Discrete Applied Mathematics 353 (2024) 44–64

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Polynomial-time equivalences and refined algorithms for longest common subsequence variants☆

Yuichi Asahiro [a], Jesper Jansson [b], Guohui Lin [c], Eiji Miyano [d,*], Hirotaka Ono [e], Tadatoshi Utashima [d]

[a] Kyushu Sangyo University, Fukuoka, Japan
[b] Kyoto University, Kyoto, Japan
[c] University of Alberta, Edmonton, Canada
[d] Kyushu Institute of Technology, Iizuka, Japan
[e] Nagoya University, Nagoya, Japan

ARTICLE INFO

Article history:
Received 31 December 2023
Received in revised form 2 April 2024
Accepted 4 April 2024
Available online xxxx

Keywords:
Longest common subsequence
Repetition-bounded
Multiset-restricted
One-side-filled
Two-side-filled
Dynamic programming
Exact algorithm
Approximation algorithm

ABSTRACT

The problem of computing the longest common subsequence of two sequences (LCS for short) is a classical and fundamental problem in computer science. In this article, we study four variants of LCS: the Repetition-Bounded Longest Common Subsequence problem (RBLCS), the Multiset-Restricted Common Subsequence problem (MRCS), the Two-Side-Filled Longest Common Subsequence problem (2FLCS), and the One-Side-Filled Longest Common Subsequence problem (1FLCS). Although the original LCS can be solved in polynomial time, all these four variants are known to be NP-hard. Recently, an exact, $O(1.44225^n)$-time, dynamic programming (DP) based algorithm for RBLCS was proposed, where the two input sequences have lengths n and $poly(n)$. Here, we first establish that each of MRCS, 1FLCS, and 2FLCS is polynomially equivalent to RBLCS. Then, we design a refined DP-based algorithm for RBLCS that runs in $O(1.41422^n)$ time, which implies that MRCS, 1FLCS, and 2FLCS can also be solved in $O(1.41422^n)$ time. Finally, we give a polynomial-time 2-approximation algorithm for 2FLCS.

1. Introduction

1.1. Longest common subsequence problems with occurrence constraints

The problem of computing the longest common subsequence of two sequences (LCS for short) is a classical and fundamental problem in computer science [4,5,10,16]. Indeed, many polynomial-time algorithms have been published for LCS [9,10,12,16]. A natural extension of LCS is to impose constraints on the occurrences of the symbols in the solution. It has been shown that even very simple constraints may make the problem computationally much harder. As an example, the Repetition-Free Longest Common Subsequence problem (RFLCS), introduced by Adi et al. [1] is: Given two sequences X and Y over an alphabet Σ, the goal of RFLCS is to find a "*repetition-free*" longest common subsequence of X and Y,

☆ A preliminary version of this article appeared in Proceedings of 33rd Annual Symposium on Combinatorics Pattern Matching (CPM 2022), Article No. 12, pp.12:1–12:18, 2022 (Asahiroet al., 2022 [3]).

* Corresponding author.
E-mail addresses: asahiro@is.kyusan-u.ac.jp (Y. Asahiro), jj@i.kyoto-u.ac.jp (J. Jansson), guohui@ualberta.ca (G. Lin), miyano@ai.kyutech.ac.jp (E. Miyano), ono@nagoya-u.jp (H. Ono), utashima.tadatoshi965@mail.kyutech.jp (T. Utashima).

https://doi.org/10.1016/j.dam.2024.04.006
0166-218X/

where each symbol appears at most once in the obtained subsequence. Adi et al. [1] proved that RFLCS is APX-hard even if each symbol appears at most twice in each of the given sequences. On the positive side, they showed that RFLCS admits a polynomial-time occ_{max}-approximation algorithm, where occ_{max} is defined as follows: Let $occ(W, \sigma)$ be the number of occurrences of a symbol σ in a sequence W. Then occ_{max} is the maximum of $\min\{occ(X, \sigma), occ(Y, \sigma)\}$ taken over all σ's in two sequences X and Y.

Mincu and Popa [13] introduced a general form of RFLCS, called the Multiset Restricted Common Subsequence problem (MRCS): Given two sequences X and Y, and a multiset $\mathcal{M}$ over the alphabet Σ, the goal of MRCS is to find a common subsequence $Z_{\mathcal{M}}$ of X and Y, that contains the maximum number of symbols from $\mathcal{M}$. If $\mathcal{M} = \Sigma$, then MRCS is essentially equivalent to RFLCS. Therefore, MRCS is also APX-hard. In [13], the authors showed that there exists an exact algorithm solving MRCS with running time $O(|X||Y|(t+1)^{|\Sigma|})$, where t is the maximum multiplicity of symbols in $\mathcal{M}$. Also, they provided a polynomial-time $2\sqrt{\min\{|X|, |Y|\}}$-approximation algorithm for MRCS [13].

Recently, Asahiro et al. [2] introduced a slightly different generalization of RFLCS, called the Repetition-Bounded Longest Common Subsequence problem (RBLCS for short): Let $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ be an alphabet of k symbols and C_{occ} be an occurrence constraint $C_{occ} : \Sigma \to \mathbb{N}$, assigning an upper bound on the number of occurrences of each symbol in Σ. Given two sequences X and Y over the alphabet Σ and an occurrence constraint C_{occ}, the goal of RBLCS is to find a "*repetition-bounded*" longest common subsequence of X and Y, where each symbol σ_i appears at most $C_{occ}(\sigma_i)$-times in the obtained subsequence for $i = 1, 2, \ldots, k$. In [2], Asahiro et al. provided a dynamic programming (DP) based algorithm for RBLCS and proved that its running time is $O(1.44225^{|X|})$ for any occurrence constraint C_{occ}, assuming $|X| \leq |Y|$ and $|Y| = poly(|X|)$, and even less in certain special cases. In particular, for RFLCS, that algorithm runs in $O(1.41422^{|X|})$ time. NP-hardness and APX-hardness results for RBLCS on restricted instances were also shown in [2].

1.2. Longest common subsequence problems on incomplete sequences

The comparison of biological sequences is a widely investigated field of bioinformatics, in which the genomic features including DNA sequences and genes of different organisms are compared in order to identify biological differences and similarities. In genomic analyses, however, the considered genomes are usually incomplete and thus there are cases where we have to reconstruct complete genomes from incomplete genomes (so-called *scaffolds*) by filling in missing molecular data. For this purpose, Muñoz et al. [15] formulated the following combinatorial optimization problem, called the One-Sided Scaffold Filling problem (1SF): Given an incomplete genome Y, a multiset $\mathcal{M}$ of missing genes, and a reference genome X, the goal of 1SF is to insert the missing genes into Y so that the number of common adjacencies between the resulting Y^* and X is maximized. Subsequently, Jiang et al. [11] proposed the Two-Sided Scaffold Filling problem (2SF): Given two scaffolds (incomplete genomes), the goal of 2SF is to fill the missing genes into those two scaffolds respectively to result in such two genomes that the number of common adjacencies between them is maximized.

Inspired by methods for genome comparison based on LCS and by 1SF/2SF, Castelli et al. [6] introduced a new variant of LCS, called the One-Side-Filled Longest Common Subsequence problem (1FLCS), which aims to compare a complete sequence with an incomplete one, i.e., with some missing elements: Given a complete sequence X, an incomplete sequence Y, and a multiset $\mathcal{M}_Y$ of symbols missing in Y, 1FLCS asks for a sequence Y^+ obtained by inserting a subset of the symbols of $\mathcal{M}_Y$ into Y so that Y^+ induces a common subsequence with X of maximum length. The authors proved the APX-hardness of 1FLCS and designed a polynomial-time $\frac{5}{3}$-approximation algorithm for 1FLCS. They also presented an exponential-time exact algorithm for 1FLCS. (However, they did not analyze its time complexity in detail.) In [7], Castelli et al. showed that if the alphabet size $|\Sigma|$ is a constant, then there is a polynomial-time algorithm for 1FLCS, and concluded by introducing the Two-Side-Filled Longest Common Subsequence problem (2FLCS), i.e., LCS on two incomplete sequences and two multisets of missing symbols: Given two incomplete sequences X and Y, and two multisets $\mathcal{M}_X$ and $\mathcal{M}_Y$, 2FLCS asks for two sequences X^+ and Y^+ obtained by inserting subsets of the symbols of $\mathcal{M}_X$ and $\mathcal{M}_Y$ into X and Y, respectively, so that X^+ and Y^+ induce a common subsequence of maximum length. They conjectured that 2FLCS can be approximated within a constant factor in polynomial time, and that the following simple method gives a 2-approximation:

(1) Find a longest common subsequence Z_1 of X and Y. Let $\overline{X}$ and $\overline{Y}$ be the subsequences remaining after deleting the symbols corresponding to Z_1 from X and Y, respectively.
(2) Obtain a sequence Z_2 from $\overline{X}$ and $\overline{Y}$ that maximizes the number of symbols matched by inserting the symbols in $\mathcal{M}_X$ into $\overline{X}$ and the ones in $\mathcal{M}_Y$ into $\overline{Y}$, respectively.
(3) If $|Z_1| \geq |Z_2|$, then output Z_1; otherwise, output Z_2.

Moreover, they conjectured that 2FLCS can be solved in polynomial time if the alphabet size is a constant.

1.3. Our contributions

Suppose that there exist an $O(T_A)$-time algorithm for an optimization problem P_A and an $O(T_B)$-time algorithm for another optimization problem P_B. In this article, we say that two problems P_A and P_B are *polynomially equivalent*, or that *polynomial-time equivalence* between P_A and P_B holds, if an optimal solution for an instance I_A of P_A can be obtained in $O(T_B) + O(poly(|I_A|))$ time and an optimal solution for an instance I_B of P_B can be obtained in $O(T_A) + O(poly(|I_B|))$ time. Our contributions are:

1. We establish that MRCS is polynomially equivalent to RBLCS by showing the following: (i) From an input $(X, Y, \mathcal{M})$ of MRCS, we construct an input (X, Y, C_{occ}) of RBLCS in $O(poly(|(X, Y, \mathcal{M})|))$ time. Then, from an optimal solution Z_R of RBLCS on (X, Y, C_{occ}), we construct an optimal solution Z_M of MRCS on $(X, Y, \mathcal{M})$ in $O(poly(|(X, Y, \mathcal{M})|))$ time. Conversely, (ii) from an input (X, Y, C_{occ}) of RBLCS, we construct an input $(X, Y, \mathcal{M})$ of MRCS in $O(poly(|(X, Y, C_{occ})|))$ time. Then, from an optimal solution Z_M of MRCS on $(X, Y, \mathcal{M})$, we construct an optimal solution Z_R of RBLCS in $O(poly(|(X, Y, C_{occ})|))$ time. It is important to note that our constructions between two inputs are "input-sequences preserving reductions", i.e., X and Y in $(X, Y, \mathcal{M})$ and (X, Y, C_{occ}) are identical.
2. Similarly to the above, we show the polynomial-time equivalence between 1FLCS and RBLCS: (i) From an input $(X, Y, \mathcal{M}_Y)$ of 1FLCS, we construct an input (X, Y, C_{occ}) of RBLCS in $O(poly(|(X, Y, \mathcal{M}_Y)|))$ time. Then, from an optimal solution Z_R of RBLCS on (X, Y, C_{occ}), we construct an optimal solution Z_{1F} of 1FLCS on $(X, Y, \mathcal{M}_Y)$ in $O(poly(|(X, Y, \mathcal{M}_Y)|))$ time. Conversely, (ii) from an input (X, Y, C_{occ}) of RBLCS, we construct an input $(X, Y, \mathcal{M}_Y)$ of 1FLCS in $O(poly(|(X, Y, C_{occ})|))$ time. Then, from an optimal solution Z_{1F} of 1FLCS on $(X, Y, \mathcal{M}_Y)$, we construct an optimal solution Z_R of RBLCS on (X, Y, C_{occ}) in $O(poly(|(X, Y, C_{occ})|))$ time.
3. We prove the polynomial-time equivalence between 2FLCS and RBLCS. Due to the second contribution and 1FLCS being a special case of 2FLCS (see Remark 1 below), we only need to show one direction: (i) From an input $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS, we construct an input (X, Y, C_{occ}) of RBLCS in $O(poly(|(X, Y, \mathcal{M}_X, \mathcal{M}_Y)|))$ time. Then, from an optimal solution Z_R of RBLCS on (X, Y, C_{occ}), we construct an optimal solution Z_{2F} of 2FLCS on $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ in $O(poly(|Z_R|))$ time.
4. We design a refined DP-based algorithm that runs in $O(1.41422^n)$ time for RBLCS on two sequences X of length n and Y of length m (assuming that $n \leq m$ and $m = O(poly(n))$), while the previously known running time was $O(1.44225^n)$ in [2].
5. We give a simple polynomial-time 2-approximation algorithm for 2FLCS, thus resolving one of the conjectures in [7].

Remark 1. One sees that 1FLCS on $(X, Y, \mathcal{M}_Y)$ is equivalent to 2FLCS on $(X, Y, \emptyset, \mathcal{M}_Y)$; 1FLCS can be solved by using an algorithm for 2FLCS. From item (ii) in the second contribution, RBLCS can also be solved by using the algorithm for 2FLCS with some extra polynomial-time calculations. Therefore, the one-way equivalence in the third contribution demonstrates the "two-way" polynomial-time equivalence between 2FLCS and RBLCS. Furthermore, interestingly, an algorithm for 1FLCS can solve 2FLCS within an extra polynomial-time factor.

Remark 2. None of the constructions between inputs described above change the sequences X and Y. In particular, $|X|$ and $|Y|$ remain the same, so the above polynomial-time equivalences by the first, the second, and the third contributions imply that MRCS, 1FLCS, and 2FLCS can also be solved in $O(1.41422^n)$ time by the fourth contribution.

Remark 3. We also remark that the polynomial-time equivalence between 1FLCS and 2FLCS gives an affirmative answer to the conjecture on the polynomial-time solvability of 2FLCS for a constant size alphabet in [7] since we do not change Σ.

2. Preliminaries

2.1. Notation

An *alphabet* $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ is a set of k *symbols*. Let X be a sequence over the alphabet Σ and $|X|$ be the length of the sequence X. Throughout the article, a sequence X is often regarded as a *multiset* of the same symbols. For example, $X = \langle x_1, x_2, \ldots, x_n\rangle$ is a sequence of length n, where $x_i \in \Sigma$ for $1 \leq i \leq n$, i.e., $|X| = n$. A *subsequence* of X is obtained by deleting zero or more symbols from X. Then, we say that a sequence Z is a *common subsequence* of X and Y if Z is a subsequence of both X and Y. Given two sequences X and Y over the alphabet Σ as input, the goal of the Longest Common Subsequence problem (LCS) is to find a *longest* common subsequence of X and Y, which is denoted by $LCS(X, Y)$. Let $L(X, Y)$ denote the length of $LCS(X, Y)$.

For the sequence X, the *consecutive subsequence*, i.e., *substring* $\langle x_i, x_{i+1}, \ldots, x_j\rangle$ is denoted by $X_{i..j}$. Then, we define the ith *prefix* of X, for $i = 1, \ldots, n$, as $X_{1..i} = \langle x_1, x_2, \ldots, x_i\rangle$. Also, we define the ith *suffix* of X, for $i = 1, \ldots, n$, as $X_{i..n} = \langle x_i, x_{i+1}, \ldots, x_n\rangle$. $X_{1..n}$ is X.

Let $X = \langle x_1, x_2, \ldots, x_n\rangle$ and $Y = \langle y_1, y_2, \ldots, y_m\rangle$ be the given two sequences of length n and length m, respectively. Assume that $n \leq m$ and $m = poly(n)$ from now on. Suppose that $Z = \langle z_1, z_2, \ldots, z_p\rangle$ is a common subsequence with length p of X and Y. Then, we can consider two strictly increasing sequences $I_X = \langle i_1, i_2, \ldots, i_p\rangle$ of indices of X and $I_Y = \langle j_1, j_2, \ldots, j_p\rangle$ of indices of Y such that $z_\ell = x_{i_\ell} = y_{j_\ell}$ holds for each $\ell = 1, 2, \ldots, p$. We call the pair (I_X, I_Y) of such sequences an *index-expression* of the common sequence Z of X and Y. A pair (x_{i_ℓ}, y_{j_ℓ}) is called the ℓth *match*. Also, we say that the ℓth match is z_ℓ, x_{i_ℓ}, or y_{i_ℓ}.

For two sequences $A = \langle a_1, \ldots, a_i\rangle$ of length i and $B = \langle b_1, \ldots, b_j\rangle$ of length j, let $A \oplus B$ be the *concatenation* of A and B, i.e., the sequence $A \oplus B = \langle a_1, \ldots, a_i, b_1, \ldots, b_j\rangle$ of length $i + j$. For $X = \langle x_1, x_2, \ldots, x_n\rangle$ of length n, let $X \setminus \langle i\rangle$ denote the sequence obtained by deleting the ith symbol x_i from X, i.e., $X \setminus \langle i\rangle = X_{1..i-1} \oplus X_{i+1..n} = \langle x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\rangle$.

Similarly, for $1 \leq i_1 < i_2 < \cdots < i_p \leq n$, let $X \setminus \langle i_1, i_2, \ldots, i_p \rangle$ be the sequence obtained by deleting p symbols $x_{i_1}, x_{i_2}, \ldots, x_{i_p}$ from X.

