is over independent uniformly random binary strings X^1, X^2. Determining γ is an open question with a rich history [1–5, 9, 18–20, 22, 23]. Currently the best bounds are roughly $0.792\,665 \le \gamma \le 0.826\,280$ [19, 22].

Table 1 presents a summary of key works that have contributed to establishing bounds on the Chvátal–Sankoff constant. Only some studies offer rigorously proven bounds, while others present estimates.

There are two natural ways to generalise the Chvátal–Sankoff problem: (1) increase the alphabet size; and (2) increase the number of strings. In this way, we may generalise the Chvátal–Sankoff constant by asking for $\gamma_{k,d}$, the (normalised) expected longest common subsequence of d independent uniformly random strings over a size-k alphabet. Formally, let

$$\gamma_{k,d} = \lim_{n\to\infty} \frac{\mathbf{E}_{X^1,\dots,X^d\sim[k]^n}[\mathrm{LCS}(X^1,\dots,X^d)]}{n},$$

where the expectation is over independent uniformly random strings $X^1, \dots X^d \sim [k]^n$ and $[k] = \{1, \dots, k\}$. By definition, $\gamma = \gamma_{2,2}$.

The generalisation to larger alphabet size k is well studied and well understood. This line of work in [1, 6, 9, 20] culminated in a beautiful result that $\gamma_{k,2} \to 2/\sqrt{k}$ as $k \to \infty$ [16], answering a conjecture of Sankoff and Mainville [21].

We study the generalisation to more strings d, which is also an important question. Mathematically it is a fundamental generalisation of the Chvátal–Sankoff constant. In computer science, it is intimately connected to error-correcting codes list-decodable against deletions [15] (see also [10, 11, 13]). Specifically, $1 - \gamma_{k,d}$ is the maximum fraction of deletions that a positive-rate random code can list-decode against with list size $d - 1$. This connection follows from a generalisation of a martingale concentration argument shown in [15]. For completeness, we show the connection in Appendix B.

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

Several works have previously considered generalising the number of strings d, but less is known than for the larger-alphabet generalisation. Jiang and Li [14] showed that when $d = n$, the expected LCS of d strings is roughly n/k. Dančík [7] showed that, for fixed d, $\gamma_{k,d} = c/k^{1-1/d}$ for some constant $c \in [1, e]$, disproving a conjecture of Steele [23] that $\gamma_{k,d} = \gamma_{k,2}^{d-1}$. Kiwi and Soto [17] established numerical bounds on $\gamma_{k,d}$ for small values of k and d. For example, they obtain bounds on $\gamma_{k,d}$ up to $d = 14$ for a binary alphabet, and up to alphabet size $k = 10$ for $d = 3$ strings. Recent work of Soiffer *et al.* [22] improves upon [17] and establishes stronger numerical bounds.

1.1. Our contributions. We give tight asymptotic bounds on the binary Chvátal–Sankoff constant as the number of strings increases, showing $\gamma_{2,d} = \frac{1}{2} + \Theta(1/\sqrt{d})$.

THEOREM 1.1. *There exists constants $0 < c_1 < c_2$ such that, for all integers $d \geq 2$,*

$$\frac{1}{2} + \frac{c_1}{\sqrt{d}} \leq \gamma_{2,d} \leq \frac{1}{2} + \frac{c_2}{\sqrt{d}}$$

Our main contribution is the lower bound, which combines a technique of Lueker [19] and Kiwi and Soto [17] with a greedy matching strategy. Our upper bound follows from a counting argument of Guruswami and Wang [13], who studied codes for list-decoding deletions.

We also give bounds that are asymptotically near-optimal for larger alphabets.

THEOREM 1.2. *There exist constants $c_0, c_1, c_2 > 0$ such that, for all integers d and k with $d \geq c_0 \log k$,*

$$\frac{1}{k}\left(1 + \frac{c_1}{\sqrt{d}}\right) \leq \gamma_{k,d} \leq \frac{1}{k}\left(1 + c_2\sqrt{\frac{\log k}{d}}\right).$$

The lower bound of Theorem 1.2 follows from Theorem 1.1 by noting that $\gamma_{k,d} \geq 2\gamma_{2,d}/k$: random k-ary strings of length n typically have binary subsequences of length roughly $2n/k$ (see Appendix A). The upper bound again follows from a counting argument of Gurusuwami and Wang [13].

1.2. Organisation of the paper. In Section 2, we illustrate the ideas in our proof by sketching the proof in the binary case, $k = 2$. In Section 3, we present preliminaries for the proofs. In Section 4, we prove Theorem 1.1.

2. Proof overview

We now sketch the proof of Theorem 1.1 in the binary case, $k = 2$. We start with the lower bound.

