String Editing Under a Combination of Constraints

Slobodan Petrović* and Jovan Golić**

*Institute of Applied Mathematics and Electronics, Belgrade

**Institute of Applied Mathematics and Electronics, Belgrade
School of Electrical Engineering, University of Belgrade

Abstract

Let X and Y be any two strings of finite lengths N and M, respectively, over a finite alphabet. An edit distance between X and Y is defined as the minimum sum of elementary edit distances associated with edit operations of substitutions, deletions, and insertions needed to transform X to Y . In this paper, the problem of efficient computation of such a distance is considered under the assumption that the numbers of edit operations are limited and that the maximum lengths, F and G, of runs of deletions and insertions are given, respectively. Besides, the edit sequence is ordered in a sense that between every two successive runs of substitutions there can be either at most one run of deletions followed by at most one run of insertions or just one run of deletions or insertions. An algorithm is derived that computes the minimum edit distance associated with editing X to Y subject to the specified constraints. The algorithm requires $O(NM \min\{N, M\})$ time and $O(NM)$ space. Possible applications for the synchronization error-correcting codes and for the cryptanalysis of certain stream ciphers are also discussed.

1. INTRODUCTION

The concepts of sequence comparison and string editing have been studied in order to develop efficient text-editing systems and error-correcting compilers [3]. They have also been studied in the research of molecular evolution of organisms [7, 8] as well as in the decoding for channels with deletions, insertions, and substitutions [1, 4]. An excellent review of the results in this area can be found in [7]. Recently, a string editing algorithm for the cryptanalysis of key-stream generators based on the clock-controlled shift registers has been derived [2].

A basic problem in string editing is to establish an appropriate distance measure between two strings. A widely used measure is the generalized edit distance, (see [7], for example). It is defined as the minimum sum of elementary edit distances associated with substitutions, insertions, and deletions needed to transform one string into another. Certain constraints are usually incorporated in the general definition of the edit distance in connection with particular applications. Three types of constraints are characteristic:

(a) the total number of insertions and/or deletions [5, 6, 7];

(b) the maximum lengths of runs of insertions and/or deletions [2, 7], where 'run' is defined to be a sequence of consecutive identical edit operations that is neither preceeded nor succeeded by the the same edit operation ;

(c) the total number of runs of deletions and/or insertions [7].

Combinations of these types are also possible. In this paper we derive an algorithm for computing the edit distance between two strings of finite lengths, under the constraints based on the combination of the type a) and a specific variation of the type b); namely, we consider the following problem.

Given two arbitrary finite length strings X and Y of symbols from a finite alphabet $\mathcal{A}$, and the constraints

1. the number of insertions belongs to a set T;
2. the maximum lengths of runs of deletions and insertions are F and G, respectively;
3. the edit sequence is ordered in a sense that between every two successive runs of substitutions there can be at most one run of deletions followed by at most one run of insertions;
4. the edit sequence is ordered in a sense that between every two successive runs of substitutions there can be at most one run of deletions or at most one run of insertions;

determine the edit distance between X and Y under the combination of the constraints 1, 2, and 3 or 1, 2, and 4. Note that every editing transformation as a sequence of insertions, deletions, and substitutions can be equivalently represented in an ordered way defined by the third constraint. The fourth constraint is quite natural in most applications, especially for channels with synchronization errors. As for the first constraint, it has been shown [5, 6] that every constraint concerning the limited number of deletions, insertions, and substitutions can be expressed only in terms of the number of insertions.

In the literature, two basic approaches to the solution of the problems of this type are known. The first approach involves recursive computation of the distance directly; that is, the recursion deals with the input sequences themselves (see [7], for example). The second approach involves recursive computation of elements of the array of partial constrained edit distances [2, 5, 6]. For calculating the constrained edit distance under the combination of the given constraints 1, 2, and 3 or 4, it is difficult to modify the algorithms using the first approach. Such modifications generally yield complex algorithms. We find the second approach much more convenient for solving our problem. We propose an algorithm, that is efficient and consumes moderate space.