Let $\mathcal{M}$ be a multiset of symbols in Σ and let $|\mathcal{M}|$ be the cardinality of $\mathcal{M}$. Let $occ(\mathcal{M}, \sigma)$ denote the occurrences (i.e., the *multiplicity*) of a symbol $\sigma \in \Sigma$ in a multiset $\mathcal{M}$. Let $\mathcal{M} \setminus \{\sigma^\ell\}$ be the multiset obtained by removing ℓ σ's from a multiset $\mathcal{M}$. Let $\mathcal{M} \setminus \{\sigma^*\}$ be the multiset obtained by removing all σ's from a multiset $\mathcal{M}$.

Consider a multiset $\mathcal{M}$ of cardinality ℓ and obtain an arbitrarily fixed sequence $M = \langle \mu_1, \mu_2, \ldots, \mu_\ell \rangle$ of ℓ symbols in $\mathcal{M}$, called a *sequence-expression* of the multiset $\mathcal{M}$. In the following, the multiset $\mathcal{M}$ is often regarded as its sequence-expression M; $\mathcal{M}$ and M are used interchangeably. Similarly to the above, for $1 \leq i_1 < i_2 < \cdots < i_p \leq \ell$, let $M \setminus \langle i_1, i_2, \ldots, i_p \rangle$ be the sequence obtained by deleting p symbols $\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_p}$ from M.

An algorithm ALG is called an α-approximation algorithm and ALG's approximation ratio is α if $OPT(x)/ALG(x) \leq \alpha$ holds for every input x of an LCS-type problem, where $ALG(x)$ and $OPT(x)$ are the lengths of solutions obtained by ALG and an optimal algorithm, respectively.

2.2. Repetition-bounded longest common subsequence

Recall that $occ(W, \sigma)$ is the number of occurrences of $\sigma \in \Sigma$ in a sequence W. Without loss of generality, we assume that two input sequences X and Y have all k symbols in Σ, and thus $occ(X, \sigma_i) \geq 1$ and $occ(Y, \sigma_i) \geq 1$ for every symbol σ_i. Let C_{occ} be an occurrence constraint, i.e., a function $C_{occ} : \Sigma \to \mathbb{N}$ assigning an upper bound on the number of occurrences of each symbol in Σ. The Repetition-Bounded Longest Common Subsequence problem (RBLCS) can be formally defined as follows [2]:

> Repetition-Bounded Longest Common Subsequence problem (RBLCS)
>
> **Input:** A pair of sequences X and Y over the alphabet Σ, and an occurrence constraint C_{occ}.
>
> **Goal:** Find a longest common subsequence Z of X and Y such that $occ(Z, \sigma) \leq C_{occ}(\sigma)$ is satisfied for every $\sigma \in \Sigma$.

We call Z a *repetition-bounded* longest common subsequence. Let $LCS(X, Y, C_{occ})$ denote the repetition-bounded longest common subsequence for the input triple (X, Y, C_{occ}). Also, $L(X, Y, C_{occ})$ denotes the length of $LCS(X, Y, C_{occ})$.

Example 1. Let (X, Y, C_{occ}) be an instance of RBLCS defined by:

$X = \langle t, g, t, c, a, c, g, t, g, a, a, g \rangle$, $Y = \langle a, t, g, c, a, t, g, g, a, c, a, g, c \rangle$; and
$C_{occ}(a) = 1, C_{occ}(c) = 1, C_{occ}(g) = 2, C_{occ}(t) = 1$.

$Z = \langle g, c, t, g, a \rangle$ of length five is an optimal solution of RBLCS since $occ(Z, a) = 1$, $occ(Z, c) = 1$, $occ(Z, g) = 2$, $occ(Z, t) = 1$, and $\sum_{\sigma \in \{a,c,g,t\}} C_{occ}(\sigma) = 5$, i.e., $L(X, Y, C_{occ}) = 5$. As a side note, $\langle t, g, c, a, t, g, a, a, g \rangle$ of length nine is an optimal solution of the original LCS.

Consider an input triple (X, Y, C_{occ}) of RBLCS and a feasible solution Z_R for (X, Y, C_{occ}). Then, for every $\sigma \in \Sigma$, the number of occurrences $occ(Z_R, \sigma)$ of σ must be bounded from above by $C_{occ}(\sigma)$. If $C_{occ}(\sigma') > \min\{occ(X, \sigma'), occ(Y, \sigma')\}$ for some σ', then the constraint C_{occ} is somewhere redundant. Therefore, if the input (X, Y, C_{occ}) of RBLCS satisfies $C_{occ}(\sigma) \leq \min\{occ(X, \sigma), occ(Y, \sigma)\}$ for every $\sigma \in \Sigma$, then we call (X, Y, C_{occ}) the *standard* input. Without loss of generality, we assume that every input of RBLCS is standard in the following.

2.3. Multiset restricted common subsequence

The formal definition of the Multiset Restricted Common Subsequence problem (MRCS) is as follows [13]:

> Multiset Restricted Common Subsequence problem (MRCS)
>
> **Input:** A pair of sequences X and Y, and a multiset $\mathcal{M}$ over the alphabet Σ.
>
> **Goal:** Find a common subsequence Z of X and Y such that Z contains the maximum number of symbols from $\mathcal{M}$.

That is, the goal of MRCS is to maximize $|\mathcal{M} \cap Z|$ as a multiset intersection or, equivalently, to minimize $|\mathcal{M} \setminus Z|$ as a multiset difference (if Z is regarded as the corresponding multiset). The optimal solution Z is denoted by $LCS(X, Y, \mathcal{M})$ in the following. The length of $LCS(X, Y, \mathcal{M})$ is denoted by $L(X, Y, \mathcal{M})$.

Example 2. Consider the following input triple $(X, Y, \mathcal{M})$ of MRCS:

$X = \langle t, g, t, c, a, c, g, t, g, a, a, g \rangle$, $Y = \langle a, t, g, c, a, t, g, g, a, c, a, g, c \rangle$, and
$\mathcal{M} = \{a, c, g, g, t\}$.

One sees that a common subsequence $\langle g, c, t, g, a \rangle$ of X and Y is an optimal solution of MRCS since $|\mathcal{M}| = 5$ and solutions of length five with all the symbols in $\mathcal{M}$ are equally as good as longer solutions. For example, the objective function value of a longer common subsequence $Z = \langle g, c, t, g, a, a, g \rangle$ is also five since $|\mathcal{M} \cap Z| = 5$.

2.4. Filled longest common subsequence

Let $\mathcal{M}_X$ ($\mathcal{M}_Y$, resp.) be a multiset of symbols in Σ. Then, we denote the cardinality of the multiset $\mathcal{M}_X$ ($\mathcal{M}_Y$, resp.) by $|\mathcal{M}_X|$ ($|\mathcal{M}_Y|$, resp.), i.e., $\sum_{\sigma\in\mathcal{M}_X} occ(\mathcal{M}_X,\sigma)$ ($\sum_{\sigma\in\mathcal{M}_Y} occ(\mathcal{M}_Y,\sigma)$, resp.). A *filling* X^+ (Y^+, resp.) of the sequence X (Y, resp.) is defined as a sequence obtained from X (Y, resp.) by inserting a subset of the symbols from $\mathcal{M}_X$ ($\mathcal{M}_Y$, resp.) into X (Y, resp.). That is, for some $0 \le p \le |\mathcal{M}_X|$ and $\mathcal{M}'_X = \{\chi_1, \ldots, \chi_p\} \subseteq \mathcal{M}_X$, the filling X^+ obtained by inserting $\mathcal{M}'_X$ into X is the following concatenation of $2p+1$ subsequences (some might be a *null* sequence):

$$X^+ = X_{1..j_1} \oplus \langle\chi_{i_1}\rangle \oplus X_{j_1+1..j_2} \oplus \langle\chi_{i_2}\rangle \oplus \cdots \oplus \langle\chi_{i_p}\rangle \oplus X_{j_p+1..n},$$

where $X = X_{1..j_1} \oplus X_{j_1+1..j_2} \oplus \cdots \oplus X_{j_p+1..n}$ and $\{i_1, \ldots, i_p\} = \{1, \ldots, p\}$. For some $0 \le q \le |\mathcal{M}_Y|$ and $\mathcal{M}'_Y = \{\psi_1, \ldots, \psi_q\} \subseteq \mathcal{M}_Y$, the filling Y^+ obtained by inserting $\mathcal{M}'_Y$ into Y is similarly defined. Let X^* and Y^* be fillings such that the length of $LCS(X^*, Y^*)$ is the longest among the length of $LCS(X^+, Y^+)$ over all pairs of X^+ and Y^+. The Two-Side-Filled Longest Common Subsequence problem (2FLCS) is defined as follows [7]:

> Two-Side-Filled Longest Common Subsequence problem (2FLCS)
>
> **Input:** A pair of sequences X and Y, and a pair of multisets $\mathcal{M}_X$ and $\mathcal{M}_Y$ over the alphabet Σ.
>
> **Goal:**Find two fillings X^* and Y^* such that the length of $LCS(X^*, Y^*)$ is the longest among the lengths of $LCS(X^+, Y^+)$ over all pairs of X^+ and Y^+.

In the following, the longest common subsequence $LCS(X^*, Y^*)$ of two fillings X^* and Y^* is written as $LCS(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$. The length of $LCS(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ is denoted by $L(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$. As a special case, if $\mathcal{M}_X = \emptyset$, then the problem is called the One-Side-Filled Longest Common Subsequence problem (1FLCS) [7]:

> One-Side-Filled Longest Common Subsequence problem (1FLCS)
>
> **Input:** A pair of sequences X and Y, and a multiset $\mathcal{M}_Y$ over the alphabet Σ.
>
> **Goal:** Find a filling Y^* such that the length of $LCS(X, Y^*)$ is the longest among the length of $LCS(X, Y^+)$ over all fillings Y^+.

Let $LCS(X, Y, \mathcal{M}_Y)$ and $L(X, Y, \mathcal{M}_Y)$ be the longest common subsequence $LCS(X, Y^*)$ and its length, respectively.

Example 3. Now we consider the following two sequences X and Y, and two multisets $\mathcal{M}_X$ and $\mathcal{M}_Y$, as input to 2FLCS:

$$X = \langle g, t, c, a, c, t, g, a\rangle,\ Y = \langle g, a, t, c, c, g, t, g\rangle,$$
$$\mathcal{M}_X = \{g, t\},\ \text{and}\ \mathcal{M}_Y = \{c, t, t\}$$

Here, for example, $occ(X, c) = 2$ and $occ(\mathcal{M}_Y, c) = 1$. One sees that for the input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$, an optimal pair of fillings is as follows:

$$X^* = \langle\underline{t}, g, t, c, a, c, \underline{g}, t, g, a\rangle \quad \text{and} \quad Y^* = \langle\underline{t}, g, \underline{t}, \underline{c}, a, t, c, c, g, t, g\rangle.$$

That is, the leftmost $\underline{t}$ and the seventh $\underline{g}$ in X^* are inserted into the original X from $\mathcal{M}_X$. For Y^*, the first, third, and fourth symbols ($\underline{t}$, $\underline{t}$, and $\underline{c}$, respectively) are inserted into Y from $\mathcal{M}_Y$. Then, the longest common subsequence $LCS(X^*, Y^*)$ of those fillings X^* and Y^* is $\langle t, g, t, c, a, c, g, t, g\rangle$. Note that $I_{X^*} = \langle 1, 2, 3, 4, 5, 6, 7, 8, 9\rangle$ and $I_{Y^*} = \langle 1, 2, 3, 4, 5, 7, 9, 10, 11\rangle$. One can verify that, for example, the first symbol t in $LCS(X^*, Y^*)$ originally comes from $\mathcal{M}_X$ and $\mathcal{M}_Y$, but the second symbol g comes from X and Y.

Now let $X^+ = \langle x_1, x_2, \ldots, x_n\rangle$ and $Y^+ = \langle y_1, y_2, \ldots, y_m\rangle$ be two fillings of X and Y, respectively. Let (I_{X^+}, I_{Y^+}) be an index-expression of a common subsequence of two fillings X^+ and Y^+. Then, the ℓth match (x_{i_ℓ}, y_{j_ℓ}) is one of the following four types of matches:

- $\mathcal{M}_X\mathcal{M}_Y$-match: x_{i_ℓ} and y_{i_ℓ} are inserted from $\mathcal{M}_X$ and $\mathcal{M}_Y$, respectively.
- $\mathcal{M}_X Y$-match: x_{i_ℓ} is inserted from $\mathcal{M}_X$ but y_{i_ℓ} is originally in Y.
- $X\mathcal{M}_Y$-match: x_{i_ℓ} is originally in X but y_{j_ℓ} is inserted from $\mathcal{M}_Y$.
- XY-match: x_{i_ℓ} and y_{i_ℓ} are originally in X and Y, respectively.

Let X^* and Y^* denote optimal fillings for the quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS. If there exists at least one symbol, say, σ, in $\mathcal{M}_Y$ that does not appear in an optimal filling Y^*, then the length of $LCS(X^*, Y^* \oplus \langle\sigma\rangle)$ is equal to one of $LCS(X^*, Y^*)$, which implies that $Y^* \oplus \langle\sigma\rangle$ is another optimal filling. Similarly, if $\sigma' \in \mathcal{M}_X$ does not appear in X^*, then $X^* \oplus \langle\sigma'\rangle$ is another optimal filling. Therefore, without loss of generality, we assume that all the symbols in $\mathcal{M}_X$ and $\mathcal{M}_Y$ are inserted to the optimal fillings.

2.5. Known results on exact/approximation algorithms

Here, we summarize the previously known results on exact and approximation algorithms. For RBLCS, the following exact exponential-time algorithm is known:

Proposition 1 ([2]). *There is an $O(1.44225^n)$-time algorithm for RBLCS on two sequences X and Y, where $|X| = n$, $|Y| = m$, and $n \leq m$, assuming that $m = poly(n)$.*

If $C_{occ}(\sigma) = 1$ for every symbol $\sigma \in \Sigma$, then a faster exact algorithm can be designed:

Proposition 2 ([2]). *There is an $O(1.41422^n)$-time algorithm for RFLCS on two sequences X and Y, where $|X| = n$, $|Y| = m$, and $n \leq m$, assuming that $m = poly(n)$.*

Furthermore, the following approximation algorithm is known for RFLCS:

Proposition 3 ([1]). *There is a polynomial-time occ_{max}-approximation algorithm for RFLCS on X and Y, where $occ_{max} = \max_{\sigma\in\Sigma}\{\min\{occ(X,\sigma), occ(Y,\sigma)\}\}$.*

For MRCS, the following exact exponential-time algorithm and the polynomial-time approximation algorithm are proposed in [13]:

Proposition 4 ([13]). *There is an $O(nm(t+1)^k)$-time algorithm for MRCS on two sequences X and Y, and a multiset $\mathcal{M}$, where t and k are the maximum multiplicity of $\mathcal{M}$ and the alphabet size $|\Sigma|$, respectively.*[1]

Proposition 5 ([13]). *There is a polynomial-time $2\sqrt{\min\{n,m\}}$-approximation algorithm for MRCS on two sequences X and Y, and a multiset $\mathcal{M}$, where $|X| = n$ and $|Y| = m$.*

For 1FLCS, an FPT-algorithm parameterized by the number k of $X\mathcal{M}_Y$-matches in the optimal subsequence is known [7]. Note that k may be as large as the length of X, i.e., n.

Proposition 6 ([7]). *There is an $O(2^{O(k)}poly(n+m+|\mathcal{M}_Y|))$-time algorithm for 1FLCS on an input triple $(X, Y, \mathcal{M}_Y)$ if the number of $X\mathcal{M}_Y$-matches in $LCS(X, Y^*)$ is k.*

The following algorithm for 1FLCS runs in polynomial time if $|\Sigma|$ is a constant [7]:

Proposition 7 ([7]). *There is an $O(n^{|\Sigma|+2}m)$-time algorithm for 1FLCS on $(X, Y, \mathcal{M}_Y)$.*

The following approximability result is also known for 1FLCS:

Proposition 8 ([7]). *There is a polynomial-time $\frac{5}{3}$-approximation algorithm for 1FLCS.*

3. Polynomial-time equivalence of RBLCS and MRCS

In this section we show the polynomial-time equivalence between RBLCS and MRCS. First consider any optimal solution $Z_{\mathcal{M}}$ for an input $(X, Y, \mathcal{M})$ of MRCS. Recall that the objective function value of MRCS is $|\mathcal{M} \cap Z_{\mathcal{M}}|$. That is, $|\mathcal{M} \cap Z_{\mathcal{M}}|$ can be regarded as the summation of occurrences of all the symbols appearing both in $\mathcal{M}$ and in the solution $Z_{\mathcal{M}}$. Furthermore, intuitively, the number $occ(\mathcal{M}, \sigma)$ of occurrences of every symbol $\sigma \in \mathcal{M}$ can be regarded as the occurrence constraint $C_{occ}(\sigma)$ of the solution for RBLCS, and vice versa. One sees that we can transform from/to a multiset $\mathcal{M}$ of symbols in Σ to/from an occurrence constraint C_{occ} of symbols in Σ such that $C_{occ}(\sigma) = occ(\mathcal{M}, \sigma)$ for every $\sigma \in \Sigma$ clearly in polynomial time; all we have to do is count the multiplicity/occurrences of every symbol in $\mathcal{M}$. Then, we can obtain the following theorem:

Theorem 1. *Consider a pair of a multiset $\mathcal{M}$ in an input for MRCS and an occurrence constraint C_{occ} of symbols in Σ in an input for RBLCS such that $C_{occ}(\sigma) = occ(\mathcal{M}, \sigma)$ for every $\sigma \in \Sigma$. Then, the followings hold: (1) Given an optimal solution Z_R for an input (X, Y, C_{occ}) of RBLCS, we can obtain an optimal solution for an input $(X, Y, \mathcal{M})$ of MRCS in polynomial time. (2) Given an optimal solution $Z_{\mathcal{M}}$ for an input $(X, Y, \mathcal{M})$ of MRCS, we can obtain an optimal solution for an input (X, Y, C_{occ}) of RBLCS in polynomial time.*

[1] We remark that the time complexity shown in Theorem 3 of [13] is $O(nmt^k)$, but the correct one must be $O(nm(t+1)^k)$ because the algorithm has to store $t+1$ values from 0 through t for the maximum multiplicity. As described before, if $\mathcal{M} = \Sigma$, i.e., $t = 1$, then MRCS is essentially equivalent to RFLCS and thus MRCS is NP-hard. If we could solve MRCS with $t = 1$ in $O(nmt^k) = O(nm)$ time, then that would imply P = NP.