2.1. The Kiwi–Soto reduction to diagonal LCS. Our first step is to reduce the generalised Chvátal–Sankoff $\gamma_{k,d}$ problem to estimating the expected *diagonal LCS*. This approach was considered by Lueker [18], who focused on the two-string case ($d = 2$) and obtained numerical lower bounds. It was then generalised by Kiwi and

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

Soto [17] (see also [22]) to obtain numerical lower bounds for more strings $d \geq 3$. We use the same technique to find lower bounds for any number of strings d.

Let $A_1, \ldots, A_d$ be a collection of d finite binary strings. Let $X_1, \ldots, X_d$ be a collection of d independent uniformly random binary strings of length n. For a string X, let $X[1 \cdots i]$ denote the substring formed by the first i characters of string X and $A_jX_j[1 \cdots i]$ denote the concatenation of strings A_j and $X_j[1 \cdots i]$. Lueker (for $d = 2$) and Kiwi and Soto (for all d) define

$$W_n(A_1, \ldots, A_d) = \mathbb{E}_{X_1,\ldots,X_d}\Big[\max_{i_1+\cdots+i_d=n} \mathrm{LCS}(A_1X_1[1 \cdots i_1], \ldots, A_dX_d[1 \cdots i_d])\Big],$$

and show

$$\gamma_{2,d} = \lim_{n\to\infty} \frac{W_{nd}(A_1, \ldots, A_d)}{n}, \tag{2.1}$$

for all fixed strings $A_1, \ldots, A_d$. Leuker and Kiwi and Soto combine this result with a dynamic programming approach to find numerical lower bounds on $\lim_{n\to\infty} W_{nd}/n$, and thus $\gamma_{2,d}$ (and, more generally, $\gamma_{k,d}$).

We take $A_1, \ldots, A_d$ to be the empty string. Define the expected diagonal LCS as

$$W_n \stackrel{\text{def}}{=} \mathbb{E}\Big[\max_{i_1+\cdots+i_d=n} \mathrm{LCS}(X_1[1 \cdots i_1], \ldots, X_d[1 \cdots i_d])\Big] = W_n(\lambda, \ldots, \lambda), \tag{2.2}$$

where λ denotes the empty string. By (2.1), we have

$$\gamma_{2,d} = \lim_{n\to\infty} \frac{W_{nd}}{n}. \tag{2.3}$$

Intuitively, (2.3) is true because the maximum in (2.2) is obtained when $i_1, i_2, \ldots, i_d$ are all roughly n/d, so W_n approaches the expected LCS of d strings of length n/d.

2.2. Our greedy matching scheme and binary lower bound. We now find a lower bound for the relative diagonal LCS $\lim_{n\to\infty} W_{nd}/n$, and thus $\gamma_{2,d}$. To do this, we define a greedy matching strategy that finds a common subsequence of d random strings, one bit at a time. We track the number of bits we 'consume' across the d strings, per matched LCS bit. We show that our greedy matching consumes on average $2d - \Theta(\sqrt{d})$ bits per matched LCS bit, which on average gives us

$$\frac{nd}{2d - \Theta(\sqrt{d})} = n\left(\frac{1}{2} + \Theta\left(\frac{1}{\sqrt{d}}\right)\right)$$

LCS bits for nd symbols consumed. These estimates suggest $W_{nd} \geq n(\frac{1}{2} + \Theta(1/\sqrt{d}))$, and thus $\gamma_{2,d} \geq \frac{1}{2} + \Theta(1/\sqrt{d})$, and we then prove this estimate.

We now describe the matching strategy (illustrated in Figure 1). We match the LCS bit by bit, revealing the random bits as we need them; importantly, because the bits are independently random, we can reveal them in any desired order. For each LCS bit, we reveal the next bit in each of the d strings. We then take the next LCS bit to be the majority bit, say 0, and find the next 0 in each of the d strings. The number of bits consumed can be described by a process of repeatedly flipping d fair coins until all

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press



FIGURE 1. Our matching strategy for $d = 7$ random binary strings. Because all bits are independent, we can reveal the randomness in any order. We generate seven random bits. Suppose, as illustrated, four bits are a 1, and $Y = 3$ are a 0. We reveal more bits in the strings with 0s until we see 1s. Here, in total, to get the one LCS bit, we revealed the randomness from $Z = 13$ bits across the seven strings.

coins show the same face. We first flip all d coins. We keep reflipping all the coins in the minority until they show the majority face. For example, suppose we have flipped the d coins and heads appears $\lceil d/2 \rceil$ times. Then we repeatedly reflip the $\lfloor d/2 \rfloor$ coins that landed tails, until each shows heads. We let Z be the random variable denoting the total number of coin flips, or, equivalently, the total number of bits consumed per LCS bit.