The second approach hitherto has been used only for determining the constrained edit distance under the constraint 1 in the general case [5, 6], or constraint 2 in the special case where insertions are not allowed [2]. In this paper we shall take the same approach in order to compute the edit distance under the combination of the constraints 1, 2, and 3 or 4.

We first define elementary edit distances associated with deletions, insertions, and substitutions and then give a mathematical definition of the constrained edit distance in terms of the permitted edit sequences. Further, we develop a recursion for efficient computation of the required distance, and finally present an algorithm that implements this recursion. Possible applications are also discussed.

2. PRELIMINARIES

In the previous section the statement of the problem along with some basic concepts was given. In this section we provide further mathematical definitions and details and introduce a suitable expression for the constrained edit distance.

Given arbitrary strings X and Y of finite lengths N and M, respectively, over a finite alphabet $\mathcal{A}$, we deal with the problem of transforming the string X to Y by means of the edit operations of deletion, insertion, and substitution, under the constraints 1, 2, and 3 or 4. The constrained edit distance $D(X, Y)$ is then defined as the minimum sum of elementary edit distances associated with the edit operations of deletion, insertion, and substitution needed to transform X to Y, subject to the assumed constraints.

We first define elementary distances associated with the edit operations of deletion, insertion, and substitution of symbols. Along with $\mathcal{A}$, we introduce the null symbol ϕ reserved for the operations of deletion and insertion. Let $\mathcal{A}^* = \mathcal{A} \cup \phi$. Nonnegative real-valued elementary distances are defined in the following way:

1. $d(a, \phi)$ is the elementary distance associated with deleting a symbol $a \in \mathcal{A}$;
2. $d(\phi, b)$ is the elementary distance associated with inserting a symbol $b \in \mathcal{A}$;
3. $d(a, b)$ is the elementary distance associated with substituting b for a, $a, b \in \mathcal{A}$.

We also define the compression operator C in the following way. For an arbitrary finite length string Z over $\mathcal{A}^*$, $C(Z)$ stands for the string over $\mathcal{A}$ that is obtained from Z by removing all the null symbols from it (see [5], for example). Besides, let $\mid Z \mid$ denote the length of Z ($Z = z_1, \ldots, z_{|Z|}$).

As explained in the previous section, every editing transformation as a sequence of insertions, deletions, and substitutions can be equivalently represented in an ordered way defined by constraint 3 or 4. We call such a representation an *edit sequence* and denote it by $S = (X', Y')$, where $(X', Y') \in \mathcal{A}^* \times \mathcal{A}^*$, and for every (X', Y') the following rules hold:

1° $C(X') = X$, $C(Y') = Y$;

2° $\mid X' \mid = \mid Y' \mid = L$, where $\max\{N, M\} \leq L \leq N + M$;

3° for all i, $1 \leq i \leq \mid X' \mid$, it is not the case that $x'_i = y'_i = \phi$;

4° (X', Y') satisfies constraints 2 and 3 or 4 introduced in the previous section.

The preceding set of the rules has standard interpretation [5]. That is, for any $1 \leq i \leq L$: if both x'_i and y'_i are non-null symbols, then the substitution of y'_i for x'_i takes place; if x'_i is the null symbol and y'_i is a non-null symbol, then the insertion of y'_i takes place; and if x'_i is a non-null symbol and y'_i is the null symbol, then the deletion of x'_i takes place.

By $G(X, Y)$ we denote the set of all the permitted edit sequences S that transform the string X to the string Y. Precisely, $G(X, Y) = \{(X', Y') \mid (X', Y')$ satisfies the conditions 1° to $4^\circ\}$. Also, let $G_T(X, Y)$ denote the set of all the edit sequences from $G(X, Y)$ that satisfy the constraint 1 as well. Representation of an editing transformation by an edit sequence $S = (X', Y')$ is unique, which means that there is a one-to-one correspondence between the set $G(X, Y)$ of all the edit sequences (X', Y') and the set of all the permitted editing transformations of X to Y.