Proof. In the following, we show that (1) if an optimal solution Z_R for RBLCS on (X, Y, C_{occ}) is given, then Z_R itself is an optimal solution for MRCS on $(X, Y, \mathcal{M})$, and (2) if an optimal solution $Z_{\mathcal{M}}$ for MRCS on $(X, Y, \mathcal{M})$ is given, then the subsequence consisting of symbols in $\mathcal{M} \cap Z_{\mathcal{M}}$ is an optimal solution for RBLCS on (X, Y, C_{occ}). Here we make the following observations:

(i) Suppose that we are now given an optimal solution Z_R for RBLCS on (X, Y, C_{occ}). Note that Z_R is a feasible solution for MRCS since any common subsequence of X and Y is feasible for MRCS on the triple $(X, Y, \mathcal{M})$. Then, considering a multiset intersection $\mathcal{M} \cap Z_R$, we get $|\mathcal{M} \cap Z_R| = \sum_{\sigma \in \Sigma} \min\{occ(Z_R, \sigma), occ(\mathcal{M}, \sigma)\} = \sum_{\sigma \in \Sigma} occ(Z_R, \sigma) = |Z_R|$ since Z_R satisfies the occurrence constraint $C_{occ}(\sigma)$ for every σ.

(ii) Suppose that we are now given an optimal solution $Z_{\mathcal{M}}$ for MRCS on the triple $(X, Y, \mathcal{M})$. Then, we can obtain another (probably shorter) optimal solution $Z'_{\mathcal{M}}$ for the same input that satisfies $Z'_{\mathcal{M}} = \mathcal{M} \cap Z_{\mathcal{M}}$ in polynomial time by removing arbitrarily every symbol in $Z_{\mathcal{M}} \setminus \mathcal{M}$. Note that for every $\sigma \in \Sigma$, $occ(Z'_{\mathcal{M}}, \sigma) \leq occ(\mathcal{M}, \sigma) = C_{occ}(\sigma)$. That is, $Z'_{\mathcal{M}}$ is a feasible solution for RBLCS on (X, Y, C_{occ}).

From the above observations, $|Z_R| = |\mathcal{M} \cap Z_{\mathcal{R}}| \leq |\mathcal{M} \cap Z_{\mathcal{M}}| = |Z'_{\mathcal{M}}|$ holds since $Z_{\mathcal{M}}$ is optimal for MRCS, and $|Z_R| \geq |\mathcal{M} \cap Z_{\mathcal{M}}| = |Z'_{\mathcal{M}}|$ holds since Z_R is optimal for RBLCS. Namely, $|Z_R| = |Z'_{\mathcal{M}}|$ holds. Therefore, (1) given an optimal solution Z_R for an input (X, Y, C_{occ}) of RBLCS, we can regard Z_R as an optimal solution for an input $(X, Y, \mathcal{M})$ of MRCS; (2) given an optimal solution $Z_{\mathcal{M}}$ for an input $(X, Y, \mathcal{M})$ of MRCS, we can obtain $\mathcal{M} \cap Z_{\mathcal{M}}$ as an optimal solution for an input (X, Y, C_{occ}) of RBLCS in polynomial time. □

4. Polynomial-time equivalence of RBLCS , 1FLCS, and 2FLCS

4.1. Proof tools

In this subsection we give some proof tools. Now, we introduce two partially cyclic permutations, for example, one of which transforms a sequence $\langle 1, 2, 3, 4, 5, 6\rangle$ to $\langle 1, 3, 4, 5, 2, 6\rangle$. More precisely, we define the following two permutations $\delta_{i,j+}$ and $\delta_{i,j-}$, and two sequences $X^{\langle i\rangle \to \langle j\rangle +}$ and $X^{\langle i\rangle \to \langle j\rangle -}$ for a sequence $X = \langle x_1, x_2, \ldots, x_n\rangle$ as follows:

Definition 1. (1) The *(cyclic) index permutation* $\delta_{i,j+}$ is defined as follows:

(i) if $i < j$, then $\delta_{i,j+}(\ell) = \begin{cases} \ell & 1 \leq \ell \leq i-1 \text{ or } j+1 \leq \ell \leq n \\ j & \ell = i \\ \ell - 1 & i+1 \leq \ell \leq j; \text{ and} \end{cases}$

(ii) if $i > j$, then $\delta_{i,j+}(\ell) = \begin{cases} \ell & 1 \leq \ell \leq j \text{ or } i+1 \leq \ell \leq n \\ j+1 & \ell = i \\ \ell + 1 & j+1 \leq \ell \leq i-1. \end{cases}$

(2) Let $X^{\langle i\rangle \to \langle j\rangle +}$ denote a sequence obtained by removing the ith symbol x_i from X and then inserting it right after the jth symbol of X for $i \neq j$, i.e., for the inverse $\delta^{-1}_{i,j+}$ of $\delta_{i,j+}$,

$$X^{\langle i\rangle \to \langle j\rangle +} = \langle x_{\delta^{-1}_{i,j+}(1)}, x_{\delta^{-1}_{i,j+}(2)}, \ldots, x_{\delta^{-1}_{i,j+}(n)}\rangle.$$

One sees that $\langle 1, 2, 3, 4, 5, 6\rangle$ is permuted to $\langle 1, 3, 4, 5, 2, 6\rangle$ by $\delta_{2,5+}$.

Definition 2. (1) The *(cyclic) index permutation* $\delta_{i,j-}$ is defined as follows:

(i) if $i < j$, then $\delta_{i,j-}(\ell) = \begin{cases} \ell & 1 \leq \ell \leq i-1 \text{ or } j \leq \ell \leq n \\ j-1 & \ell = i \\ \ell - 1 & i+1 \leq \ell \leq j-1; \text{ and} \end{cases}$

(ii) if $i > j$, then $\delta_{i,j-}(\ell) = \begin{cases} \ell & 1 \leq \ell \leq j-1 \text{ or } i+1 \leq \ell \leq n \\ j & \ell = i \\ \ell + 1 & j \leq \ell \leq i-1. \end{cases}$

(2) Let $X^{\langle i\rangle \to \langle j\rangle -}$ denote a sequence obtained by removing the ith symbol x_i from X and then inserting it right before the jth symbol of X for $i \neq j$, i.e., for the inverse $\delta^{-1}_{i,j-}$ of $\delta_{i,j-}$,

$$X^{\langle i\rangle \to \langle j\rangle -} = \langle x_{\delta^{-1}_{i,j-}(1)}, x_{\delta^{-1}_{i,j-}(2)}, \ldots, x_{\delta^{-1}_{i,j-}(n)}\rangle.$$

The first tool reduces the numbers of XY-matches and $\mathcal{M}_X\mathcal{M}_Y$-matches in an output subsequence.

Lemma 1. *Suppose that $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ is an input for 2FLCS, and X^* and Y^* are optimal fillings of $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$. Also, suppose that the numbers of XY-matches, $\mathcal{M}_X\mathcal{M}_Y$-matches, $X\mathcal{M}_Y$-matches, and $\mathcal{M}_XY$-matches of some σ in the index-expression (I_{X^*}, I_{Y^*}) of X^* and Y^* are $\alpha > 0$, $\beta > 0$, $\zeta \geq 0$, and $\eta \geq 0$, respectively. Then, we can obtain in polynomial time another pair of optimal fillings X^{**} and Y^{**} such that (i) the numbers of XY-matches, $\mathcal{M}_X\mathcal{M}_Y$-matches, $X\mathcal{M}_Y$-matches, and*

$\mathcal{M}_X Y$-matches of σ in the index-expression $(I_{X^{**}}, I_{Y^{**}})$ of X^{**} and Y^{**} are $\alpha - 1$, $\beta - 1$, $\zeta + 1$, and $\eta + 1$, respectively, and (ii) all the matches of any different symbol $\sigma' \neq \sigma$ do not change.

Proof. Let $X^* = \langle x_1, \ldots, x_n \rangle$, $Y^* = \langle y_1, \ldots, y_m \rangle$, $I_{X^*} = \langle i_1, \ldots, i_k \rangle$, and $I_{Y^*} = \langle j_1, \ldots, j_k \rangle$ assuming that $L(X^*, Y^*) = k$. Now assume that the numbers of XY-matches, $\mathcal{M}_X\mathcal{M}_Y$-matches, $X\mathcal{M}_Y$-matches, and $\mathcal{M}_X Y$-matches of the symbol σ in the index-expression (I_{X^*}, I_{Y^*}) are α, β, ζ, and η, respectively, where $1 \leq \alpha, \beta \leq k$. Suppose that the pth match (x_{i_p}, y_{j_p}) is an $\mathcal{M}_X\mathcal{M}_Y$-match of σ and the qth match (x_{i_q}, y_{j_q}) is an XY-match of σ.

Now recall that any symbol in the multiset $\mathcal{M}_X$ ($\mathcal{M}_Y$, resp.) can be inserted into any position in X (Y, resp.). Thus, $X^{*\langle p\rangle \to \langle q\rangle +}$ ($Y^{*\langle p\rangle \to \langle q\rangle -}$, resp.) must be a valid filling of X and $\mathcal{M}_X$ (Y and $\mathcal{M}_Y$, resp.) since the symbol of the pth match comes originally from $\mathcal{M}_X$ or $\mathcal{M}_Y$.

Consider the following two sequences of indices for the case where $p \neq q + 1$ and $p \neq q - 1$:

$$I = \langle \delta_{p,q+}(i_1), \ldots, \delta_{p,q+}(i_q), \delta_{p,q+}(i_p), \delta_{p,q+}(i_{q+1}), \ldots, \delta_{p,q+}(i_k) \rangle; \text{ and}$$
$$I' = \langle \delta_{p,q-}(j_1), \ldots, \delta_{p,q-}(j_{q-1}), \delta_{p,q-}(j_p), \delta_{p,q-}(j_q), \ldots, \delta_{p,q-}(j_k) \rangle.$$

If $p = q+1$, then we set $I = I \setminus \langle \delta_{p,q+}(i_p) \rangle$ since $\delta_{p,q+}(i_p)$ is identical to $\delta_{p,q+}(i_{q+1})$; if $p = q-1$, then we set $I' = I' \setminus \langle \delta_{p,q-}(j_p) \rangle$ since $\delta_{p,q-}(j_p)$ is identical to $\delta_{p,q-}(j_{q-1})$. One can verify that (i) if $p < q$, then the $(q-1)$st match $(\delta_{p,q+}(i_q)$ in I and $\delta_{p,q-}(j_p)$ in $I')$ is an $X\mathcal{M}_Y$-match of σ and the qth match $(\delta_{p,q+}(i_p)$ in I and $\delta_{p,q-}(j_q)$ in $I')$ is an $\mathcal{M}_X Y$-match of σ, and (ii) if $p > q$, then the qth match $(\delta_{p,q+}(i_q)$ in I and $\delta_{p,q-}(j_p)$ in $I')$ is an $X\mathcal{M}_Y$-match of σ, and the $(q+1)$st match $(\delta_{p,q+}(i_p)$ in I and $\delta_{p,q-}(j_q)$ in $I')$ is an $\mathcal{M}_X Y$-match of σ. Furthermore, (I, I') is the index-expression of a common subsequence of $X^{*\langle p\rangle \to \langle q\rangle +}$ and $Y^{*\langle p\rangle \to \langle q\rangle -}$. Hence, the numbers of XY-matches and $\mathcal{M}_X\mathcal{M}_Y$-matches decrease by one; on the other hand, the numbers of $X\mathcal{M}_Y$-matches and $\mathcal{M}_X Y$-matches increase by one. Since the length of (I, I') is k, $X^{*\langle p\rangle \to \langle q\rangle +}$ and $Y^{*\langle p\rangle \to \langle q\rangle -}$ are also optimal fillings. The running time to obtain those two sequences $X^{*\langle p\rangle \to \langle q\rangle +}$ and $Y^{*\langle p\rangle \to \langle q\rangle -}$ is clearly in polynomial. □

If we use the above tool iteratively α-times for $\alpha \leq \beta$ (β-times for $\beta \leq \alpha$, resp.), then we can obtain so-called an "XY-match-free" ("$\mathcal{M}_X\mathcal{M}_Y$-match-free", resp.) output subsequence.

Lemma 2. *Suppose that an input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ satisfies $occ(X, \sigma) > 0$ and $occ(\mathcal{M}_Y, \sigma) > 0$ for some $\sigma \in \Sigma$. Let $X = \langle x_1, \ldots, x_n \rangle$ and $\mathcal{M}_Y = \langle \psi_1, \ldots, \psi_\ell \rangle$. Then,*

$$L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) = \max_{\sigma = x_i = \psi_j} L(X \setminus \langle i \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j \rangle) + 1.$$

Proof. Suppose that for a symbol $\sigma \in \Sigma$ and a pair of i and j such that $\sigma = x_i = \psi_j$, $X_1^+ = \langle x_{1,1}^+, \ldots, x_{1,n'}^+ \rangle$ and $Y_1^+ = \langle y_{1,1}^+, \ldots, y_{1,m'}^+ \rangle$ are two optimal fillings of $(X \setminus \langle i \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j \rangle)$. Also suppose that $LCS(X_1^+, Y_1^+)$ is Z_1, where $|Z_1| = k - 1$, and (A, B) is the index-expression of Z_1, where $A = \langle a_1, \ldots, a_{k-1} \rangle$ and $B = \langle b_1, \ldots, b_{k-1} \rangle$. Since X_1^+ is a filling of $X \setminus \langle i \rangle$ and $\mathcal{M}_X$, $X \setminus \langle i \rangle$ is a subsequence of X_1^+. Therefore, there exists a strictly increasing sequence $C = \langle c_1, \ldots, c_{k-1} \rangle$ such that for each $1 \leq t < i$, $x_{1,c_t}^+ = x_t$, and for each $i < t \leq k - 1$, $x_{1,c_t-1}^+ = x_t$. Here consider a sequence $X_2^+ = \langle x_{1,1}^+, \ldots, x_{1,c_i-1}^+, x_i, x_{1,c_i}^+, \ldots, x_{1,n'}^+ \rangle$. One sees that X_2^+ is a filling of X and $\mathcal{M}_X$.

Assume that $a_p < c_i \leq a_{p+1}$ holds for some p, where $1 \leq p \leq k - 2$. Consider $Y_2^+ = \langle y_{1,1}^+, \ldots, y_{1,b_p}^+, \psi_j, y_{1,b_p+1}^+, \ldots, y_{1,m'}^+ \rangle$. Then, Y_2^+ is a filling of Y and $\mathcal{M}_Y$. It follows that the pair of $I = \langle a_1, \ldots, a_p, i, a_{p+1} + 1, \ldots, a_{k-1} + 1 \rangle$ and $J = \langle b_1, \ldots, b_p, b_p + 1, b_{p+1} + 1, \ldots, b_{k-1} + 1 \rangle$ must be the index-expression of a common subsequence $Z_2 = \langle x_{a_1}, \ldots, x_{a_p}, x_i, x_{a_{p+1}+1}, \ldots, x_{a_{k-1}+1} \rangle$ of X_2^+ and Y_2^+. Since the length of I and J is k, the longest length $L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) \geq |Z_2| = |Z_1| + 1$ for some symbol σ.

Note that the above discussions can be applied to the case where the symbol σ is matched at the head or tail position. For example, if $i = 1$, then we consider $X_2^+ = \langle x_i, x_{1,1}^+, \ldots, x_{1,c_i-1}^+, x_{1,c_i}^+, \ldots, x_{1,n'}^+ \rangle$ as a filling of X and $\mathcal{M}_X$. As another example, if $j = k - 1$, then we consider $Y_2^+ = \langle y_{1,1}^+, \ldots, y_{1,b_p}^+, y_{1,b_p+1}^+, \ldots, y_{1,m'}^+, \psi_j \rangle$ as a filling of Y and $\mathcal{M}_Y$. Furthermore, for every match of every symbol, we can follow the same lines as the above. Hence, the following inequality holds:

$$L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) \geq \max_{\sigma = x_i = \psi_j} L(X \setminus \langle i \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j \rangle) + 1.$$

Next, suppose that a solution of $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ is a pair of X^* and Y^*. Also suppose that the index-expression of $Z = LCS(X^*, Y^*)$ is (I_{X^*}, I_{Y^*}) where $I_{X^*} = \langle i_1, \ldots, i_k \rangle$ and $I_{Y^*} = \langle j_1, \ldots, j_k \rangle$.