To analyse the expected number of flips, we first consider the random variable Y, the number of coins in the minority after the first d flips. In the binary case, it is not hard to compute the expectation of Y explicitly. For example, when d is even,

$$\mathbb{E}[Y] = \frac{1}{2^d}\left(\sum_{i=0}^{d/2-1} \binom{d}{i} \cdot 2i + \binom{d}{d/2} \cdot \frac{d}{2}\right) = \frac{d}{2^d}\left(2^{d-1} - \binom{d-1}{d/2}\right) \approx \frac{d}{2} - \Theta(\sqrt{d}),$$

and a similar computation holds when d is odd. Intuitively, the estimate $\mathbf{E}[Y] = \frac{1}{2}d - \Theta(\sqrt{d})$ makes sense because $Y = \frac{1}{2}d - |\frac{1}{2}d - h|$, where h is the number of heads. The standard deviation of h is $\Theta(\sqrt{d})$, so we 'expect' $|\frac{1}{2}d - h|$ to be $\Theta(\sqrt{d})$, and thus Y to be $\frac{1}{2}d - \Theta(\sqrt{d})$.

Now that we have a handle on Y, we can study Z, the total number of bits consumed for one LCS bit. The number of reflips of each minority coin is a geometric random variable with $p = 1/2$. Thus, the expected number of reflips of each minority coin is 2. Taking into account the conditional expectations, we can show that the expected total number of reflips of minority coins is thus $2 \cdot \mathbf{E}[Y] = d - \Theta(\sqrt{d})$. Adding on the d initial flips, we have

$$\mathbf{E}[Z] = d + 2\,\mathbf{E}[Y] = 2d - \Theta(\sqrt{d}).$$

This shows (modulo some details) that our greedy matching strategy consumes $2d - \Theta(\sqrt{d})$ bits per matched bit. Our back-of-the-envelope calculation suggests that,

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

because we have nd bits to consume across the d strings, and we consume an average of $2d - \Theta(\sqrt{d})$ bits per matched bit, we expect to find a common subsequence of length at least $nd/(2d - \Theta(\sqrt{d})) = n(\frac{1}{2} + \Theta(1/\sqrt{d}))$, as desired.

However, we have to work harder to formally justify this. Let $Z_1, Z_2, \dots$ be the random variables where Z_i denotes the number of bits we need to consume to match the ith bit with our matching strategy. By carefully choosing the order in which we reveal our bits, we can arrange for $Z_1, Z_2, \dots$ to be mutually independent. Further, the Z_i are identically distributed as Z, and thus have expectation $2d - \Theta(\sqrt{d})$. The number of bits we matched by our strategy is the largest L such that $Z_1 + \cdots + Z_L \le nd$. Importantly, because we work with diagonal LCS, we do not need to worry that we use a different number of bits in different strings. To show the expected number of bits matched is close to our estimate, we show that

$$\Pr[Z_1 + \cdots + Z_{L_0} \le nd] > 1 - o(1) \quad \text{for } L_0 = \frac{nd}{\mathbf{E}[Z]}(1 - o(1)). \tag{2.4}$$

We cannot use a standard concentration inequality because the Z_i are unbounded. However, each Z_i is the sum of at most d geometric random variables. Thus, setting $Z_i' \stackrel{\text{def}}{=} \min(Z_i, O_d(\log n))$, with high probability, $Z_i' = Z_i$ for all i. We then use concentration inequalities to show $Z_1' + \cdots + Z_{L_0}' \le nd$ with high probability, and then (2.4) holds. Thus, the expected number of bits matched is at least $L_0 \cdot (1 - o(1)) \ge n(\frac{1}{2} + \Omega(1/\sqrt{d}))$. Hence, we arrive at our bound,

$$\gamma_{2,d} \ge \frac{1}{2} + \Omega\Big(\frac{1}{\sqrt{d}}\Big).$$

2.3. The binary upper bound. The upper bound follows from a counting argument. Guruswami and Wang [13, Lemma 2.3] (Lemma 4.1 below) bound the number of supersequences of any string of length $\ell > n/k$. By applying this bound and carefully tracking the lower-order terms, we show that $\Pr[\mathrm{LCS}(X_1, \dots, X_d) \ge \ell]$ is exponentially small for

$$\ell = \frac{n}{k}\Big(1 + \Theta\Big(\sqrt{\frac{\log k}{d}}\Big)\Big).$$

Our bound on the expectation follows.

3. Preliminaries

Throughout log is log base 2 unless otherwise specified, and ln is log base e. We use Wald's lemma and Hoeffding's inequality.

LEMMA 3.1 (Wald). *Let $Y, W_1, W_2, \dots$ be independent random variables supported on the nonnegative integers with finite expectations and such that $W_1, W_2, \dots$ are identically distributed. Define $W = W_1 + W_2 + \cdots + W_Y$. Then*

$$\mathbf{E}[W] = \mathbf{E}[Y]\,\mathbf{E}[W_1].$$

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

LEMMA 3.2 (Hoeffding). *Let $X_1, X_2, \dots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ almost surely. Then, for any $t > 0$,*

$$\mathbb{P}\Big(\sum_{i=1}^{n} X_i - \mathbb{E}\Big[\sum_{i=1}^{n} X_i\Big] \geq t\Big) \leq \exp\Big(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\Big).$$

For $p \in (0, 1)$, we define the q-ary entropy by

$$H_q(p) = p \log_q(q-1) - p \log_q(p) - (1-p)\log_q(1-p),$$

where $h(p)$ is the binary entropy function. We use a well-known estimate on binomial terms and the following estimate for k-ary entropy.