EXAMPLE. Let $\mathcal{A} = \{0, 1\}$, $T = \{5, 6\}$, $F = 2$, $G = 3$, $N = 8$, $M = 9$, $X =$ 11100110, and $Y = 100110110$. Two permitted editing transformations from $G_T(X, Y)$ are $(11\phi\phi\phi100110\phi\phi, \phi\phi1001101\phi\phi10)$ and $(11\phi\phi\phi1001\phi\phi\phi10, \phi\phi10011\phi\phi0110\phi)$.

Thus, for the constrained edit distance we have an expression:

$$D(X,Y) = \min\left\{\sum_{i=1}^{L} d(x'(i), y'(i)) \mid (X', Y') \in G_T(X,Y)\right\}, \tag{1}$$

where $L = N + M - s$ and $0 \le s \le \min\{N, M\}$, depends on (X', Y').

Having an explicit expression Equation (1) for the constrained edit distance, the question is whether the minimization can be performed efficiently. Direct computation using Equation (1) is not efficient because it involves the blind search over the set of all possible editings of X to Y. Therefore, following the concepts from [5], we introduce a new quantity, the *partial constrained edit distance* $W(i,e,s)$ as a constrained edit distance between the prefix X_{e+s} of length $e+s$ and the prefix Y_{i+s} of length $i+s$, where for an arbitrary finite length string Z over $\mathcal{A}$, Z_k denotes the prefix of length k. More precisely, let $G_{i,e,s}(X,Y)$ or simply $G_{i,e,s}$ denote the set of all the edit sequences from $G(X_{e+s}, Y_{i+s})$ that transform X_{e+s} to Y_{i+s} using exactly i insertions, e deletions, and s substitutions. Say that a triple (i,e,s) is feasible if $G_{i,e,s}$ is not empty. Then

$$W(i,e,s) = \min\left\{\sum_{j=1}^{i+e+s} d(x'(j), y'(j)) \mid (X'_{e+s}, Y'_{i+s}) \in G_{i,e,s}\right\}. \tag{2}$$

In [5], the connection between the constrained edit distance and the elements of the array of partial constrained edit distances has been established. It is easy to verify that the same connection holds here as well and hence for the constrained edit distance we have

$$D(X,Y) = \min_{i \in T} W(i, N-M+i, M-i). \tag{3}$$

where each i from T automatically satisfies the inequality $max\{0, M-N\} \le i \le M$.

As in [5], the main idea here is to establish a recursive property of $W(i,e,s)$ that will enable efficient computation of $D(X,Y)$. In the next section we show that this is possible. However, specific editing transformation and specific constraints lead to the recursion which is essentially different from the expressions derived in [5, 6].

Note that the set of feasible values of (i,e,s) is, due to the assumed constraints, given by

$$0 \le s \le \min\{N, M\}, \tag{4}$$

$$0 \le e \le \min\{N-s, (s+1)F\}, \tag{5}$$

$$0 \le i \le \min\{M-s, (s+1)G\}. \tag{6}$$

Namely, a triple (i,e,s) is feasible if and only if it satisfies (4), (5), and (6).

3. RECURSION

In this section we establish a recursion satisfied by the partial constrained edit distance $W(i,e,s)$, introduced by Equation (2) in the previous section. The following theorem enables efficient computation of the constrained edit distance $D(X,Y)$.