(1) If (I_{X^*}, I_{Y^*}) has an $X\mathcal{M}_Y$-match of a symbol σ at the qth position, then we can obtain a common subsequence $Z \setminus \langle q \rangle$ of two fillings $X^* \setminus \langle i_q \rangle$ and $Y^* \setminus \langle j_q \rangle$ by removing σ at the qth position from X^* and Y^*.

(2) If (I_{X^*}, I_{Y^*}) does not have any $X\mathcal{M}_Y$-match of σ, then, from Lemma 1, either (i) all the matches of σ in (I_{X^*}, I_{Y^*}) are XY-matches or (ii) all the matches of σ in (I_{X^*}, I_{Y^*}) are $\mathcal{M}_X\mathcal{M}_Y$-matches. (i) Suppose that all are XY-matches. Then, for j such that $\sigma = \psi_j$, $L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) = L(X, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j \rangle)$ is satisfied since any other matches are not affected. Furthermore, if the ith symbol x_i of X is σ and x_i is removed from X, then the number of matches is reduced by at most one. Hence, $L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) - 1 \leq L(X \setminus \langle i \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y) = L(X \setminus \langle i \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j \rangle)$ holds. (ii) Suppose that all are $\mathcal{M}_X\mathcal{M}_Y$-matches. One sees that if $\psi_j = \sigma$ is removed, then the number of matches is reduced by at most one. Also, even if $x_i = \sigma$ is removed, any other matches do not change. Therefore, $L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) - 1 \leq L(X \setminus \langle i \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j \rangle)$ holds.

Therefore, by applying a similar discussion to all matches in (I_{X^*}, I_{Y^*}), we have

$$L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) \leq \max_{\sigma=x_i=\psi_j} L(X \setminus \langle i\rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j\rangle) + 1.$$

In summary,

$$L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) = \max_{\sigma=x_i=\psi_j} L(X \setminus \langle i\rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j\rangle) + 1$$

holds. □

We can apply very similar arguments to the pair Y and $\mathcal{M}_X$, which gives:

Corollary 1. *Suppose that an input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ satisfies $occ(Y, \sigma) > 0$ and $occ(\mathcal{M}_X, \sigma) > 0$ for some $\sigma \in \Sigma$. Let $Y = \langle y_1, \ldots, y_m\rangle$ and $\mathcal{M}_X = \langle \chi_1, \ldots, \chi_\ell\rangle$. Then,*

$$L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) = \max_{\sigma=y_i=\chi_j} L(X, Y \setminus \langle i\rangle, \mathcal{M}_X \setminus \langle j\rangle, \mathcal{M}_Y) + 1.$$

The following lemma and corollary deal with symbol additions to multisets:

Lemma 3. *Let X^+ be a filling of X and $\mathcal{M}_X$, and let Y^+ be a filling of Y and $\mathcal{M}_Y$. Suppose that a common subsequence Z of X^+ and Y^+ satisfies $occ(Z, \sigma) < occ(Y^+, \sigma)$ for some symbol $\sigma \in \Sigma$. Then, we can find in polynomial time a new filling X^{++} of X and $\mathcal{M}_X \cup \{\sigma\}$ and a common subsequence Z' of X^{++} and Y^+ satisfying the following conditions: (1) $occ(Z, \sigma)+1 = occ(Z', \sigma)$, and (2) for every σ' except for σ $occ(Z, \sigma') = occ(Z', \sigma')$.*

Proof. Suppose that $X^+ = \langle x_1, \ldots, x_n\rangle$ and $Y^+ = \langle y_1, \ldots, y_m\rangle$. Also, suppose that (A, B) is the index-expression of $Z = LCS(X^+, Y^+)$ where $A = \langle a_1, \ldots, a_k\rangle$ and $B = \langle b_1, \ldots, b_k\rangle$. Since $occ(Z, \sigma) < occ(Y^+, \sigma)$, there exists an index i such that $y_i = \sigma$ and $i \notin B$. For ease of exposition, assume that $b_p < i < b_{p+1}$ holds for some $1 \leq p \leq k-1$ (even if $y_{b_1} = \sigma$ or $y_{b_k} = \sigma$, we can have the same discussion as the following). Then we consider $X^{++} = \langle x_1, \ldots, x_{a_p}, \sigma, x_{a_p+1}, \ldots, x_n\rangle$, $I = \langle a_1, \ldots, a_p, a_p + 1, a_{p+1} + 1, \ldots, a_k + 1\rangle$ and $J = \langle b_1, \ldots, b_p, i, b_{p+1}, \ldots, b_k\rangle$. Then, we can obtain a common subsequence $Z' = \langle y_{b_1}, \ldots, y_{b_p}, y_i, y_{b_{p+1}}, \ldots, y_{b_k}\rangle$ of X^{++} and Y^+ that has the index-expression (I, J). One sees that $Z' \setminus \langle p+1\rangle = Z$ and $y_i = \sigma$. Therefore, Z' satisfies $occ(Z, \sigma) + 1 = occ(Z', \sigma)$ and $occ(Z, \sigma') = occ(Z', \sigma')$ for every $\sigma' \neq \sigma$. Clearly, we can construct X^{++} as follows: (i) Scan Y^+ to find the index i, (ii) scan B to find the index p, and (iii) insert the corresponding symbol into X^+. The running times is $O(|X^+| + |Y^+|)$. This completes the proof. □

From the symmetry of X and Y, we obtain:

Corollary 2. *Let X^+ be a filling of X and $\mathcal{M}_X$, and let Y^+ be a filling of Y and $\mathcal{M}_Y$. Suppose that a common subsequence Z of X^+ and Y^+ satisfies $occ(Z, \sigma) < occ(X^+, \sigma)$ for some symbol $\sigma \in \Sigma$. Then, we can find in polynomial time a new filling Y^{++} of Y and $\mathcal{M}_Y \cup \{\sigma\}$ and a common subsequence Z' of Y^{++} and X^+ satisfying the following conditions: (1) $occ(Z, \sigma)+1 = occ(Z', \sigma)$, and (2) for every σ' except for σ, $occ(Z, \sigma') = occ(Z', \sigma')$.*

4.2. RBLCS and 1FLCS

In this subsection we show that 1FLCS is polynomially equivalent to RBLCS. Consider an input triple $(X, Y, \mathcal{M}_Y)$ of 1FLCS. In [14], Mincu and Popa observed that a filling-procedure of a symbol $\sigma \in \mathcal{M}_Y$ into Y to match some σ in X can be seen as a deleting-procedure of the matched σ from X [14]. Our basic ideas are based on their observation: Every symbol $\sigma \in \mathcal{M}_Y$ can be matched to σ at any position in X without restrictions. After all σ's in $\mathcal{M}_Y$ are matched, the number of remaining unmatched σ's in X is $occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma)$, which can be seen as the occurrence constraint $C_{occ}(\sigma)$ of the input (X, Y, C_{occ}) for RBLCS. In the following, we show that (i) from the input $(X, Y, \mathcal{M}_Y)$ for 1FLCS, we can construct the input (X, Y, C_{occ}) for RBLCS such that $C_{occ}(\sigma) = occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma)$ for every $\sigma \in \Sigma$ in polynomial time, and vice versa; (ii) from an optimal solution of the former problem, we can construct an optimal solution of the latter problem in polynomial time, and vice versa.

Consider an input triple $(X, Y, \mathcal{M}_Y)$ of 1FLCS and a feasible solution Z_{1F}. Then, for every symbol σ, $occ(Z_{1F}, \sigma) \leq occ(X, \sigma)$ holds. If $occ(X, \sigma) < occ(\mathcal{M}_Y, \sigma)$, then $occ(\mathcal{M}_Y, \sigma) - occ(X, \sigma)$ σ's in $\mathcal{M}_Y$ are clearly redundant. If the input $(X, Y, \mathcal{M}_Y)$ of 1FLCS satisfies $occ(X, \sigma) \geq occ(\mathcal{M}_Y, \sigma)$ for every $\sigma \in \Sigma$, then we call $(X, Y, \mathcal{M}_Y)$ the *standard* input. Without loss of generality, we assume that every input of 1FLCS is standard.

Lemma 4. *Suppose that a triple $(X, Y, \mathcal{M}_Y)$ is a standard input for 1FLCS, Y^* is an optimal filling, and Z is the longest common subsequence of X and Y^*. Then, for every σ in Σ, $occ(Z, \sigma) \geq occ(\mathcal{M}_Y, \sigma)$ is satisfied.*

Proof. Let $X = \langle x_1, \ldots, x_n\rangle$, $Y^* = \langle y^*_1, \ldots, y^*_m\rangle$, and $Z = \langle z_1, \ldots, z_\ell\rangle = \langle x_{i_1}, \ldots, x_{i_\ell}\rangle = \langle y^*_{j_1}, \ldots, y^*_{j_\ell}\rangle$. Since the input is standard, for every σ, $occ(\mathcal{M}_Y, \sigma) \leq occ(X, \sigma)$ holds.

Now suppose for the purpose of obtaining a contradiction that there exists at least one symbol, say, σ', $occ(Z,\sigma') < occ(\mathcal{M}_Y,\sigma') \le occ(X,\sigma')$ holds. Since $occ(Z,\sigma') < occ(X,\sigma')$ holds, we can find an index q such that the qth symbol x_q in X is σ' but q is not in $I_X = \langle i_1, i_2, \ldots, i_\ell\rangle$. First, we assume that $i_p < q < i_{p+1}$ holds for some p where $1 \le p \le \ell - 1$. Then, we construct a new sequence $Z' = \langle x_{i_1}, \ldots, x_{i_p}\rangle \oplus \langle\sigma'\rangle \oplus \langle x_{i_{p+1}} \ldots, x_{i_\ell}\rangle$ of length $\ell + 1$. If $q < i_1$ ($i_\ell < q$, resp.), then we insert σ' to the head position, i.e., $Z' = \langle\sigma'\rangle \oplus \langle x_{i_1}, \ldots, x_{i_\ell}\rangle$ (to the tail position, i.e., $Z' = \langle x_{i_1}, \ldots, x_{i_\ell}\rangle \oplus \langle\sigma'\rangle$, resp.). Moreover, since $occ(Z,\sigma') < occ(\mathcal{M}_Y,\sigma')$, we can find an index q' such that the q'th symbol $y_{q'}$ inserted into Y^* is σ' but q' is not in $I_{Y^*} = \langle j_1, j_2, \ldots, j_\ell\rangle$. Then we construct a new filling Y^{**} as follows: (1) First remove the q'th symbol $y_{q'}$ ($= \sigma'$) from Y^*, and then (2) insert $y_{q'}$ right after y_{j_p} of Y^*. Note that the $(p+1)$st symbol in the new sequence Z' is σ'. It follows that $LCS(X, Y^{**}) = Z'$ and thus we can obtain the sequence of length $\ell + 1$ from $(X, Y, \mathcal{M}_Y)$, which is a contradiction. Therefore, for all σ in Σ, $occ(Z,\sigma) \ge occ(\mathcal{M}_Y,\sigma)$ holds. □

Consider an input triple $(X, Y, \mathcal{M}_Y)$ of 1FLCS and its optimal solution Z_{1F}. Suppose that there is a symbol σ such that $occ(X,\sigma) > occ(Y,\sigma) + occ(\mathcal{M}_Y,\sigma)$. Let $\ell = occ(X,\sigma) - (occ(Y,\sigma) + occ(\mathcal{M}_Y,\sigma)) \ge 0$. Then, at least ℓ σ's in X do not appear in Z_{1F}. Let $\mathcal{S}_\sigma$ be a multiset of ℓ σ's. Now, suppose that for a new triple $(X, Y, \mathcal{M}_Y \cup \mathcal{S}_\sigma)$, we can obtain an optimal solution Z. Then, the length of Z must be equal to $|Z_{1F}| + \ell$. Moreover, by removing ℓ σ's in $\mathcal{S}_\sigma$ from Z, we can easily find the original optimal solution Z_{1F} for $(X, Y, \mathcal{M}_Y)$. For every symbol σ' in Σ satisfying $occ(X,\sigma') > occ(Y,\sigma') + occ(\mathcal{M}_Y,\sigma')$, the similar discussion as the above can be applied. Let $\mathcal{S} = \bigcup_{\sigma': occ(X,\sigma') > occ(Y,\sigma') + occ(\mathcal{M}_Y,\sigma')} \mathcal{S}_{\sigma'}$. If we are given the triple $(X, Y, \mathcal{M}_Y \cup \mathcal{S})$, then by finding its optimal solution Z' first, and then removing all the symbols in $\mathcal{S}$ from Z', we obtain Z_{1F}. We call the triple $(X, Y, \mathcal{M}_Y \cup \mathcal{S})$ obtained by merging $\mathcal{S}$ with $\mathcal{M}_Y$ an *extended triple*. If the extended triple $(X, Y, \mathcal{M}_Y)$ of 1FLCS satisfies $occ(X,\sigma) \ge occ(\mathcal{M}_Y,\sigma)$ for every $\sigma \in \Sigma$ then it is called *ex-standard*. If $occ(X,\sigma) < occ(\mathcal{M}_Y,\sigma)$ holds for a symbol $\sigma \in \Sigma$, $(occ(\mathcal{M}_Y,\sigma) - occ(X,\sigma))$ σ's in X can easily be matched as observed in Lemma 2. Therefore, to simplify the discussion, we assume that every input triple $(X, Y, \mathcal{M}_Y)$ of 1FLCS is always ex-standard.

The following lemma is quite trivial but plays an important role:

Lemma 5. *(1) Suppose that an input triple $(X, Y, \mathcal{M}_Y)$ for 1FLCS is ex-standard. Then, we can construct a standard input triple (X, Y, C_{occ}) for RBLCS satisfying $C_{occ}(\sigma) = occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma)$ for every $\sigma \in \Sigma$ in polynomial time. (2) Suppose that an input triple (X, Y, C_{occ}) for RBLCS is standard. Then, we can construct an ex-standard input triple $(X, Y, \mathcal{M}_Y)$ for 1FLCS satisfying $occ(\mathcal{M}_Y,\sigma) = occ(X,\sigma) - C_{occ}(\sigma)$ for every $\sigma \in \Sigma$ in polynomial time.*

Proof. (1) Since the (ex-standard) triple $(X, Y, \mathcal{M}_Y)$ is standard, $occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma) \ge 0$ for every σ. Therefore, we can always obtain the valid occurrence constraint such that $C_{occ}(\sigma) = occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma)$ for every σ. Furthermore, since the triple $(X, Y, \mathcal{M}_Y)$ is extended, $C_{occ}(\sigma) = occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma) \le occ(Y,\sigma)$. It follows that $C_{occ}(\sigma) \le \min\{occ(X,\sigma), occ(Y,\sigma)\}$. Hence, the triple (X, Y, C_{occ}) must be standard for RBLCS.

(2) Since the triple (X, Y, C_{occ}) is standard, $C_{occ}(\sigma) \le \min\{occ(X,\sigma), occ(Y,\sigma)\}$. Therefore, we can always obtain the valid multiset $\mathcal{M}_Y$ such that $occ(\mathcal{M}_Y,\sigma) = occ(X,\sigma) - C_{occ}(\sigma) \ge 0$ for every σ.

Both of the above constructions can be executed in polynomial time by scanning X, Y, and $\mathcal{M}$. □

Lemma 6. *Consider an ex-standard input $(X, Y, \mathcal{M}_Y)$ for 1FLCS and a standard input (X, Y, C_{occ}) for RBLCS such that $C_{occ}(\sigma) = occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma)$ holds for every $\sigma \in \Sigma$. Let $Z_F = LCS(X, Y, \mathcal{M}_Y)$ and Y^* be an optimal filling for 1FLCS. Also, let $Z_R = LCS(X, Y, C_{occ})$ be an optimal solution for RBLCS. Then, $|Z_R| + |\mathcal{M}_Y| = |Z_F|$ holds.*

Proof. First, from Lemma 5, we always find a pair of triples $(X, Y, \mathcal{M}_Y)$ and (X, Y, C_{occ}) such that the former and the latter are the ex-standard input for 1FLCS and the standard input for RBLCS satisfying $C_{occ}(\sigma) = occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma)$ for every $\sigma \in \Sigma$, respectively.

(1) We first show that $|Z_F| \le |Z_R| + |\mathcal{M}_Y|$ holds. Let $X = \langle x_1, \ldots, x_n\rangle$, $Y = \langle y_1, \ldots, y_m\rangle$, and $M_Y = \langle\psi_1, \ldots, \psi_\ell\rangle$, where M_Y is the sequence-expression of $\mathcal{M}_Y$. By the assumption that $(X, Y, \mathcal{M}_Y)$ is ex-standard, there exists a sequence $\langle i_1, i_2, \ldots, i_\ell\rangle$ of indices of X satisfying $L(X, Y, \mathcal{M}_Y) = L(X \setminus \langle i_1, \ldots, i_\ell\rangle, Y, \emptyset) + \ell$, by regarding $L(X, Y, \emptyset, \mathcal{M}_Y)$ as $L(X, Y, \mathcal{M}_Y)$, and by using the formula in Lemma 2 recursively. Since $\mathcal{M}_Y = \emptyset$, $L(X \setminus \langle i_1, \ldots, i_\ell\rangle, Y, \emptyset)$ is clearly equal to the length of the longest common subsequence Z' of $X \setminus \langle i_1, \ldots, i_\ell\rangle$ and Y. Therefore, $|Z_F| = |Z'| + |\mathcal{M}_Y|$. Note that Z' is a common subsequence of the original X and Y and satisfies the following for every σ:

$$occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma) = occ(X \setminus \langle i_1, \ldots, i_\ell\rangle, \sigma) \ge occ(Z',\sigma).$$

That is, every symbol in Z' satisfies the occurrence constraint $C_{occ} = occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma)$ of RBLCS, which implies that $|Z'| \le |Z_R|$. As a result, $|Z_F| = |Z'| + |\mathcal{M}_Y| \le |Z_R| + |\mathcal{M}_Y|$ holds.