LEMMA 3.3 (see, for example, [12, Proposition 3.3.1]). *We have*

$$\binom{m}{pm}(q-1)^{pm} \leq q^{H_q(p)m}.$$

LEMMA 3.4 (see, for example, [12, Proposition 3.3.5]). *For small enough $\varepsilon \in (0, 1/k)$,*

$$H_k\Big(1 - \frac{1}{k} - \varepsilon\Big) \leq 1 - c_k \cdot \varepsilon^2 \quad \textit{for a constant } c_k \geq \frac{k^2}{4(k-1)\ln k} \geq \frac{k}{4\ln k}.$$

As described in Section 2, define the expected diagonal LCS

$$W_n \stackrel{\text{def}}{=} \mathbf{E}\Big[\max_{i_1+\cdots+i_d=n} \mathrm{LCS}(X^1[1\cdots i_1], \dots, X^d[1\cdots i_d])\Big],$$

where the randomness is over uniformly random infinite binary strings $X^1, \dots, X^d$. The following lemma shows that the diagonal LCS equals the expected LCS up to lower-order terms.

LEMMA 3.5 [17]. *We have $\gamma_{k,d} = \lim_{n\to\infty} W_{nd}/n$.*

4. Full proof of the k-ary LCS

4.1. Theorem 1.1, lower bound

PROOF OF THEOREM 1.1, LOWER BOUND. By Lemma 3.5, it suffices to show that there is an absolute constant $c_1 > 0$ such that, for sufficiently large n,

$$W_{nd} = \mathbf{E}\Big[\max_{i_1+\cdots+i_d=nd} \mathrm{LCS}(X^1[1\cdots i_1], \dots, X^d[1\cdots i_d])\Big] \geq n\Big(\frac{1}{2} + \frac{c_1}{\sqrt{d}}\Big).$$

We now present our greedy matching strategy for finding a long 'diagonal' common subsequence, that is, a common subsequence of $X^1[1\cdots i_1], \dots, X^d[1\cdots i_d]$ for $i_1 + \cdots + i_d = nd$. Given d random infinite strings $X^1, \dots, X^d$, we find the LCS bit by bit, revealing the random bits of $X^1, \dots, X^d$ as we need them. Importantly, because the bits are independently random, we can reveal them in any desired order. Use the following process.

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

(1) Initialize a string s to the empty string, representing our common subsequence of $X^1, \dots, X^d$.
(2) Repeat the following.
   (a) Reveal the next unrevealed bit $b_1, \dots, b_d$ in each of $X^1, \dots, X^d$.
   (b) Let b be the majority bit among these d bits.
   (c) For each string X^j that did not reveal the majority bit ($b_i \neq b$), reveal bits of X^j until we reveal a bit equal to b.
   (d) If the number of revealed bits is at most nd, append b to s, else exit.

See Figure 1 for an illustration of this process. The length of the subsequence we find is the number of times we successfully complete the loop.

Let Y denote the random variable that denotes the number of minority bits among d uniformly random bits. Let Z denote the random variable that first samples Y and is set to $d + W_1 + \cdots + W_Y$, where $W_1, \dots, W_Y$ are independent geometric random variables with probability 1/2. Because the bits are independent, the number of bits revealed in each iteration of the loop is distributed as Z. Thus, letting $Z_1, Z_2, \dots$ be independent random variables identically distributed as Z, the length of our LCS is distributed as

$$L_{\text{greedy}} \stackrel{\text{def}}{=} \max(L : Z_1 + \cdots + Z_L \leq nd).$$

We wish to find a lower bound for $\mathbf{E}[L_{\text{greedy}}]$.

We start by analysing the expectations of Y and Z. Explicit calculations yield, for all d,

$$\mathbf{E}[Y] \leq \frac{d}{2} - c\sqrt{d} \tag{4.1}$$

for some absolute constant $c > 0$. To see this, note that for d even,

$$\begin{aligned}\mathbb{E}[Y] &= \frac{1}{2^d}\Bigg(\sum_{i=0}^{d/2-1} \binom{d}{i} \cdot 2i + \binom{d}{d/2} \cdot \frac{d}{2}\Bigg) = \frac{1}{2^d}\Bigg(\sum_{i=0}^{d/2-1} 2d \cdot \binom{d-1}{i-1} + d \cdot \binom{d-1}{d/2-1}\Bigg) \\ &= \frac{d}{2^d}\Bigg(2^{d-1} - \binom{d-1}{d/2}\Bigg) \leq \frac{d}{2} - c\sqrt{d},\end{aligned}$$