Theorem:

(1) Under the combination of the constraints 1,2, and 3, the partial constrained edit distance $W(i,e,s)$ satisfies the recursion:

$$W(i,e,s) = \min_{(i_1,e_1) \in Q}\left\{W(i-i_1, e-e_1, s-1) + \sum_{j=1}^{e_1} d(x(e+s-e_1+j), \phi) +\right.$$

$$+\sum_{j=1}^{i_1} d(\phi, y(i+s-i_1+j)) + d(x(e+s-e_1), y(i+s-i_1))\Bigg\}, \tag{7}$$

$1 \le s \le \min\{N, M\}$, $0 \le e \le \min\{N-s, (s+1)F\}$, $0 \le i \le \min\{M-s, (s+1)G\}$, where Q is the set of all the pairs (i_1, e_1) such that

$$\max\{0, i - \min\{M-s+1, sG\}\} \le i_1 \le \min\{G, i\} \tag{8}$$

$$\max\{0, e - \min\{N-s+1, sF\}\} \le e_1 \le \min\{F, e\}. \tag{9}$$

For $s = 0$, $0 \le e \le \min\{N, F\}$, and $0 \le i \le \min\{M, G\}$,

$$W(i, e, 0) = \sum_{j=1}^{e} d(x(j), \phi) + \sum_{j=1}^{i} d(\phi, y(j)). \tag{10}$$

(2) Under the combination of the constraints 1, 2, and 4, the partial constrained edit distance $W(i, e, s)$ satisfies the recursion:

$$W(i,e,s) = \min\Bigg\{ \min_{i_1 \in Q_G} \Bigg\{ W(i-i_1, e, s-1) + \sum_{j=1}^{i_1} d(\phi, y(i+s-i_1+j)) + \\ + d(x(e+s), y(i+s-i_1)) \Bigg\}, \\ \min_{e_1 \in Q_F} \Bigg\{ W(i, e-e_1, s-1) + \sum_{j=1}^{e_1} d(x(e+s-e_1+j), \phi) + \\ d(x(e+s-e_1), y(i+s)) \Bigg\} \Bigg\} \tag{11}$$

$1 \le s \le \min\{N, M\}$, $0 \le e \le \min\{N-s, (s+1)F\}$, $0 \le i \le \min\{M-s, (s+1)G\}$, where Q_G and Q_F are the sets of all $i_1 \ge 1$ and e_1 such that (8) and (9) hold, respectively. For $s = 0$, $0 \le e \le \min\{N, F\}$, and $0 \le i \le \min\{M, G\}$,

$$W(i, e, 0) = \min\left\{ \sum_{j=1}^{i} d(\phi, y(j)), \sum_{j=1}^{e} d(x(j), \phi) \right\}. \tag{12}$$

The proof of the theorem is given in the Appendix. In a particular, practically important case when the deletion or insertion elementary distances are equal for all the symbols, the formulation of the theorem is obvious.

4. ALGORITHM

In this section, we combine Equation (3) with the recursion Equations (7) - (10) in an algorithm for efficient computation of the constrained edit distance under the combination of constraints 1, 2, and 3. We also discuss its computational complexity. For the combination of constraints 1, 2, and 4 the algorithm is analogous.

Algorithm:

Parameters:

$\mathcal{A}$	$\rightarrow$	a finite alphabet;
F	$\rightarrow$	maximum number of consecutive deletions in the string X;
G	$\rightarrow$	maximum number of consecutive insertions in the string Y;
$d(x, \phi)$	$\rightarrow$	elementary distance associated with deleting a symbol x, for all $x \in \mathcal{A}$;
$d(\phi, y)$	$\rightarrow$	elementary distance associated with inserting a symbol y, for all $y \in \mathcal{A}$;
$d(x, y)$	$\rightarrow$	elementary distance associated with substituting a symbol y for a symbol x, for all $(x, y) \in \mathcal{A} \times \mathcal{A}$.

Input:
Two strings X and Y of lengths N and M, respectively, over $\mathcal{A}$.

Output:
Constrained edit distance between the strings X and Y, under the constraints 1, 2, and 3, introduced in Section 1.