(2) Next, we show that $|Z_R| + |\mathcal{M}_Y| \le |Z_F|$. Recall that for every σ, $occ(Z_R,\sigma) \le C_{occ}(\sigma) = occ(X,\sigma) - occ(\mathcal{M}_Y,\sigma)$ is satisfied. Here, from the viewpoint of 1FLCS, we can obtain a longer sequence than Z_R by filling symbols of $\mathcal{M}_Y$ into Y. Suppose that Z_R is a common subsequence for RBLCS on (X, Y, C_{occ}) and $(X, Y, \emptyset)$ is an input triple for 1FLCS. From Corollary 2, by setting a multiset $\mathcal{M}'_Y = \{\sigma\}$ and filling σ into Y as matched with some σ in X, we can obtain a common subsequence Z_1 such that $|Z_1| = |Z_R| + 1$, $occ(Z_1,\sigma) = occ(Z_R,\sigma) + 1$, and $occ(Z_1,\sigma') = occ(Z_R,\sigma')$ for every σ' except for σ. By repeating the merge $\mathcal{M}'_Y \cup \{\sigma\}$ and the filling of σ $occ(\mathcal{M}_Y,\sigma)$-times for every $\sigma \in \Sigma$, we can eventually obtain $\mathcal{M}_Y$, the filling of Y and $\mathcal{M}_Y$, and a common subsequence Z satisfying $|Z| = |Z_R| + \sum_{\sigma\in\Sigma} occ(\mathcal{M}_Y,\sigma) = |Z_R| + |\mathcal{M}_Y|$. Since Z_F is the longest, $|Z| \le |Z_F|$. Hence, $|Z_R| + |\mathcal{M}_Y| = |Z| \le |Z_F|$ holds.

From (1) and (2), $|Z_R| + |\mathcal{M}_Y| = |Z_F|$. This completes the proof. □

Theorem 2. *Consider an ex-standard input $(X, Y, \mathcal{M}_Y)$ for 1FLCS and a standard input (X, Y, C_{occ}) for RBLCS such that $C_{occ}(\sigma) = occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma)$ holds for every $\sigma \in \Sigma$. Let $Z_F = LCS(X, Y, \mathcal{M}_Y)$ and Y^* be an optimal filling for 1FLCS. Also, let $Z_R = LCS(X, Y, C_{occ})$ be an optimal solution for RBLCS. Then, the followings hold: (1) Given an optimal solution Z_R for RBLCS, we can obtain an optimal solution for* 1FLCS *in polynomial time. (2) Given an optimal filling Y^* for 1FLCS, we can obtain an optimal solution for* RBLCS *in polynomial time.*

Proof. Consider two sequences X and Y, a multiset $\mathcal{M}_Y$, and an occurrence constraint C_{occ} such that $C_{occ}(\sigma) = occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma)$ holds for every $\sigma \in \Sigma$.

(1) Suppose that the optimal solution Z_R for RBLCS is now given. From Lemma 6, every optimal solution for 1FLCS is of length $|Z_R| + |\mathcal{M}_Y|$. Hence, it is enough to prove that we can obtain an optimal filling Y^* of Y and $\mathcal{M}_Y$ from Z_R and a common subsequence Z_F of X and Y^* such that $|Z_R| + |\mathcal{M}_Y| = |Z_F|$ in polynomial time. As seen in the proof of Lemma 6, by repeating the merge $\mathcal{M}'_Y = \mathcal{M}'_Y \cup \{\sigma\}$ and the filling of σ $occ(\mathcal{M}_Y, \sigma)$-times for every $\sigma \in \Sigma$, we eventually obtain Y^* and Z_F satisfying $|Z_F| = |Z_R| + \sum_{\sigma \in \Sigma} occ(\mathcal{M}_Y, \sigma) = |Z_R| + |\mathcal{M}_Y|$. The total number of iterations is $|\mathcal{M}_Y|$. Since each iteration of the procedure in part (2) of the proof of Lemma 6 can be done in polynomial time, Y^* and Z_F of 1FLCS can be obtained in polynomial time.

(2) Suppose that the optimal filling Y^* is now given. The longest common subsequence Z_F of X and Y^*, and its index-expression (I_X, I_{Y^*}) can be obtained in polynomial time. From Lemma 4, $occ(Z_F, \sigma) \geq occ(\mathcal{M}_Y, \sigma)$ holds for every $\sigma \in \Sigma$. Therefore, we can find $|Z_F| - |\mathcal{M}_Y|$ XY-matches in (I_X, I_{Y^*}). Letting z_ℓ be the symbol of the ℓth XY-match $(1 \leq \ell \leq |Z_F| - |\mathcal{M}_Y|)$, we construct the sequence $Z_F^- = \langle z_1, z_2, \ldots, z_{|Z_F|-|\mathcal{M}_Y|} \rangle$ of length $|Z_F| - |\mathcal{M}_Y|$. Note that Z_F^- must be a common subsequence of X and Y. Moreover, Z_F^- satisfies the occurrence constraint $C_{occ}(\sigma) = occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma) \geq occ(Z_F, \sigma) - occ(\mathcal{M}_Y, \sigma)$ for every $\sigma \in \Sigma$. Since $|Z_F^-| = |Z_F| - |\mathcal{M}_Y|$, Z_F^- is an optimal solution for RBLCS from Lemma 6. The construction of Z_F^- can be easily executed by scanning the index-expression (I_X, I_{Y^*}) and thus it can be done in polynomial time. □

4.3. RBLCS and 2FLCS

In this subsection we consider the polynomial-time equivalence between 2FLCS and RBLCS. Since 1FLCS on $(X, Y, \mathcal{M}_Y)$ is equivalent to 2FLCS on $(X, Y, \emptyset, \mathcal{M}_Y)$, 1FLCS can be solved by using any algorithm for 2FLCS. From the polynomial-time equivalence between 1FLCS and RBLCS in the previous subsection, RBLCS can also be solved by the same algorithm with some extra polynomial-time calculations. Therefore, to establish the equivalence between RBLCS and 2FLCS, only one direction remains to be proved. To do so, we first give a pair of two inputs $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ for 2FLCS and (X, Y, C_{occ}) for RBLCS. Then, we show that given an optimal solution Z_R of RBLCS on (X, Y, C_{occ}), we can obtain optimal fillings X^* and Y^* of 2FLCS on $(X, Y, \mathcal{M}_X \mathcal{M}_Y)$ in polynomial time.

Lemma 7. *Suppose that an input $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS satisfies $occ(X, \sigma) = p < occ(\mathcal{M}_Y, \sigma) = q$ and $\min\{occ(\mathcal{M}_X, \sigma), occ(Y, \sigma) + q - p\} = \lambda \geq 0$ for some positive integers p and q. Then the following holds:*

$$L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) \leq L(X, Y, \mathcal{M}_X \setminus \{\sigma^*\}, \mathcal{M}_Y \setminus \{\sigma^{q-p}\}) + \lambda$$

Proof. Suppose that an input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS satisfies $occ(X, \sigma) = p < occ(\mathcal{M}_Y, \sigma) = q$. If we set $X = \langle x_1, \ldots, x_n \rangle$ and $\mathcal{M}_Y = \langle \psi_1, \ldots, \psi_\ell \rangle$, and apply the formula in Lemma 2 recursively, then there exist two sequences of $\langle i_1, \ldots, i_p \rangle$ and $\langle j_1, \ldots, j_p \rangle$ of indices such that $\sigma = x_{i_r} = \psi_{j_r}$ for every $1 \leq r \leq p$. Therefore, we obtain

$$L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) = L(X \setminus \langle i_1, \ldots, i_p \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j_1, \ldots, j_p \rangle) + p.$$

Suppose that X^+ and Y^+ are optimal fillings of $(X \setminus \langle i_1, \ldots, i_p \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j_1, \ldots, j_p \rangle)$. Then, $occ(X^+, \sigma) \leq occ(\mathcal{M}_X, \sigma)$ since $\sigma \notin X \setminus \langle i_1, \ldots, i_p \rangle$ and $occ(Y^+, \sigma) \leq occ(Y, \sigma) + q - p$. Therefore, we obtain

$$occ(Z, \sigma) \leq \min\{occ(\mathcal{M}_X, \sigma), occ(Y, \sigma) + q - p\}$$

for $Z = LCS(X^+, Y^+)$. Now, we set $\min\{occ(\mathcal{M}_X, \sigma), occ(Y, \sigma) + q - p\} = \lambda$. Then, we have:

$$L(X \setminus \langle i_1, \ldots, i_p \rangle, Y, \mathcal{M}_X \setminus \{\sigma^*\}, \mathcal{M}_Y \setminus \{\sigma^*\}) + \lambda \geq L(X \setminus \langle i_1, \ldots, i_p \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j_1, \ldots, j_p \rangle).$$

As defined above, the sequence $\langle j_1, \ldots, j_p \rangle$ of indices satisfies $\sigma = \psi_{j_r}$ for $1 \leq r \leq p$. Suppose also that $\langle j_{p+1}, \ldots, j_q \rangle$ satisfies $\psi_{j_{r'}} = \sigma$ for every $p + 1 \leq r' \leq q$. Then, we obtain:

$$\begin{aligned} L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) &= L(X \setminus \langle i_1, \ldots, i_p \rangle, Y, \mathcal{M}_X, \mathcal{M}_Y \setminus \langle j_1, \ldots, j_p \rangle) + p \\ &\leq L(X \setminus \langle i_1, \ldots, i_p \rangle, Y, \mathcal{M}_X \setminus \{\sigma^*\}, \mathcal{M}_Y \setminus \{\sigma^*\}) + \lambda + p \\ &= L(X, Y, \mathcal{M}_X \setminus \{\sigma^*\}, \mathcal{M}_Y \setminus \langle j_{p+1}, \ldots, j_q \rangle) + \lambda. \end{aligned}$$

This completes the proof. □

Theorem 3. *Suppose that an input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS satisfies $occ(X, \sigma) = p < occ(\mathcal{M}_Y, \sigma) = q$ for some positive integers p and q, and optimal fillings X_1^+ and Y_1^+ of $(X, Y, \mathcal{M}_X \setminus \{\sigma^*\}, \mathcal{M}_Y \setminus \{\sigma^{q-p}\})$ are given. Then, optimal fillings X_2^+ and Y_2^+ of an input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ can be obtained in polynomial time.*

Proof. Suppose that Z_1 is the longest common subsequence of X_1^+ and Y_1^+ such that the index-expression of Z_1 is (I, J), where $I = \langle i_1, \ldots, i_k \rangle$ and $J = \langle j_1, \ldots, j_k \rangle$. Also suppose that Z_2 is the longest common subsequence of X_2^+ and Y_2^+. From Lemma 7, for $\lambda = \min\{occ(\mathcal{M}_X, \sigma), occ(Y, \sigma) + q - p\}$, $|Z_1| + \lambda \geq |Z_2|$ holds.

Now suppose that $X_1^+ = \langle x_1, \ldots, x_n \rangle$ and $Y_1^+ = \langle y_1, \ldots, y_m \rangle$. Also suppose that $Y_2^+ = Y_1^+ \oplus \langle \overbrace{\sigma, \ldots, \sigma}^{q-p} \rangle$. One can see that $occ(Y_2^+, \sigma) = occ(Y, \sigma) + q \geq q$. Therefore, $occ(Z_1, \sigma) \leq p < occ(\mathcal{M}_Y, \sigma) = q \leq occ(Y_2^+, \sigma)$ and Z_1 is a common subsequence of X_1^+ and Y_2^+. We can get the target sequence X_2^+ in polynomial time by applying Lemma 3 $\min\{occ(\mathcal{M}_X, \sigma), occ(Y, \sigma) + q - p\}$-times. □

It is important to note that $(X, Y, \mathcal{M}_X \setminus \{\sigma^*\}, \mathcal{M}_Y \setminus \{\sigma^{q-p}\})$ does not satisfy both $occ(X, \sigma) < occ(\mathcal{M}_Y, \sigma)$ and $occ(Y, \sigma) < occ(\mathcal{M}_X, \sigma)$. For Y and $\mathcal{M}_X$, we have:

Corollary 3. *Suppose that an input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS satisfies $occ(Y, \sigma) = p < occ(\mathcal{M}_X, \sigma) = q$ for some positive integers p and q, and optimal fillings X_1^+ and Y_1^+ of $(X, Y, \mathcal{M}_X \setminus \{\sigma^{q-p}\}, \mathcal{M}_Y \setminus \{\sigma^*\})$ are given. Then, optimal fillings X_2^+ and Y_2^+ of an input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ can be obtained in polynomial time.*

From Theorem 3 and Corollary 3, any input of 2FLCS can be reduced in polynomial time to the quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ such that for every σ, both $occ(X, \sigma) \geq occ(\mathcal{M}_Y, \sigma)$ and $occ(Y, \sigma) \geq occ(\mathcal{M}_X, \sigma)$ are satisfied. Therefore, if the input $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS satisfies both $occ(X, \sigma) \geq occ(\mathcal{M}_Y, \sigma)$ and $occ(Y, \sigma) \geq occ(\mathcal{M}_X, \sigma)$ for every $\sigma \in \Sigma$, then we call $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ the *standard* input.

Theorem 4. *For a standard input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS, consider an occurrence constraint C_{occ} such that for every $\sigma \in \Sigma$, $C_{occ}(\sigma) = \min\{occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma), occ(Y, \sigma) - occ(\mathcal{M}_X, \sigma)\}$ holds. Then, the triple (X, Y, C_{occ}) must be standard for RBLCS. If an optimal solution Z_R of RBLCS on (X, Y, C_{occ}) is given, then we can obtain optimal fillings X^* and Y^* of 2FLCS on a standard input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ in polynomial time.*

Proof. Suppose that the input $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS is standard, $|\mathcal{M}_X| = p$, and $|\mathcal{M}_Y| = q$. Let $X = \langle x_1, \ldots, x_n \rangle$ and $Y = \langle y_1, \ldots, y_m \rangle$. Then, by applying the arguments of Lemma 2 and Corollary 1 to all the symbols recursively, we can obtain the sequences $\langle i_1, \ldots, i_q \rangle$ and $\langle j_1, \ldots, j_p \rangle$ of different indices that satisfy the following:

$$L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) = L(X \setminus \langle i_1, \ldots, i_q \rangle, Y \setminus \langle j_1, \ldots, j_p \rangle, \emptyset, \emptyset) + p + q.$$

One can verify that for the input $(X \setminus \langle i_1, \ldots, i_q \rangle, Y \setminus \langle j_1, \ldots, j_p \rangle, \emptyset, \emptyset)$ of 2FLCS, the longest common subsequence of $X \setminus \langle i_1, \ldots, i_q \rangle$ and $Y \setminus \langle j_1, \ldots, j_p \rangle$ is clearly an optimal solution of the classical LCS. Let Z' be such a sequence. Here, note that for every $\sigma \in \Sigma$, we can obtain:

$$occ(X \setminus \langle i_1, \ldots, i_q \rangle, \sigma) = occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma), \text{ and}$$
$$occ(Y \setminus \langle j_1, \ldots, j_p \rangle, \sigma) = occ(Y, \sigma) - occ(\mathcal{M}_X, \sigma).$$

Therefore, we have:

$$occ(Z', \sigma) \leq \min\{occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma), occ(Y, \sigma) - occ(\mathcal{M}_X, \sigma)\}.$$

Since Z' is a common subsequence of X and Y, Z' is a feasible solution of RBLCS on (X, Y, C_{occ}). Therefore, $|Z_R| \geq |Z'|$ holds. It follows that $|Z_R| + |\mathcal{M}_X| + |\mathcal{M}_Y| \geq |Z'| + |\mathcal{M}_X| + |\mathcal{M}_Y| = L(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$.

As for Z_R,

$$\begin{aligned} occ(Z_R, \sigma) &\leq C_{occ}(\sigma) \\ &= \min\{occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma), occ(Y, \sigma) - occ(\mathcal{M}_X, \sigma)\} \end{aligned}$$

holds for every σ. Therefore, by applying Lemma 3 $occ(\mathcal{M}_X, \sigma)$-times to X for every symbol $\sigma \in \Sigma$, we can construct in polynomial time the filling X^+ of X and $\mathcal{M}_X$, and a common subsequence Z_1 of X^+ and Y such that $|Z_1| = |Z_R| + |\mathcal{M}_X|$ and $occ(Z_1, \sigma) \leq C_{occ}(\sigma) + |\mathcal{M}_X|$.