and for d odd,

$$\begin{aligned}\mathbb{E}[Y] &= \frac{1}{2^d}\Bigg(\sum_{i=0}^{(d-1)/2} \binom{d}{i} \cdot 2i\Bigg) = \frac{1}{2^d}\Bigg(\sum_{i=0}^{(d-1)/2} 2d \cdot \binom{d-1}{i-1}\Bigg) \\ &= \frac{d}{2^d}\Bigg(2^{d-1} - \binom{d-1}{(d-1)/2}\Bigg) \leq \frac{d}{2} - c\sqrt{d},\end{aligned}$$

where $c > 0$ is some absolute constant. Thus, (4.1) holds. By Lemma 3.1 we have $\mathbf{E}[Z] \leq d + 2\,\mathbf{E}[Y] = d - 2c\sqrt{d}$.

Let $L_0 = nd(1-\gamma)/\mathbb{E}[Z]$ for $\gamma = 1/100 \log n$. We show that the sum $\sum_{i=1}^{L_0} Z_i$ is less than nd with very high probability, so that $L_{\text{greedy}} \geq L_0$ with very high probability. This follows from concentration inequalities, but we cannot apply the inequalities

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

directly because our random variables Z_i are unbounded. Define truncated variables $Z'_i = \min(Z_i, T)$ for $T = 100d \log n$, so that each Z'_i is in $[0, T]$.

We show that all $Z_i = Z'_i$ with high probability. In step 2(c), for each X^j, we see the correct bit b within $99 \log n$ steps with probability at least $1 - 1/n^{99}$. By the union bound, this happens for all $j = 1, \dots, d$ with probability at least $1 - d/n^{99}$, in which case $Z_i \le d + 99d \log n < T$ and $Z_i = Z'_i$. Thus, union-bounding over $i = 1, \dots, L_0$,

$$\Pr[Z_i = Z'_i \text{ for all } i = 1, \dots, L_0] \ge 1 - nd \cdot \frac{d}{n^{99}} \ge 1 - \frac{1}{n^{97}}. \tag{4.2}$$

Since $Z'_1, \dots, Z'_{L_0}$ are independent, Hoeffding's inequality (Lemma 3.2) implies

$$\mathbb{P}\Big[\sum_{i=1}^{L_0} Z'_i > \mathbb{E}\Big[\sum_{i=1}^{L_0} Z'_i\Big] + t\Big] \le \exp\Big(-\frac{2t^2}{\sum_{i=1}^{L_0} T^2}\Big),$$

where $t = nd - \mathbb{E}[\sum_{i=1}^{L_0} Z'_i] = nd - L_0 \cdot \mathbf{E}[Z'] \ge \gamma nd$. Substituting t gives

$$\mathbb{P}[Z'_1 + \cdots + Z'_{L_0} > nd] \le \exp\Big(-\frac{\gamma^2 n^2 d^2}{T^2 \cdot L_0}\Big) \le \exp\Big(-\Omega_d\Big(\frac{n}{\log^3 n}\Big)\Big). \tag{4.3}$$

Combining (4.2) and (4.3), for sufficiently large n,

$$\mathbb{P}[Z_1 + \cdots + Z_{L_0} > nd] \le \mathbb{P}[Z'_1 + \cdots + Z'_{L_0} > nd] + \Pr[Z_i \ne Z'_i \text{ for some } i] \le \frac{2}{n^{97}}.$$

Finally, the expected LCS length after nd bits is

$$\begin{aligned}\mathbf{E}[L_{\text{greedy}}] &\ge \mathbb{E}[\max(L : Z_1 + \cdots + Z_L \le nd)] \\ &\ge L_0 \cdot \Pr[Z_1 + \cdots + Z_{L_0} \le nd] \\ &\ge \frac{nd}{\mathbb{E}[Z]}(1 - \gamma) \cdot \Big(1 - \frac{2}{n^{97}}\Big) \ge n\Big(\frac{1}{2} + \frac{c_1}{\sqrt{d}}\Big)\end{aligned}$$

for some absolute constant $c_1 > 0$. Hence,

$$\gamma_{2,d} = \lim_{n\to\infty} \frac{W_{nd}}{n} \ge \lim_{n\to\infty} \frac{\mathbf{E}[L_{\text{greedy}}]}{n} \ge \frac{1}{2}\Big(1 + \frac{c_1}{\sqrt{d}}\Big). \qquad \square$$

4.2. Theorem 1.1, upper bound. We use the following lemma from [13] that counts superstrings of a string of a given length.