1. Initialization
   FOR $e := 0$ TO $\min\{N, F\}$ DO
     FOR $i := 0$ TO $\min\{M, G\}$ DO
       $W_0[i, e] := \sum_{j=1}^{e} d[x[e], \phi] + \sum_{j=1}^{i} d[\phi, y[i]]$;
2. Main loop
   $D :=$ MAXINT ;
   FOR $s := 1$ TO $\min\{N, M\}$ DO
     BEGIN
       FOR $e := 0$ TO $\min\{N - s, (s + 1)F\}$ DO
         BEGIN
           FOR $i := 0$ TO $\min\{M - s, (s + 1)G\}$ DO
             BEGIN
               $W_1[i, e] :=$ MAXINT ;
               FOR $e_1 := \max\{0, e - \min\{N - s + 1, sF\}\}$ TO $\min\{F, e\}$ DO
                 BEGIN
                   FOR $i_1 := \max\{0, i - \min\{M - s + 1, sG\}\}$ TO $\min\{G, i\}$ DO
                     BEGIN (** Update the matrix W_1 **)
   $W_1[i, e] := \min\{W_1[i, e], W_0[i - i_1, e - e_1] + \sum_{j=1}^{e_1} d[x[e + s - e_1 + j], \phi] +$
   $+ \sum_{j=1}^{i_1} d[\phi, y[i + s - i_1 + j]] + d[x[e + s - e_1], y[i + s - i_1]]\}$;
                     END ;
                 END ;
             END ;
         END ;
       IF $(M - s) \in T$ THEN
         $D := \min\{D, W_1[M - s, N - s]\}$;
       $W_0 := W_1$ (** Copy W_1 to W_0 **)
     END ;
3. Final computation

   $D(X, Y) := D$.

We now consider the time and space complexities of the algorithm. Clearly, the time complexity is determined by the complexity of the main loop. The number of

operations consumed by that loop is proportional to $FGNM \min\{N, M\}$. Because F and G as parameters to the algorithm are constant, the time complexity is upper-bounded by $O(NM \min\{N, M\})$.

As for the space complexity, note that we reduced the dimension of the array $W(i, e, s)$ by 1 during the implementation of the formula (7), in view of the fact that for any $s = 1, \ldots, \min\{N, M\}$, in order to compute $W(i, e, s)$ for all the feasible values of (i, e), it suffices to store only the values of $W(i, e, s-1)$ for all the corresponding feasible values of (i, e). So, instead of taking an array W of dimension 3, we take two matrices of dimension 2, W_0 and W_1, corresponding to the previous and the current values of s, respectively. Hence, the space complexity of the algorithm is $O(NM)$. Note also that computing the edit distance is transferred into the main loop, unlike most of the algorithms for solving similar problems (see, [5, 6], for example), in order to enable reduction of the dimensionality of the array W.

Finally, having computed the partial constrained edit distances $W(i, e, s)$ for all the required values of (i, e, s), one obtains not only the constrained edit distance, but can also reconstruct an optimum edit sequence by backtracking through the array $W(i, e, s)$, starting from the element corresponding to $D(X, Y)$, see [5, 6]. Note that an optimum edit sequence is not unique in general.

5. APPLICATIONS

In most applications of sequence comparison one assumes that a sequence out of a set of finite length sequences is transformed into a variable length sequence by a stochastic sequence of deletions, insertions, and/or substitutions. A stochastic edit process is usually with high probability subject to a set of constraints often related to the numbers of edit operations and/or the maximum lengths of runs of deletions and insertions. Then, using a minimum edit distance decision rule, the objective is to recover the original sequence given the observed one. The more constraints satisfied either deterministically or with high probability one incorporates into an edit distance, the more successful the statistical decision will be. The four types of constraints that we consider in this paper are expected to be satisfied with high probability in most applications.

We proceed now by describing in more detail two particular potential applications of the preceding kind. First, consider a communication channel with additive noise and synchronization errors due to nonstationary delay characteristics. The constraints 1, 2, and 4 are then effective with high probability. The constrained edit distance involving these constraints, which can be computed efficiently according to the second part of our theorem, can then be used for the error-correcting decoding procedure.