Note that for every σ, $occ(X^+, \sigma) = occ(X, \sigma) + occ(\mathcal{M}_X, \sigma)$ and $occ(Z_1, \sigma) \leq occ(X, \sigma) - occ(\mathcal{M}_Y, \sigma) + occ(\mathcal{M}_X, \sigma)$ hold. Hence, by applying Corollary 2 $occ(\mathcal{M}_Y, \sigma)$-times to Y for every symbol σ, we can construct in polynomial time the filling Y^+ of Y and $\mathcal{M}_Y$, and a common subsequence Z_2 of X^+ and Y^+ such that $|Z_2| = |Z_1| + |\mathcal{M}_Y| = |Z_R| + |\mathcal{M}_X| + |\mathcal{M}_Y|$. Recall that $|Z_R| + |\mathcal{M}_X| + |\mathcal{M}_Y| \geq L(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$. Therefore, $|Z_2| \geq L(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ holds.

As a result, X^+ and Y^+ are optimal fillings of 2FLCS on $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ and those can be obtained in polynomial time if Z_R is given. This completes the proof. □

5. $O(1.41422^n)$-Time exact algorithm for RBLCS

5.1. Overview

Let us first explain the overview of the strategy to design a new DP-based algorithm. In [2], a dynamic programming (DP) based algorithm for RBLCS was provided whose running time is $O(1.44255^n)$. We improve the running time from $O(1.44255^n)$ to $O(1.41422^n)$.

Now, consider the original LCS and its typical DP-based algorithm. Let $L(i,j)$ denote the length of a longest common subsequence of the ith prefix $X_{1..i}$ of X and the jth prefix $Y_{1..j}$ of Y. In the process of execution, each value of $L(i,j)$ is computed and is stored into a two-dimensional DP-table of size $(n+1)\times(m+1)$. For more details, see, e.g., [8].

For RBLCS, the previous DP-based algorithm proposed in [2] has to store not only the length of a subsequence, say, Z, but also the occurrence $occ(Z,\sigma)$ of every σ in Z not to break the occurrence constraint $C_{occ}(\sigma)$. To store the occurrences, the algorithm introduces an occurrence vector $\boldsymbol{v}$. Let $L(i,j,\boldsymbol{v})$ be the length of a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j}$ satisfying the occurrence vector $\boldsymbol{v}$, i.e., the length of the subsequence which does not break the occurrence constraint. Then, each value of $L(i,j,\boldsymbol{v})$ is stored into a three-dimensional DP-table of size $(n+1)\times(m+1)\times\prod_\sigma(C_{occ}(\sigma)+1)$, where $C_{occ}(\sigma)+1$ is the number of candidates $0,1,\ldots,C_{occ}(\sigma)$ of occurrences of a symbol σ. In [2], the authors showed that the size of this three-dimensional DP-table is bounded from above by $O(1.44255^n)$.

Our new DP-based algorithm prepares another three-dimensional DP-table whose size is smaller than the one in [2]. To this end, using a concise form of the occurrence vector $\boldsymbol{v}$, we narrow down the range of values that each component of $\boldsymbol{v}$ takes. More specifically, we switch values to store with respect to symbol σ: if $C_{occ}(\sigma)\le occ(X,\sigma)/2$, then we store how many σ's can still be used, or otherwise, we store how many σ's must be deleted hereafter. This strategy narrows down the range of values to store from $0,1,\ldots,C_{occ}(\sigma)$ to $0,1,\ldots,\min\{C_{occ}(\sigma),occ(X,\sigma)-C_{occ}(\sigma)\}$, which reduces the size of DP-table to $(n+1)\times(m+1)\times\prod_\sigma(\min\{C_{occ}(\sigma),occ(X,\sigma)-C_{occ}(\sigma)\}+1)$. We will show that this new DP-table has size $O(1.41422^n)$, and hence the next theorem is obtained (its proof is given in Section 5.3).

Theorem 5. *There is an $O(1.41422^n)$-time DP-based algorithm to solve RBLCS for two input sequences X and Y where $|X|=n$, $|Y|=poly(n)$, and $|X|\le|Y|$.*

Recall that all reductions in the previous sections preserve X and Y. By our polynomial-time equivalences, we obtain the following corollary.

Corollary 4. *MRCS, 1FLCS, and 2FLCS for two input sequences X and Y can be solved in $O(1.41422^n)$ time, where $|X|=n$, $|Y|=poly(n)$, and $|X|\le|Y|$*

Proof. By Theorem 5, RBLCS can be solved in $O(1.41422^n)$ time. By the polynomial-time equivalence between MRCS (1FLCS, or 2FLCS) and RBLCS in Theorem 1 (2 or 4), MRCS (1FLCS, or 2FLCS) can be solved in $O(1.41422^n+poly(n))=O(1.41422^n)$ time, since reductions between these problems take $O(poly(n))$ time. □

5.2. Previous DP-based algorithm

In this subsection we review the DP-based algorithm proposed in [2]. Note that if $C_{occ}(\sigma)\le occ(X,\sigma)$ for a symbol σ, then we do not need to worry about the occurrences of σ in a common subsequence. Thus, let $\Sigma_{>C_{occ}}=\{\sigma_i \mid occ(X,\sigma_i)>C_{occ}(\sigma_i)\}$, a set of symbols such that each σ_i occurs at least $C_{occ}(\sigma_i)+1$ times in X. Now suppose that $|\Sigma_{>C_{occ}}|=\ell$ and, without loss of generality, $\Sigma_{>C_{occ}}=\{\sigma_1,\sigma_2,\ldots,\sigma_\ell\}$. Then, we prepare an occurrence vector of length ℓ, denoted by $\boldsymbol{v}=(v_1,v_2,\ldots,v_\ell)$. The pth component v_p of $\boldsymbol{v}$ represents how many σ_p's appear in a (temporal) repetition-bounded common subsequence for $1\le p\le\ell$ and $v_p\in\{0,1,\ldots,C_{occ}(\sigma_p)\}$. Maintaining $\boldsymbol{v}$, not to break the occurrence constraint, the algorithm solves a subproblem of finding a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j}$. For the occurrence vector $\boldsymbol{v}=(v_1,v_2,\ldots,v_{p-1},v_p,v_{p+1}\ldots,v_\ell)$, we define a new vector $\boldsymbol{v}|_{p=q}=(v_1,v_2,\ldots,v_{p-1},q,v_{p+1},\ldots,v_\ell)$. Note that if $v_p=q$ in the original occurrence vector $\boldsymbol{v}$, then $\boldsymbol{v}|_{p=q}=\boldsymbol{v}$.

First, the algorithm is based on the optimal substructure of a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j}$ [2].

Theorem 6 ([2])**.** *There is a repetition-bounded longest common subsequence $Z_{1..h}=\langle z_1,z_2,\ldots,z_h\rangle$ of $X_{1..i}$ and $Y_{1..j}$ satisfying an occurrence vector $\boldsymbol{v}$ and the following properties:*

(1) If $x_i=y_j=\sigma_p$ and $\sigma_p\notin\Sigma_{>C_{occ}}$, then $z_h=\sigma_p$ and $Z_{1..h-1}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{v}$.

(2) If $x_i=y_j=\sigma_p$, $\sigma_p\in\Sigma_{>C_{occ}}$, and $v_p>0$, then $z_h=\sigma_p$ implies that $Z_{1..h-1}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{v}|_{p=v_p-1}$.

(3) If $x_i=y_j=\sigma_p$, $\sigma_p\in\Sigma_{>C_{occ}}$, and $v_p=0$, then $z_h\ne\sigma_p$ and $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{v}$.

(4) If $x_i \neq y_j$, then

(a) $z_h \neq x_i$ implies that $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j}$ satisfying $\boldsymbol{v}$;
(b) $z_h \neq y_j$ implies that $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{v}$.

Then, the above optimal substructure gives the following recursive formula:

$$L(i,j,\boldsymbol{v}) = \begin{cases} 0 \\ \quad \text{if } i=0 \text{ or } j=0 \\ L(i-1,j-1,\boldsymbol{v})+1 \\ \quad \text{if } i,j>0,\ x_i=y_j=\sigma_p, \text{ and } \sigma_p \notin \Sigma_{>C_{occ}} \\ L(i-1,j-1,\boldsymbol{v}|_{p=v_p-1})+1 \\ \quad \text{if } i,j>0,\ x_i=y_j=\sigma_p,\ \sigma_p \in \Sigma_{>C_{occ}}, \text{ and } v_p>0 \\ L(i-1,j-1,\boldsymbol{v}) \\ \quad \text{if } i,j>0,\ x_i=y_j=\sigma_p,\ \sigma_p \in \Sigma_{>C_{occ}}, \text{ and } v_p=0 \\ \max\{L(i-1,j,\boldsymbol{v}), L(i,j-1,\boldsymbol{v})\} \\ \quad \text{otherwise, i.e., if } i,j>0, \text{ and } x_i \neq y_j, \end{cases}$$

where $L(i,j,\boldsymbol{v})$ is the length of the repetition-bounded longest common subsequence for $(X_{1..i}, Y_{1..j}, C'_{occ})$ when the following is satisfied:

$$C'_{occ}(\sigma_p) = \begin{cases} v_p & \sigma_p \in \Sigma_{>C_{occ}} \\ occ(X,\sigma_p) & \sigma_p \notin \Sigma_{>C_{occ}}. \end{cases}$$

Note that $\sigma_p \notin \Sigma_{>C_{occ}}$ appears at most $occ(X,\sigma_p)$-times in the solution and thus the necessary condition of $C'_{occ}(\sigma_p)$ is only $C'_{occ}(\sigma_p) \geq occ(X,\sigma_p)$.

As described in Proposition 1, the algorithm implementing the above recursive formula can solve RBLCS in $O(1.44225^n)$ time [2].

5.3. New DP-based algorithm

In this subsection, we propose a new DP-based algorithm. First, we group symbols in $\Sigma_{>C_{occ}}$ into two types in Section 5.3.1, and introduce a new occurrence vector $\boldsymbol{w}$ in Section 5.3.2. Section 5.3.3 describes how to initialize the table used by the proposed algorithm. Then, we rewrite Theorem 6 with regard to the new occurrence vector $\boldsymbol{w}$ in Sec 5.3.4. Section 5.3.5 gives recursive formulas on the table based on Section 5.3.4. Finally, we estimate the time complexity of the proposed algorithm in Section 5.3.6.

5.3.1. Type

We define $Type : \Sigma_{>C_{occ}} \to \{0,1\}$ that divides the alphabet into two types as follows:

$$Type(\sigma) = \begin{cases} 0 & C_{occ}(\sigma) \leq occ(X,\sigma)/2 \\ 1 & C_{occ}(\sigma) > occ(X,\sigma)/2. \end{cases} \tag{1}$$

That is, if the occurrence constraint on σ is tight/small, then $Type(\sigma)=0$. Otherwise, i.e., if the occurrence constraint on σ is loose/large, then $Type(\sigma)=1$. Then, if $Type(\sigma)=0$ (or 1) for a symbol σ, $C_{occ}(\sigma) \leq occ(X,\sigma) - C_{occ}(\sigma)$ (or $occ(X,\sigma) - C_{occ}(\sigma) < C_{occ}(\sigma)$).

5.3.2. Occurrence vector $\boldsymbol{w}$

For symbols in $\Sigma_{>C_{occ}}$, we introduce a new occurrence vector $\boldsymbol{w} = (w_1, \ldots, w_\ell)$ of length ℓ, assuming without loss of generality that $\Sigma_{>C_{occ}} = \{\sigma_1, \sigma_2, \ldots, \sigma_\ell\}$. In case of $Type(\sigma_p)=0$, w_p stores how many σ_p's we can still use in (the later part of) a common subsequence. On the other hand, if $Type(\sigma_p)=1$, then w_p stores how many σ_p's in (the later part of) X must be deleted. Thus, if $type(\sigma_p)=0$, $0 \leq w_p \leq C_{occ}(\sigma_p) (\leq occ(X,\sigma_p) - C_{occ}(\sigma_p))$, and otherwise (i.e., $Type(\sigma_p)=1$), $0 \leq w_p \leq occ(X,\sigma_p) - C_{occ}(\sigma_p) (< C_{occ}(\sigma_p))$. If there is an index p such that w_p of $\boldsymbol{w}$ is not in this range of values, we say that $\boldsymbol{w}$ is *invalid*. This switch of the values to store based on the type of a symbol reduces the number of candidates of $\boldsymbol{w}$ as seen below.

The occurrence vector $\boldsymbol{w}$ is initialized for each $1 \leq p \leq \ell$ as follows:

$$w_p = \begin{cases} C_{occ}(\sigma_p) & Type(\sigma_p)=0 \\ occ(X,\sigma_p) - C_{occ}(\sigma_p) & Type(\sigma_p)=1. \end{cases} \tag{2}$$

Example 4. Let $\Sigma = \{a,b,c\}$, $C_{occ}(a)=2$, $C_{occ}(b)=3$, and $C_{occ}(c)=4$. Consider two sequences $X = ccaabbabccbab$ and $Y = abcabcabcabcabc$, where $occ(X,a)=4$, $occ(X,b)=5$, and $occ(X,c)=4$. Then, $\Sigma_{>C_{occ}} = \{a,b\}$ (e.g., $\sigma_1 = a$ and

$\sigma_2 = b$), $Type(a) = 0$, and $Type(b) = 1$. By these, $\boldsymbol{w}$ is initialized as $\boldsymbol{w} = (2, 2)$, which means that the symbol a in X can be used at most twice in a solution and the symbol b must be deleted at least twice from X to obtain a solution. When we consider a temporal solution $Z' = abb$ of $X_{1..6} = ccaabb$ and $Y_{1..5} = abcab$, we set $\boldsymbol{w} = (1, 2)$, which means that a can be used at most once and b must be deleted at least twice later. In a repetition-bounded subsequence, a appears 0, 1, or 2 times, and b appears 0, 1, 2, or 3 times by respectively deleting 5, 4, 3, or 2 times (c appears 0 through 4 times). Hence, the original occurrence vector $\boldsymbol{v}$ has $3 \times 4 = 12$ candidates. A crucial point here is that one value 2 in the second component (corresponding to b) of $\boldsymbol{w}$ covers all the cases of deleting 2, 3, 4, and 5 occurrences of b. This reduces the number of candidates of $\boldsymbol{w}$ to $3 \times 3 = 9$ from 12 of $\boldsymbol{v}$.

5.3.3. Base cases

We prepare a three-dimensional table L. Let $LCS(i, j, \boldsymbol{w})$ represent a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j}$ satisfying the occurrence vector $\boldsymbol{w}$. Then, $L(i, j, \boldsymbol{w})$ stores the length of $LCS(i, j, \boldsymbol{w})$. We let $L(i, j, \boldsymbol{w}) = -\infty$ for $i < 0$, $j < 0$, or invalid $\boldsymbol{w}$'s.

To demonstrate how to define the base cases of the recurrences, we return to Example 4.

Example 5. As in Example 4, let $X = ccaabbabccbab$ and $Y = abcabcabcabcabc$. Only three occurrence vectors $(0, 2)$, $(1, 2)$, and $(2, 2)$ give repetition-bounded common subsequences; $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$, $(2, 0)$, or $(2, 1)$ does not give any feasible solution, since the number of deletion of b from X is at most 1 and hence they correspond to a sequence including at least 4 b's which breaks the condition $C_{occ}(b) = 3$. Also it is obvious that a repetition-bounded longest common subsequence of X and Y under the occurrence vector $(2, 2)$ is longer than that under $(0, 2)$ and $(1, 2)$. Thus, it is sufficient to start only with the occurrence vector $(2, 2)$.