LEMMA 4.1 [13, Lemma 2.3]. *For any string w of length $\ell > n/k$, the number of strings of length n with w as a subsequence is at most*

$$n \cdot \binom{n-1}{\ell-1}(k-1)^{n-\ell}.$$

We remark that the result in [13] is stated for $\ell > (1 - 1/k)n$, and states that there are at most $\sum_{t=\ell}^{n} \binom{t-1}{\ell-1} k^{n-t}(k-1)^{t-\ell}$ subsequences. However, this bound comes from a counting argument and actually holds for all ℓ. For $\ell > n/k$, the summands increase with t, so bounding each summand by the $t = n$ summand gives the bound stated here.

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

PROOF OF THEOREM 1.1, UPPER BOUND. With hindsight, let $c_0 = 16$, and let $\ell = n(1+\varepsilon)/k$ where $\varepsilon = 4 \cdot \sqrt{\ln k/d}$. Assume $c_0 \log k < d$, so that $\varepsilon < 1$. By Lemma 4.1, for all strings w of length ℓ,

$$\Pr[X^1, \ldots, X^d \text{ have } w \text{ as a subsequence}] \le \left(\frac{n \cdot \binom{n}{n-\ell}(k-1)^{n-\ell}}{k^n}\right)^d.$$

By a union bound over all strings of length ℓ, taking $c_k = k/(4 \ln k)$ in Lemma 3.4,

$$\begin{aligned}
\Pr[\mathrm{LCS}(X^1, \ldots, X^d) \ge \ell] &\le k^\ell \cdot \left(\frac{n \cdot \binom{n}{n-\ell}(k-1)^{n-\ell}}{k^n}\right)^d \\
&\le n^d \cdot k^\ell \cdot \left(\frac{k^{nH_k(1-1/k-\varepsilon/k)}}{k^n}\right)^d \\
&\le n^d \cdot k^\ell \cdot \left(\frac{k^{n(1-c_k(\varepsilon/k)^2)}}{k^n}\right)^d \\
&\le n^d \cdot k^{2n/k} \cdot \left(\frac{k^{n(1-c_k(\varepsilon/k)^2)}}{k^n}\right)^d \\
&= n^d k^{-2n/k} < k^{-n/k}.
\end{aligned}$$

The second inequality uses Lemma 3.3 and the definition of ℓ. The third inequality uses Lemma 3.4. The fourth inequality follows from $\varepsilon < 1$. The equality follows from substituting c_k. Our bound on the expectation follows.

$$\begin{aligned}
\mathbf{E}[\mathrm{LCS}(X^1, \ldots, X^d)] &\le \ell \cdot \Pr[\mathrm{LCS}(X^1, \ldots, X^d) < \ell] + n \cdot \Pr[\mathrm{LCS}(X^1, \ldots, X^d) \ge \ell] \\
&\le \ell \cdot 1 + n \cdot k^{-n/k} \le \ell + o(1),
\end{aligned}$$

Taking the limit $n \to \infty$, we conclude $\gamma_{k,d} \le (1+\varepsilon)/k$, as desired. □

Appendix A. Binary lower bounds implies k-ary lower bounds

The k-ary lower bound in Theorem 1.2 follows from the binary lower bound in Theorem 1.1 because of the following lemma.

LEMMA A.1. $\gamma_{k,d} \ge 2\gamma_{2,d}/k$.

PROOF. Consider d random strings $X^1, \ldots, X^d$ over the alphabet $[k]$. Let $Y^1, \ldots, Y^d$ be the subsequences of $X^1, \ldots, X^d$ consisting of the symbols $\{1, 2\}$. By standard concentration arguments, the lengths $|Y^1|, \ldots, |Y^d|$ are all at least $n_0 = 2n/k - O_k(n^{2/3})$ with high probability $1 - 2^{\Omega_k(n^{1/3})}$. Conditioned on the lengths $|Y^1|, \ldots, |Y^d|$ all being at least n_0, the expected LCS of $Y^1, \ldots, Y^d$ is at least $\gamma_{2,d} \cdot n_0$. Thus,

$$\begin{aligned}
&\mathbf{E}[\mathrm{LCS}(X^1, \ldots, X^d)] \\
&\ge \mathbf{E}[\mathrm{LCS}(Y^1, \ldots, Y^d)] \\
&\ge \mathbf{E}[\mathrm{LCS}(Y^1, \ldots, Y^d) \mid |Y^1|, \ldots, |Y^d| \ge n_0] \cdot \Pr[|Y^1|, \ldots, |Y^d| \ge n_0] \\
&\ge \gamma_{2,d} \cdot n_0 \cdot (1 - 2^{-\Omega(n^{1/3})}) \ge \frac{2}{k}\gamma_{2,d} \cdot n \cdot (1 - o(1)),
\end{aligned}$$

and it follows that $\gamma_{k,d} \ge (2/k)\gamma_{2,d}$ □

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

Appendix B. List-decoding against deletions

We connect the generalised Chvátal–Sankoff constant to list-decoding against deletions. The connection uses Azuma's inequality.