Second, consider a stream cipher with a binary pseudorandom sequence generator consisting of three blocks. The first block is a memoryless combination of several regularly or irregularly clocked linear feedback shift registers (LFSR's) [2], which produces a binary output sequence. The second and the third blocks are arbitrary schemes that produce a binary insertion sequence and an integer insertion control sequence, respectively. The three blocks are combined by an interleaver that inserts a segment of the insertion sequence of length determined by the insertion control sequence before each bit of the binary sequence generated by the first block. Now, as in [2], assume that the output sequence of the first block is correlated to an irregularly clocked LFSR input sequence. Then the output sequence of the whole generator can be obtained from the regularly clocked LFSR input sequence by a pseudorandom edit sequence of deletions, insertions, and substitutions subject to the combination of constraints 1, 2,

and 3. Thus in order to reconstruct the LFSR initial state given the output sequence, one can apply essentially the same statistical procedure as in [2], but using the new constrained edit distance, which can be computed efficiently according to the first part of our theorem.

6. CONCLUSION

In this paper we consider the problem of editing a string X to a string Y subject to the combination of edit constraints represented by the numbers of editing operations and the maximum numbers of consecutive deletions and insertions. Besides, an edit sequence is ordered in a sense that between every two successive runs of substitutions there can be either at most one run of deletions followed by at most one run of insertions, or just one run of deletions or insertions. A novel efficient algorithm for computing this distance is derived. Its time and space complexities are shown to be $O(NM\min\{N,M\})$ and $O(NM)$, respectively, where N and M are the lengths of the strings X and Y, respectively. An optimum edit transformation can be obtained by backtracking through the array of partial constrained edit distances. The proposed algorithm may have applications in text-editing systems, error-correcting compilers, error-correcting decoders for communication channels with additive noise and synchronization errors, cryptanalysis of certain stream ciphers, and also in molecular biology for macromolecule sequence comparison.

APPENDIX

Proof of the Theorem:

(1) For $s = 0$ the proof is trivial. Assume that $s \geq 1$. We start from the basic expression Equation (2) for $W(i,e,s)$. Following the dynamic programming principle [5, 6] we also partition the set of all the permitted edit sequences $G_{i,e,s}$ into a number of subsets. How we do it makes our proof specific and nontrivial. The main idea is to divide the set $G_{i,e,s}$ according to the sequences of deletions and insertions *after the last substitution*; namely, let $G^q_{i,e,s}$, $q = (i_1, e_1)$ denote the set of all the sequences in $G_{i,e,s}$ that end with e_1 deletions and i_1 insertions, after the last substitution, for $0 \leq e_1 \leq \min\{F, e\}$, $0 \leq i_1 \leq \min\{G, i\}$. However, some of these subsets may be empty. It follows that $G^q_{i,e,s}$ is empty iff $(i - i_1, e - e_1, s - 1)$ is not feasible. Therefore, in view of Equations (4), (5), and (6) we obtain that $G^q_{i,e,s}$ is not empty iff Equations (8) and(9) are true. Accordingly, we partition the set $G_{i,e,s}$ into the subsets $G^q_{i,e,s}$, $q \in Q$. Rewriting Equation (2), we get

$$W(i,e,s) = \min_{q\in Q}\left\{\min\left\{\sum_{j=1}^{i+e+s-i_1-e_1-1} d(x'(j),y'(j)) + d(x(e+s-e_1), y(i+s-i_1)+\right.\right.$$

$$\left.\left. + \sum_{j=1}^{e_1} d(x(e+s-e_1+j),\phi) + \sum_{j=1}^{i_1} d(\phi, y(i+s-i_1+j)) \mid (X'_{e+s}, Y'_{i+s}) \in G^q_{i,e,s}\right\}\right\} \tag{13}$$

breaking the sum into three parts corresponding to the sequence of edit operations before the last substitution, to the last substitution, and to the sequence of edit operations after the last substitution, respectively. Given the set $G^q_{i,e,s}$, the contributions of the second and third part are constant and hence

$$W(i,e,s) = \min_{q\in Q}\left\{\sum_{j=1}^{e_1} d(x(e+s-e_1+j),\phi) + \sum_{j=1}^{i_1} d(\phi, y(i+s-i_1+j))+\right.$$

$$+d(x(e+s-e_1), y(i+s-i_1))+$$

$$\left.+\min\left\{\sum_{j=1}^{i+e+s-i_1-e_1-1} d(x'(j),y'(j)) \mid (X'_{e+s}, Y'_{i+s}) \in G^q_{i,e,s}\right\}\right\}. \qquad (14)$$