To conclude the example, we would like the following relations to hold:

$$L(0, 0, (2, 2)) = 0$$
$$L(0, 0, (0, 0)) = L(0, 0, (0, 1)) = L(0, 0, (0, 2)) = -\infty$$
$$L(0, 0, (1, 0)) = L(0, 0, (1, 1)) = L(0, 0, (1, 2)) = -\infty$$
$$L(0, 0, (2, 0)) = L(0, 0, (2, 1)) = -\infty.$$

Following the idea in Example 5, we now define the base cases of the recurrence in general. First, for $i = j = 0$, we let:

$$L(0, 0, \boldsymbol{w}) = \begin{cases} 0 & \boldsymbol{w} \text{ satisfies (2).} \\ -\infty & \boldsymbol{w} \text{ does not satisfy (2).} \end{cases}$$

Since the occurrence vector $\boldsymbol{w}$ maintains the number of symbols from X, we then define the following relations for $i = 0$ and $j > 0$:

$$L(0, j, \boldsymbol{w}) = \begin{cases} 0 & j > 0 \text{ and } \boldsymbol{w} \text{ satisfies (2)} \\ -\infty & j > 0 \text{ and } \boldsymbol{w} \text{ does not satisfy (2)} \end{cases}$$

Next let us consider the case $j = 0$ and $i > 0$. For example, increasing i from 0 to 1 under the condition $j = 0$ corresponds to deleting x_1 from X, since a repetition-bounded longest common subsequence of $X_{1..1}$ and the empty string ($Y_{1..0}$) is also the empty string. If $x_i \notin \Sigma_{>C_{occ}}$, then $\boldsymbol{w}$ is not related to x_i. Thus,

$$L(i, 0, \boldsymbol{w}) = L(i - 1, 0, \boldsymbol{w}). \tag{3}$$

Consider the case that $x_i = \sigma_p \in \Sigma_{>C_{occ}}$ and $Type(\sigma_p) = 0$. Since $\boldsymbol{w}$ maintains how many σ_p's can be used later, w_p will not change and hence

$$L(i, 0, \boldsymbol{w}) = L(i - 1, 0, \boldsymbol{w}). \tag{4}$$

As the last case, suppose that $x_i = \sigma_p \in \Sigma_{>C_{occ}}$ and $Type(\sigma_p) = 1$. Since one σ_p in X is deleted, we basically decrease w_p by one. One exception is the case $w_p = 0$. Although we may have already deleted sufficient number of σ_p's so far, we delete σ_p further. In such a case, we keep $w_p = 0$. Therefore, if $w_p \geq 1$, then

$$L(i, 0, \boldsymbol{w}) = L(i - 1, 0, \boldsymbol{w}|_{p=w_p+1}),$$

and if $w_p = 0$, then

$$L(i, 0, \boldsymbol{w}) = \max\{L(i - 1, 0, \boldsymbol{w}|_{p=w_p+1}), L(i - 1, 0, \boldsymbol{w})\}.$$

In summary, we define the base cases of the recurrence by:

$$L(i,j,\boldsymbol{w}) = \begin{cases} 0 \\ \quad i=0,\ j\geq 0,\ \text{and } \boldsymbol{w} \text{ satisfies (2)} \\ -\infty \\ \quad i=0,\ j\geq 0,\ \text{and } \boldsymbol{w} \text{ does not satisfy (2)} \\ L(i-1,0,\boldsymbol{w}|_{p=w_p+1}) \\ \quad i\geq 1,\ j=0,\ x_i=\sigma_p\in\Sigma_{>C_{occ}},\ Type(\sigma_p)=1,\ \text{and } w_p\geq 1 \\ \max\{L(i-1,0,\boldsymbol{w}|_{p=w_p+1}), L(i-1,0,\boldsymbol{w})\} \\ \quad i\geq 1,\ j=0,\ x_i=\sigma_p\in\Sigma_{>C_{occ}},\ Type(\sigma_p)=1,\ \text{and } w_p=0 \\ L(i-1,0,\boldsymbol{w}) \\ \quad i\geq 1,\ j=0,\ \text{and } x_i\notin\Sigma_{>C_{occ}} \end{cases} \tag{5}$$

5.3.4. Optimal substructure

We rewrite the optimal substructure of a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j}$ satisfying an occurrence vector $\boldsymbol{v}$ in Theorem 6 with regard to the new occurrence vector $\boldsymbol{w}$. Basically, the optimal substructure is the same, and what we need to do is updating the occurrence vector $\boldsymbol{w}$ following its definition. We give three theorems. The first one is for a symbol not in $\Sigma_{>occ}$. Then, the second (or the third) theorem is for a symbol $\sigma\in\Sigma_{>occ}$ such that $Type(\sigma)=0$ (or $Type(\sigma)=1$).

Theorem 7. *Suppose that $x_i\notin\Sigma_{>C_{occ}}$. There is a repetition-bounded longest common subsequence $Z_{1..h}=\langle z_1,z_2,\ldots,z_h\rangle$ of $X_{1..i}$ and $Y_{1..j}$ satisfying an occurrence vector $\boldsymbol{w}$ and the following properties:*

1. *If $x_i=y_j$, then $z_h=x_i$ and $Z_{1..h-1}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{w}$*
2. *If $x_i\neq y_j$, then (a) $z_h\neq x_i$ implies that $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j}$ satisfying $\boldsymbol{w}$; and (b) $z_h\neq y_j$ implies that $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{w}$.*

Proof. Since $\boldsymbol{w}$ is independent of a symbol $\sigma_p\notin S_{>C_{occ}}$, the proofs of Theorem 6(1) and (4) can be also applied. □

Theorem 8. *Suppose that $x_i=\sigma_p\in\Sigma_{>C_{occ}}$ and $Type(\sigma_p)=0$. There is a repetition-bounded longest common subsequence $Z_{1..h}=\langle z_1,z_2,\ldots,z_h\rangle$ of $X_{1..i}$ and $Y_{1..j}$ satisfying an occurrence vector $\boldsymbol{w}$ and the following properties:*

1. *If $x_i=y_j$, then (a) $z_h=x_i$ implies that $Z_{1..h-1}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{w}|_{p=w_p+1}$; and (b) $z_h\neq x_i$ implies that $Z_{1..h-1}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{w}$.*
2. *If $x_i\neq y_j$, then (a) $z_h\neq x_i$ implies that $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j}$ satisfying $\boldsymbol{w}$; and (b) $z_h\neq y_j$ implies that $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{w}$.*

Proof. 1(a): The proof of Theorem 6(2) applies, except for the following point. Since we use $\sigma_p(=x_i)$ for z_h from X, we need to decrease w_p from the value for $X_{1..i-1}$ and $Y_{1..j-1}$.

1(b): The proof of Theorem 6(3) applies, since $\boldsymbol{w}$ does not change.

2(a) and 2(b): The proof of Theorem 6(4) applies, since $\boldsymbol{w}$ does not change. □

Theorem 9. *Suppose that $x_i=\sigma_p\in\Sigma_{>C_{occ}}$ and $Type(\sigma_p)=1$. There is a repetition-bounded longest common subsequence $Z_{1..h}=\langle z_1,z_2,\ldots,z_h\rangle$ of $X_{1..i}$ and $Y_{1..j}$ satisfying an occurrence vector $\boldsymbol{w}$ and the following properties:*

1. *If $x_i=y_j$, then (a) $z_h=x_i$ implies that $Z_{1..h-1}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{w}$; and (b) $z_h\neq x_i$ implies that $Z_{1..h-1}$ is an repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{w}|_{p=w_p+1}$ if $w_p\geq 1$, $\boldsymbol{w}|_{p=w_p+1}$ or $\boldsymbol{w}$ if $w_p=0$.*
2. *If $x_i\neq y_j$, then (a) $z_h\neq x_i$ implies that $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i-1}$ and $Y_{1..j}$ satisfying $\boldsymbol{w}|_{p=w_p+1}$ if $w_p\geq 1$, $\boldsymbol{w}|_{p=w_p+1}$ or $\boldsymbol{w}$ if $w_p=0$; and (b) $z_h\neq y_j$ implies that $Z_{1..h}$ is a repetition-bounded longest common subsequence of $X_{1..i}$ and $Y_{1..j-1}$ satisfying $\boldsymbol{w}$.*

Proof. 1(a): The proof of Theorem 6(2) applies, since $\boldsymbol{w}$ does not change.

1(b): The proof of Theorem 6(3) applies, except for the following point. Since we delete $\sigma_p(=x_i)$ for z_h from X, we need to decrease w_p from the value for $X_{1..i-1}$ and $Y_{1..j-1}$ if $w_p>0$. When $w_p=0$, σ_p might be deleted even after sufficient number of σ_p's are deleted, and hence $w_p=0$ may also hold for the subsequence $Z_{1..h-1}$.

2(a): The proof of Theorem 6(4) applies, and similarly to 1(b), we decrease w_p from that for $X_{1..i-1}$ and $Y_{1..j-1}$ if $w_p \geq 0$. Then, in case $w_p = 0$, $Z_{1..h-1}$ may satisfy $\boldsymbol{w}$.

2(b): The proof of Theorem 6(4) applies, since $\boldsymbol{w}$ does not change. □

5.3.5. Recursive formula

For $i \geq 1$ and $j \geq 1$, the recursive formula is obtained as follows, based on Theorems 7, 8, and 9.

$$L(i, j, \boldsymbol{w}) = \begin{cases} L(i-1, j-1, \boldsymbol{w}) + 1 \\ \quad x_i \notin \Sigma_{>C_{occ}} \text{ and } x_i = y_j \text{ (Theorem 7-1)} \\ \max\{L(i-1, j, \boldsymbol{w}),\ L(i, j-1, \boldsymbol{w})\} \\ \quad x_i \notin \Sigma_{>C_{occ}} \text{ and } x_i \neq y_j \text{ (Theorem 7-2)} \\ \max\{L(i-1, j-1, \boldsymbol{w}|_{p=w_p+1}) + 1,\ L(i-1, j-1, \boldsymbol{w})\} \\ \quad x_i = \sigma_p \in \Sigma_{>C_{occ}},\ Type(\sigma_p) = 0, \text{ and } x_i = y_j \text{ (Theorem 8-1)} \\ \max\{L(i-1, j, \boldsymbol{w}),\ L(i, j-1, \boldsymbol{w})\} \\ \quad x_i = \sigma_p \in \Sigma_{>C_{occ}},\ Type(\sigma_p) = 0, \text{ and } x_i \neq y_j \text{ (Theorem 8-2)} \\ \max\{L(i-1, j-1, \boldsymbol{w}) + 1,\ L(i-1, j-1, \boldsymbol{w}|_{p=w_p+1})\} \\ \quad x_i = \sigma_p \in \Sigma_{>C_{occ}},\ Type(\sigma_p) = 1,\ x_i = y_j, \text{ and } w_p \geq 1 \text{ (Theorem 9-1)} \\ \max\{L(i-1, j-1, \boldsymbol{w}) + 1,\ L(i-1, j-1, \boldsymbol{w}|_{p=w_p+1}),\ L(i-1, j-1, \boldsymbol{w})\} \\ \quad x_i = \sigma_p \in \Sigma_{>C_{occ}},\ Type(\sigma_p) = 1,\ x_i = y_j, \text{ and } w_p = 0 \text{ (Theorem 9-1)} \\ \max\{L(i-1, j, \boldsymbol{w}|_{p=w_p+1}),\ L(i, j-1, \boldsymbol{w})\} \\ \quad x_i = \sigma_p \in \Sigma_{>C_{occ}},\ Type(\sigma_p) = 1, x_i \neq y_j, \text{ and } w_p \geq 1 \text{ (Theorem 9-2)} \\ \max\{L(i-1, j, \boldsymbol{w}|_{p=w_p+1}),\ L(i-1, j, \boldsymbol{w}),\ L(i, j-1, \boldsymbol{w})\} \\ \quad x_i = \sigma_p \in \Sigma_{>C_{occ}},\ Type(\sigma_p) = 1, x_i \neq y_j, \text{ and } w_p = 0 \text{ (Theorem 9-2)} \end{cases} \tag{6}$$

Note that since $L(i-1, j-1, \boldsymbol{w}) + 1 \geq L(i-1, j-1, \boldsymbol{w})$ clearly holds, the fifth and the sixth formulas above for the case of Theorem 9-1 can be merged into just one formula

$$\max\{L(i-1, j-1, \boldsymbol{w}) + 1,\ L(i-1, j-1, \boldsymbol{w}|_{p=w_p+1})\}$$

for the case $x_i = \sigma_p \in \Sigma_{>C_{occ}}$, $Type(\sigma_p) = 1$, and $x_i = y_j$. Similarly, for the last formula in the above can be simplified to

$$\max\{L(i-1, j, \boldsymbol{w}),\ L(i, j-1, \boldsymbol{w})\},$$

since $L(i-1, j, \boldsymbol{w}|_{p=w_p+1}) \leq L(i-1, j, \boldsymbol{w})$ holds. However, we write the recursive formula as above to make it easier to check the correspondence between the formulas and the theorems.

Let W be a subset of the candidates of occurrence vector $\boldsymbol{w}$ such as

$$\{(w_1, w_2, \ldots, w_k) \mid 0 \leq w_p \leq C_{occ}(\sigma_p) \text{ if } Type(\sigma_p) = 0 \text{ and; } w_p = 0 \text{ if } Type(\sigma_p) = 1\}.$$

Note that for σ_p with $Type(\sigma_p) = 1$, w_p must be 0 for a repetition-bounded longest common subsequence which means that we successfully delete sufficient number of σ_p's to satisfy $C_{occ}(\sigma_p)$. Finally, we can obtain the length of a repetition-bounded longest common subsequence by computing

$$\max_{\boldsymbol{w} \in W}\{L(|X|, |Y|, \boldsymbol{w})\}. \tag{7}$$

Although the table L only maintains the lengths of repetition-bounded common subsequences, a repetition-bounded longest common subsequence that corresponds to the maximum length can be found by adding a trace-back step (without increasing the time complexity).

5.3.6. Time complexity

We estimate the time complexity of the new DP-based algorithm.

First we consider the maximum value of each w_p in $\boldsymbol{w}$. See the initial value (2) of w_p again. Since w_p does not increase during the calculations of the recursive formulas, the maximum value of w_p is $C_{occ}(\sigma_p)$ if $Type(\sigma_p) = 0$; otherwise, i.e., if $Type(\sigma_p) = 1$, then $occ(X, \sigma_p) - C_{occ}(\sigma_p)$. Recall that for a symbol σ_p with $Type(\sigma_p) = 0$, $C_{occ}(\sigma_p) \leq \frac{1}{2}occ(X, \sigma_p)$ must be satisfied from the definition (1). Moreover, for a symbol σ_p with $Type(\sigma_p) = 1$, $occ(X, \sigma_p) - C_{occ}(\sigma_p) < occ(X, \sigma_p) - \frac{1}{2}occ(X, \sigma_p) = \frac{1}{2}occ(X, \sigma_p)$ holds from the inequality $C_{occ}(\sigma_p) > occ(X, \sigma_p)/2$ in (1). Since w_p is an integer for every $1 \leq p \leq \ell$, $w_p \leq \lfloor \frac{1}{2}occ(X, \sigma_p) \rfloor$ holds for every p.

Recall that for $L(i, j, \boldsymbol{w})$, $0 \le i \le |X|$ and $0 \le j \le |Y|$. Also, for $\sigma_p \in \Sigma_{>C_{occ}}$, $0 \le w_p \le \lfloor \frac{1}{2} occ(X, \sigma_p) \rfloor$ as explained in the above. Therefore, the size of the DP-table L is

$$(|X|+1)(|Y|+1) \prod_{\sigma_p \in \Sigma_{>C_{occ}}} \left(\left\lfloor \frac{1}{2} occ(X, \sigma_p) \right\rfloor + 1 \right),$$

where $\prod_{\sigma_p \in \Sigma_{>C_{occ}}} \left(\lfloor \frac{1}{2} occ(X, \sigma_p) \rfloor + 1 \right)$ corresponds to the number of candidates of $\boldsymbol{w}$.

Let $f(i) = |\{\sigma_p \mid occ(X, \sigma_p) = i,\ \sigma_p \in \Sigma_{>C_{occ}}\}| \times i$, i.e., $\frac{f(i)}{i}$ denotes the number of symbols in $\Sigma_{>C_{occ}}$, that appear exactly i times in X. Then, the number of candidates of $\boldsymbol{w}$ is bounded from above by the following:

$$\begin{aligned}
&\prod_{\sigma_p \in \Sigma_{>C_{occ}}} \left(\left\lfloor \frac{1}{2} occ(X, \sigma_p) \right\rfloor + 1 \right) \\
=&\prod_{i=1}^{|X|} \left(\left\lfloor \frac{i}{2} \right\rfloor + 1 \right)^{\frac{f(i)}{i}} \\
=&1^{f(1)} \times 2^{\frac{1}{2} f(2)} \times 2^{\frac{1}{3} f(3)} \times \prod_{i=4}^{|X|} i^{\frac{f(i)}{i}} \\
\le&(2^{\frac{1}{2}})^{f(1)} \times (2^{\frac{1}{2}})^{f(2)} \times (2^{\frac{1}{2}})^{f(3)} \times \prod_{i=4}^{|X|} (2^{\frac{1}{2}})^{f(i)} \\
=&(2^{\frac{1}{2}})^{\sum_{i=1}^{|X|} f(i)} \\
=&(2^{\frac{1}{2}})^{|X|} \\
<&1.414214^{|X|}, \qquad (8)
\end{aligned}$$

where the first inequality comes from the following facts: (i) $1 \le 2^{1/2}$, (ii) $2^{1/3} \le 2^{1/2}$, and (iii) $i^{1/i} \le 2^{1/2}$ for $i \ge 4$. Thus, the size of the DP-table L is $O(1.414214^{|X|}(|X|+1)(|Y|+1)) = O(1.41422^n)$, where $|X| = n$, $|Y| = poly(n)$, and $|X| \le |Y|$.

Each entry of the DP-table L can be obtained in constant time by (5) and (6). Then (the length of) a repetition-bounded longest common subsequence of X and Y is obtained by (7) in $O(1.414214^n)$ time since $|W|$ is at most the total number of candidates of $\boldsymbol{w}$ whose upper bound is given by (8). In summary, the time complexity of the proposed DP-based algorithm is $O(1.41422^n)$. This completes the proof of Theorem 5.

6. A polynomial-time 2-approximation algorithm for 2FLCS

In this section, we give a polynomial-time algorithm for 2FLCS and show that its approximation ratio is bounded from above by two by using the proof tools introduced in Section 4.1.

Algorithm. Suppose that a standard input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ is given, i.e., $occ(X, \sigma) \ge occ(\mathcal{M}_Y, \sigma)$ and $occ(Y, \sigma) \ge occ(\mathcal{M}_X, \sigma)$ are satisfied. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$. Here is an outline of our algorithm ALG:

(Step 1) Let $X_b = \varepsilon$ and $Y_f = \varepsilon$ be two empty sequences.