LEMMA B.1. (Azuma's inequality). *Let $Z_1, Z_2, \ldots, Z_n$ be a martingale with bounded differences, that is, $|Z_{i+1} - Z_i| \le c$ for some constant c. Then, for any $\varepsilon \ge 0$,*

$$\Pr(|Z_n - \mathbb{E}[Z_n]| \ge \varepsilon) \le 2\exp\Big(-\frac{\varepsilon^2}{2nc^2}\Big).$$

A code is a subset of $[k]^n$. A random code C is obtained by sampling independent uniformly random strings from $[k]^n$. For $p \in (0,1)$ and an integer $d \ge 2$, a code C is $(p, d-1)$ list-decodable against deletions if any d strings $X^1, \ldots, X^d \in C$ satisfy $\mathrm{LCS}(X^1, \ldots, X^d) < (1-p)n$.

The first result in Proposition B.2 says that random codes of positive rate (with $|C| \ge 2^{\Omega(n)}$) are list-decodable against deletions with radius $p = 1 - \gamma_{k,d} - \varepsilon$. The second result says that random codes even of constant size are not list-decodable against deletions with radius $1 - \gamma_{k,d} + \varepsilon$. Thus, $1 - \gamma_{k,d}$ is the maximum fraction of deletions that a positive-rate random code list-decodes against with list-size d.

PROPOSITION B.2. *For all $\varepsilon > 0$, there exists a constant $c > 0$ such that a random code $C \subset [k]^n$ of size $|C| \ge 2^{cn}$ is $(1 - \gamma_{k,d} - \varepsilon, d-1)$ list-decodable against deletions. Furthermore, a random code of size d, with high probability, is not $(1 - \gamma_{k,d} + \varepsilon, d-1)$ list-decodable against deletions.*

PROOF. With hindsight, choose $c = \varepsilon^2/10d$. We construct the code C as a set of 2^{cn} independent random strings, each of length n, drawn from the alphabet $[k]$. We consider the longest common subsequence (LCS) of d codewords $X^1, X^2, \ldots, X^d$ from C.

The length $\mathrm{LCS}(X^1, X^2, \ldots, X^d)$ can be treated as a *martingale sequence* by revealing the symbols one at a time. Define Z_i as the expected value of the LCS length given the first i symbols of each sequence $X^1, \ldots, X^d$:

$$Z_i = \mathbb{E}[LCS(X^1, \ldots, X^d) \mid X^1[1, \ldots, i], \ldots, X^d[1, \ldots, i]].$$

Here, $Z_0, Z_1, \ldots, Z_n$ form a martingale, where

$$Z_0 = \mathbb{E}[LCS(X^1, \ldots, X^d)], \quad Z_n = LCS(X^1, X^2, \ldots, X^d).$$

Further, this martingale has bounded difference $|Z_{i+1} - Z_i| \le 1$. By Azuma's inequality, for any $\varepsilon > 0$,

$$\Pr(|LCS(X^1, X^2, \ldots, X^d) - \gamma_{k,d} n| \ge \varepsilon n) = \Pr[|Z_n - Z_0| \ge \varepsilon n] \le 2\exp\Big(-\frac{\varepsilon^2 n}{2}\Big). \quad \text{(B.2)}$$

This result implies that, with high probability, the LCS length is close to its expected value $\gamma_{k,d} n$. With large enough n, the probability that LCS exceeds $\gamma_{k,d} n$ is exponentially small. Thus, for each individual set of d codewords,

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

$$\Pr(LCS(X^1, X^2, \ldots, X^d) > (\gamma_{k,d} + \varepsilon)n) \le 2\exp\left(-\frac{\varepsilon^2 n}{2}\right).$$

By the union bound, the probability that any d-tuple of codewords in C violates this bound is at most

$$|C|^d \cdot 2\exp\left(-\frac{\varepsilon^2 n}{2}\right) \le 2^{-\Omega(n)}.$$

Thus, with high probability, $LCS(X^1, X^2, \ldots, X^d) \le (\gamma_{k,d} + \varepsilon)n$ for all codewords $X^1, X^2, \ldots, X^d \in C$, and our code is $(1 - \gamma_{k,d} - \varepsilon, d - 1)$ list-decodable against deletions.

To show the second result, note that, by (B.2), for d independent random strings $X^1, \ldots, X^d$,

$$\Pr(LCS(X^1, X^2, \ldots, X^d) > (\gamma_{k,d} - \varepsilon)n) \ge 1 - 2\exp\left(-\frac{\varepsilon^2 n}{2}\right),$$

so a random code of size d is *not* $(1 - \gamma_{k,d} + \varepsilon, d - 1)$ list-decodable with high probability. □

Acknowledgement

We thank Shamil Asgarli for helpful discussions.