We have thus come to the main point: the minimization over $G^q_{i,e,s}$. Because every edit sequence in $G^q_{i,e,s}$ ends in one substitution followed by exactly e_1 deletions and i_1 insertions, and the constraints relate to the numbers of consecutive deletions and insertions, it follows that an edit sequence (X'_{e+s}, Y'_{i+s}) belongs to $G^q_{i,e,s}$ iff its prefix $(X'_{e-e_1+s-1}, Y'_{i-i_1+s-1})$ belongs to $G_{i-i_1,e-e_1,s-1}$, for any $q \in Q$. Therefore, considering Equation (2), we have

$$\min\left\{\sum_{j=1}^{i+e+s-i_1-e_1-1} d(x'(j),y'(j)) \mid (X'_{e+s}, Y'_{i+s}) \in G^q_{i,e,s}\right\} =$$

$$= \min\left\{\sum_{j=1}^{i+e+s-i_1-e_1-1} d(x'(j),y'(j)) \mid (X'_{e-e_1+s-1}, Y'_{i-i_1+s-1}) \in G_{i-i_1,e-e_1,s-1}\right\} =$$

$$= W(i-i_1, e-e_1, s-1). \qquad (15)$$

Consequently, Eq. (14) is reduced to Eq. (7).

(2) For $s = 0$ the proof is trivial. Assume that $s \geq 1$. Start with the partition of the set of all the permitted edit sequences $G_{i,e,s}$ into the sets $G^{i_1}_{i,e,s}$, $1 \leq i_1 \leq \min\{G, i\}$, and $G^{e_1}_{i,e,s}$, $0 \leq e_1 \leq \min\{F, e\}$, that denote the sets of all the sequences in $G_{i,e,s}$ that end with i_1 insertions and e_1 deletions after the last substitution, respectively. The proof is then analogous to the proof of the first part of the theorem. Q.E.D.

Remark. Unlike the dynamic programming expressions related to string editing (see [5, 6, 7], for example), the order of the recursion regarding insertions and deletions is greater than 1 due to the specific partition of the set of permitted edit sequences $G_{i,e,s}$. It is necessary to introduce this partition in order to deal with the specific constraints considered in this paper.

References

[1] L.R. Bahl and F. Jelinek, Decoding for channels with insertions, deletions and substitutions with applications to speech recognition, *IEEE Trans. Inform. Theory* IT-21(4):404-411 (1975).

[2] J. Golić and M. Mihaljević, A noisy clock-controlled shift register cryptanalysis concept based on sequence comparison approach, *Advances in Cryptology - EUROCRYPT '90, Lecture Notes in Computer Science*, vol. 473, pp. 487-491, Springer-Verlag, 1991.

[3] P.A.V. Hall and G.R. Dowling, Approximate string matching, *Comput. Surveys* 12:381-402 (1980).

[4] A. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, *Soviet Phys. Dokl.* 10:707-710 (1966).

[5] B.J. Oommen, Constrained string editing, *Inform. Sci.* 40:267-284 (1986).

[6] B.J. Oommen, Recognition of noisy subsequences using constrained edit distances, *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-9(5):676-685 (1987).

[7] D. Sankoff and J.B. Kruskal (eds.), *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley, 1983.

[8] P.H. Sellers, On the theory and computation of evolutionary distances, *SIAM J. Appl. Math.* 26(4):787-793, (1974).