- (1-1) While scanning from x_1 to x_n of X, if the ith symbol x_i in X matches a symbol, say, σ_y, in $\mathcal{M}_Y$, then x_i $(= \sigma_y)$ is concatenated to Y_f, i.e., $Y_f = Y_f \oplus \langle \sigma_y \rangle$ and removed from $\mathcal{M}_Y$. Then, obtain a filling $Y^+ = Y_f \oplus Y$ of Y and $\mathcal{M}_Y$.
- (1-2) While scanning from y_1 to y_m of Y, if the ith symbol y_i in Y matches a symbol, say, σ_x, in $\mathcal{M}_X$, then y_i $(= \sigma_x)$ is concatenated to X_b, i.e., $X_b = X_b \oplus \langle \sigma_x \rangle$ and removed from $\mathcal{M}_X$. Then, obtain a filling $X^+ = X \oplus X_b$ of X and $\mathcal{M}_X$ (n.b., not $X_b \oplus X$, while $Y^+ = Y_f \oplus Y$).

(Step 2) Obtain a longest common subsequence Z of two fillings X^+ and Y^+.
(Step 3) Output a solution triple (X^+, Y^+, Z).

Example 6. Consider the input quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ for 2FLCS given in Example 3 again:

$$X = \langle g, t, c, a, c, t, g, a \rangle,\ Y = \langle g, a, t, c, c, g, t, g \rangle,$$
$$\mathcal{M}_X = \{g, t\},\ \text{and}\ \mathcal{M}_Y = \{c, t, t\}$$

In Step (1-1), ALG first constructs $Y_f = \langle t, c, t \rangle$ by scanning X from left to right, and then obtains a filling $Y^+ = Y_f \oplus Y = \langle t, c, t, g, a, t, c, c, g, t, g \rangle$. In Step (1-2), ALG constructs $X_b = \langle g, t \rangle$ by scanning Y from left to right, and then obtains a filling $X^+ = X \oplus X_b = \langle g, t, c, a, c, t, g, a, g, t \rangle$. In Step 2, ALG finds a longest common subsequence $Z = \langle t, c, t, g, a, g, t \rangle$ of X^+ and Y^+, and finally outputs the triple (X^+, Y^+, Z) in Step 3.

See Algorithm 1 for the detailed description of ALG.

Algorithm 1: ALG

Input: Two sequences $X = \langle x_1, \ldots, x_n \rangle$ and $Y = \langle y_1, \ldots, y_m \rangle$; and two multisets $\mathcal{M}_X$ and $\mathcal{M}_Y$
Output: Two fillings X^+ of X and $\mathcal{M}_X$, Y^+ of Y and $\mathcal{M}_Y$, and a common subsequence Z of X^+ and Y^+

$X_b := \varepsilon$, $Y_f := \varepsilon$;
for $i = 1$ to n **do**
  if $x_i = \sigma_y$ for $\sigma_y \in \mathcal{M}_Y$ **then**
    $Y_f := Y_f \oplus \langle \sigma_y \rangle$, $\mathcal{M}_Y := \mathcal{M}_Y \setminus \{\sigma_y\}$;
$Y^+ := Y_f \oplus Y$;
for $i = 1$ to m **do**
  if $y_i = \sigma_x$ for $\sigma_x \in \mathcal{M}_X$ **then**
    $X_b := X_b \oplus \langle \sigma_x \rangle$, $\mathcal{M}_X := \mathcal{M}_X \setminus \{\sigma_x\}$;
$X^+ := X \oplus X_b$;
Find a longest common subsequence Z of the two sequences X^+ and Y^+;
return (X^+, Y^+, Z);

Theorem 10. *Algorithm ALG is a polynomial-time 2-approximation algorithm for 2FLCS on a standard input quadruple* $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$.

Proof. Suppose that the input $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ of 2FLCS is standard. Let $X = \langle x_1, \ldots, x_n \rangle$ and $Y = \langle y_1, \ldots, y_m \rangle$. Then, by applying the arguments of Lemma 2 and Corollary 1 to all the symbols recursively, we can obtain the sequences $\langle i_1, \ldots, i_{|\mathcal{M}_Y|} \rangle$ and $\langle j_1, \ldots, j_{|\mathcal{M}_X|} \rangle$ of different indices that satisfy the following:

$$\begin{aligned} & L(X, Y, \mathcal{M}_X, \mathcal{M}_Y) \\ =& L(X \setminus \langle i_1, \ldots, i_{|\mathcal{M}_Y|} \rangle, Y \setminus \langle j_1, \ldots, j_{|\mathcal{M}_X|} \rangle, \emptyset, \emptyset) + |\mathcal{M}_Y| + |\mathcal{M}_X|. \end{aligned}$$

Clearly, the first term $L(X \setminus \langle i_1, \ldots, i_{|\mathcal{M}_Y|} \rangle, Y \setminus \langle j_1, \ldots, j_{|\mathcal{M}_X|} \rangle, \emptyset, \emptyset)$ of the right-hand side is at most $L(X, Y)$ since $X \setminus \langle i_1, \ldots, i_{|\mathcal{M}_Y|} \rangle$ and $Y \setminus \langle j_1, \ldots, j_{|\mathcal{M}_X|} \rangle$ are subsequences of X and Y, respectively. Therefore, the longest length OPT of 2FLCS is at most $L(X, Y) + |\mathcal{M}_X| + |\mathcal{M}_Y|$.

Let $ALG = |Z|$ be the length obtained by our algorithm ALG for the input $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$, i.e., $ALG = L(X^+, Y^+)$. Since a longest common subsequence of X and Y is a common subsequence of X^+ and Y^+, $ALG \geq L(X, Y)$ holds. Furthermore, since $LCS(X, Y_f) \oplus LCS(X_b, Y)$ is another common subsequence of X^+ and Y^+, $ALG \geq L(X, Y_f) + L(X_b, Y) = |\mathcal{M}_Y| + |\mathcal{M}_X|$ holds. As a result, the approximation ratio of ALG is bounded as follows:

$$\begin{aligned} \frac{OPT}{ALG} &\leq \frac{L(X, Y) + |\mathcal{M}_X| + |\mathcal{M}_Y|}{\max\{L(X, Y), |\mathcal{M}_X| + |\mathcal{M}_Y|\}} \\ &= \frac{2(L(X, Y) + |\mathcal{M}_X| + |\mathcal{M}_Y|)}{2(\max\{L(X, Y), |\mathcal{M}_X| + |\mathcal{M}_Y|\})} \\ &\leq \frac{2(L(X, Y) + |\mathcal{M}_X| + |\mathcal{M}_Y|)}{L(X, Y) + |\mathcal{M}_X| + |\mathcal{M}_Y|} \\ &= 2. \end{aligned}$$

Clearly, ALG runs in polynomial time. This completes the proof. □

For non-standard inputs, we can also obtain a 2-approximation algorithm by slightly modifying ALG. If the input is not standard, then after **Line 10** of ALG, $\mathcal{M}_X$ and $\mathcal{M}_Y$ are not empty. Roughly speaking, all we have to do is to add $\mathcal{M}_X\mathcal{M}_Y$-matches for those "redundant" symbols.

Suppose that $|\mathcal{M}_X \cap \mathcal{M}_Y| = \ell$ at the end of **Line 10** of ALG. Let Z' be an arbitrarily ordered sequence consisting of those ℓ symbols in $\mathcal{M}_X \cap \mathcal{M}_Y$. Then, we modify ALG by adding the concatenation procedures, $X^+ = X^+ \oplus Z'$, $Y^+ = Y^+ \oplus Z'$, and $Z = Z \oplus Z'$ after **Line 10**. See the modified algorithm ALG'.

It can be shown that the approximation ratio of the modified algorithm ALG' is at most two as follows: Consider the status right after **Line 10** in ALG'; we get $\overline{\mathcal{M}_X}$ and $\overline{\mathcal{M}_Y}$ at this moment. Note that the quadruple $(X, Y, \overline{\mathcal{M}_X}, \overline{\mathcal{M}_Y})$ is standard. Let $OPT(X, Y, \overline{\mathcal{M}_X}, \overline{\mathcal{M}_Y})$ be the length of an optimal solution if $(X, Y, \overline{\mathcal{M}_X}, \overline{\mathcal{M}_Y})$ is given as input. Suppose that after **Line 10**, ALG' obtains the subsequence Z_0 for the input $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$. One sees that even if the input is $(X, Y, \overline{\mathcal{M}_X}, \overline{\mathcal{M}_Y})$, ALG' obtains the same subsequence Z_0 after **Line 10**. Therefore, from the similar arguments to the proof of Theorem 10, the following inequality holds:

$$\frac{OPT(X, Y, \overline{\mathcal{M}_X}, \overline{\mathcal{M}_Y})}{|Z_0|} \leq 2.$$

Algorithm 2: ALG'

Input: Two sequences $X = \langle x_1, \ldots, x_n \rangle$ and $Y = \langle y_1, \ldots, y_m \rangle$; and two multisets $\mathcal{M}_X$ and $\mathcal{M}_Y$
Output: Two fillings X^+ of X and $\mathcal{M}_X$, Y^+ of Y and $\mathcal{M}_Y$, and a common subsequence Z of X^+ and Y^+

1 $X_b := \varepsilon$, $Y_f := \varepsilon$;
  // Set $\overline{\mathcal{M}_X} = \emptyset$ and $\overline{\mathcal{M}_Y} = \emptyset$
2 **for** $i = 1$ to n **do**
3 **if** $x_i = \sigma_y$ for $\sigma_y \in \mathcal{M}_Y$ **then**
4 $Y_f := Y_f \oplus \langle \sigma_y \rangle$, $\mathcal{M}_Y := \mathcal{M}_Y \setminus \{\sigma_y\}$;
  // Set $\overline{\mathcal{M}_Y} := \overline{\mathcal{M}_Y} \cup \{\sigma_y\}$ to keep the matched symbols
5 $Y^+ := Y_f \oplus Y$;
6 **for** $i = 1$ to m **do**
7 **if** $y_i = \sigma_x$ for $\sigma_x \in \mathcal{M}_X$ **then**
8 $X_b := X_b \oplus \langle \sigma_x \rangle$, $\mathcal{M}_X := \mathcal{M}_X \setminus \{\sigma_x\}$;
  // Set $\overline{\mathcal{M}_X} := \overline{\mathcal{M}_X} \cup \{\sigma_x\}$ to keep the matched symbols
9 $X^+ := X \oplus X_b$;
10 Find a longest common subsequence Z of the two sequences X^+ and Y^+;
  // Set $Z_0 := Z$ to keep the current common subsequence
11 Construct an arbitrarily ordered sequence Z' consisting of all the symbols in $\mathcal{M}_X \cap \mathcal{M}_Y$;
12 $X^+ := X^+ \oplus Z'$, $Y^+ := Y^+ \oplus Z'$, $Z := Z \oplus Z'$;
13 **return** (X^+, Y^+, Z);

Let $OPT(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ and $ALG(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ be the lengths of an optimal algorithm and ALG', respectively, if the quadruple $(X, Y, \mathcal{M}_X, \mathcal{M}_Y)$ is given as input. Since the sequence Z' of length ℓ is concatenated to Z in **Line 11**, we can obtain the followings:

$$OPT(X, Y, \mathcal{M}_X, \mathcal{M}_Y) = OPT(X, Y, \overline{\mathcal{M}_X}, \overline{\mathcal{M}_Y}) + \ell$$
$$ALG(X, Y, \mathcal{M}_X, \mathcal{M}_Y) = |Z_0| + \ell.$$

Hence, for non-standard inputs, the approximation ratio of ALG' is bounded as follows:

$$\begin{aligned} \frac{OPT(X, Y, \mathcal{M}_X, \mathcal{M}_Y)}{ALG(X, Y, \mathcal{M}_X, \mathcal{M}_Y)} &= \frac{OPT(X, Y, \overline{\mathcal{M}_X}, \overline{\mathcal{M}_Y}) + \ell}{|Z_0| + \ell} \\ &\leq \frac{OPT(X, Y, \overline{\mathcal{M}_X}, \overline{\mathcal{M}_Y})}{|Z_0|} \\ &\leq 2. \end{aligned}$$

Corollary 5. *Algorithm ALG' is a polynomial-time 2-approximation algorithm for 2FLCS.*

7. Conclusion

We have studied four variants of the problem of computing the longest common subsequence of two sequences X and Y (LCS): the Repetition-Bounded Longest Common Subsequence problem (RBLCS), the Multiset-Restricted Common Subsequence problem (MRCS), the Two-Side-Filled Longest Common Subsequence problem (2FLCS), and the One-Side-Filled Longest Common Subsequence problem (1FLCS). We first showed that each of MRCS, 1FLCS, and 2FLCS is polynomially equivalent to RBLCS. Then, we designed a DP-based algorithm for RBLCS that runs in $O(1.41422^n)$ time, which implies that MRCS, 1FLCS, and 2FLCS can also be solved in $O(1.41422^n)$ time. Finally, we gave a polynomial-time 2-approximation algorithm for 2FLCS, and answered one conjecture in [7] affirmatively.

For MRCS, 1FLCS, and RBLCS, a $2\sqrt{\min\{n, m\}}$-approximation algorithm [13], a $\frac{5}{3}$-approximation algorithm [7], and an occ_{max}-approximation algorithm [1] are known, respectively, where $occ_{max} = \max_{\sigma \in \Sigma}\{\min\{occ(X, \sigma), occ(Y, \sigma)\}\}$, $|X| = n$, and $|Y| = m$. Future work includes designing even better approximation algorithms and faster (exponential-time) exact algorithms for the LCS variants studied here.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is partially supported by NSERC Canada, JST CREST JPMJR1402, and JSPS, Japan KAKENHI Grant Numbers JP20H05967, JP21K11755, JP21K19765, JP22H00513, JP22K11915, and JP24K02902.

References

[1] Said Sadique Adi, Marília D.V. Braga, Cristina G. Fernandes, Carlos Eduardo Ferreira, Fábio Viduani Martinez, Marie-France Sagot, Marco Aurelio Stefanes, Christian Tjandraatmadja, Yoshiko Wakabayashi, Repetition-free longest common subsequence, Electron. Notes Discret. Math. 30 (2008) 243–248.
[2] Yuichi Asahiro, Jesper Jansson, Guohui Lin, Eiji Miyano, Hirotaka Ono, Tadatoshi Utashima, Exact algorithms for the repetition-bounded longest common subsequence problem, Theoret. Comput. Sci. 838 (2020) 238–249.
[3] Yuichi Asahiro, Jesper Jansson, Guohui Lin, Eiji Miyano, Hirotaka Ono, Tadatoshi Utashima, Polynomial-time equivalences and refined algorithms for longest common subsequence variants, in: Hideo Bannai, Jan Holub (Eds.), 33rd Annual Symposium on Combinatorial Pattern Matching, CPM 2022, June 27–29, 2022, Prague, Czech Republic, in: LIPIcs, vol. 223, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 15:1–15:17.
[4] Lasse Bergroth, Harri Hakonen, Timo Raita, A survey of longest common subsequence algorithms, in: Pablo de la Fuente (Ed.), Seventh International Symposium on String Processing and Information Retrieval, SPIRE 2000, A Coruña, Spain, September 27–29, 2000, IEEE Computer Society, 2000, pp. 39–48.
[5] Laurent Bulteau, Falk Hüffner, Christian Komusiewicz, Rolf Niedermeier, Multivariate algorithmics for NP-hard string problems: The algorithmics column by Gerhard J. Woeginger, Bull. EATCS 114 (2014).
[6] Mauro Castelli, Riccardo Dondi, Giancarlo Mauri, Italo Zoppis, The longest filled common subsequence problem, in: Juha Kärkkäinen, Jakub Radoszewski, Wojciech Rytter (Eds.), 28th Annual Symposium on Combinatorial Pattern Matching, CPM 2017, July 4-6, 2017, Warsaw, Poland, in: LIPIcs, vol. 78, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 14:1–14:13.
[7] Mauro Castelli, Riccardo Dondi, Giancarlo Mauri, Italo Zoppis, Comparing incomplete sequences via longest common subsequence, Theoret. Comput. Sci. 796 (2019) 272–285.
[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 4th Edition, MIT Press, 2022.
[9] Daniel S. Hirschberg, A linear space algorithm for computing maximal common subsequences, Commun. ACM 18 (6) (1975) 341–343.
[10] Daniel S. Hirschberg, Algorithms for the longest common subsequence problem, J. ACM 24 (4) (1977) 664–675.
[11] Haitao Jiang, Chunfang Zheng, David Sankoff, Binhai Zhu, Scaffold filling under the breakpoint and related distances, IEEE ACM Trans. Comput. Biol. Bioinform. 9 (4) (2012) 1220–1229.
[12] Roy Lowrance, Robert A. Wagner, An extension of the string-to-string correction problem, J. ACM 22 (2) (1975) 177–183.
[13] Radu Stefan Mincu, Alexandru Popa, Better heuristic algorithms for the repetition free LCS and other variants, in: Travis Gagie, Alistair Moffat, Gonzalo Navarro, Ernesto Cuadros-Vargas (Eds.), String Processing and Information Retrieval - 25th International Symposium, SPIRE 2018, Lima, Peru, October 9–11, 2018, Proceedings, in: Lecture Notes in Computer Science, vol. 11147, Springer, 2018, pp. 297–310.
[14] Radu Stefan Mincu, Alexandru Popa, Heuristic algorithms for the longest filled common subsequence problem, in: 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2018, Timisoara, Romania, September 20–23, 2018, IEEE, 2018, pp. 449–453.
[15] Adriana Muñoz, Chunfang Zheng, Qian Zhu, Victor A. Albert, Steve Rounsley, David Sankoff, Scaffold filling, contig fusion and comparative gene order inference, BMC Bioinform. 11 (2010) 304.
[16] Robert A. Wagner, Michael J. Fischer, The string-to-string correction problem, J. ACM 21 (1) (1974) 168–173.