References

[1] R. Baeza-Yates, R. Gavaldá, G. Navarro and R. Scheihing, 'A new approach to the longest common subsequence problem', *Algorithmica* **23** (1999), 107–122.
[2] A. Boutet de Monvel, 'Longest common subsequences for large alphabets', *Theoret. Comput. Sci.* **228** (1999), 45–60.
[3] B. Bukh and C. Cox, 'The length of the longest common subsequence of random permutations', *Random Structures Algorithms* **61**(2) (2022), 211–230.
[4] R. Bundschuh, 'An analysis of the longest common subsequence problem with lattice methods', *J. Phys. A* **34** (2001), 1665–1673.
[5] V. Chvátal and D. Sankoff, 'Longest common subsequences of two random sequences', *J. Appl. Probab.* **12**(2) (1975), 306–315.
[6] V. Dančík, *Expected Length of Longest Common Subsequences* (PhD Thesis, University of Warwick, 1994).
[7] V. Dančík, 'Common subsequences and supersequences and their expected length', *Combin. Probab. Comput.* **7**(4) (1998), 365–373.
[8] V. Dančík and M. S. Paterson, 'On the expected length of the longest common subsequence', *Algorithmica* **13** (1995), 51–60.
[9] J. G. Deken, 'Some limit results for longest common subsequences', *Discrete Math.* **26**(1) (1979), 17–31.
[10] V. Guruswami, B. Haeupler and A. Shahrasbi, 'Optimally resilient codes for list-decoding from insertions and deletions', in: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing* (eds. K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath and J. Chuzhoy) (ACM, New York, 2020), 524–537.
[11] V. Guruswami, X. He and R. Li, 'The zero-rate threshold for adversarial bit-deletions is less than 1/2', *IEEE Trans. Inform. Theory* **69**(4) (2022), 2218–2239.

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press

[12] V. Guruswami, A. Rudra and M. Sudan, *Essential Coding Theory* (draft). Available at http://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/.2025
[13] V. Guruswami and C. Wang, 'Deletion codes in the high-noise and high-rate regimes', *IEEE Trans. Inform. Theory* **63**(4) (2017), 1961–1970.
[14] T. Jiang and M. Li, 'On the approximation of shortest common supersequences and longest common subsequences', *SIAM J. Comput.* **24**(5) (1995), 1122–1139.
[15] I. A. Kash, M. Mitzenmacher, J. Thaler and J. Ullman, 'On the zero-error capacity threshold for deletion channels', in: *2011 Information Theory and Applications Workshop* (IEEE, New York, 2011), 1–5.
[16] M. Kiwi, M. Loebl and J. Matouěk, 'Expected length of the longest common subsequence for large alphabets', *Adv. Math.* **197**(2) (2005), 480–498.
[17] M. Kiwi and J. Soto, 'On a speculated relation between Chvátal–Sankoff constants of several sequences', *Combin. Probab. Comput.* **18**(4) (2009), 517–532.
[18] G. Lueker, 'Improved bounds on the average length of longest common subsequences', *J. ACM* **56** (2003), 130–131.
[19] G. S. Lueker, 'Improved bounds for the average length of longest common subsequences', *J. ACM* **56**(3) (2009), 1–38.
[20] M. Paterson and V. Dančík, 'Longest common subsequences', in: *Mathematical Foundations of Computer Science, 1994*, Lecture Notes in Computer Science, 841 (eds. I. Prívara, B. Rovan and P. Ruzička) (Springer, Berlin, 1994), 127–142.
[21] D. Sankoff and S. Mainville, 'Common subsequences and monotone subsequences', in: *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison* (eds. D. Sankoff and S. Mainville) (CLSI Publications, Stanford, CA, 1983), 363–365.
[22] D. Soiffer, A. Salls, C. Miller, D. Reichman, G. Sárközy and G. Heineman, 'Improved lower bounds on the expected length of longest common subsequences', *2025 IEEE International Symposium on Information Theory (ISIT)* (IEEE, New York, 2025), 6 pages. doi: 10.1109/ISIT63088.2025.11195592.
[23] J. M. Steele, 'Longest common subsequences: a probabilistic perspective', *SIAM J. Appl. Math.* **46**(6) (1986), 936–942.
[24] R. A. Wagner and M. J. Fischer, 'The string-to-string correction problem', *J. ACM* **21**(1) (1974), 168–173.

RAY LI, Department of Mathematics and Computer Science,
Santa Clara University, Santa Clara, 95053, USA
e-mail: rli6@scu.edu

WILLIAM REN, Department of Mathematics and Computer Science,
Santa Clara University, Santa Clara 95053, USA
e-mail: wren@scu.edu

YIRAN WEN, Department of Mathematics and Computer Science,
Santa Clara University, Santa Clara 95053, USA
e-mail: ywen@scu.edu

https://doi.org/10.1017/S000497272610104X Published online by Cambridge University Press