Title
The undecidability of the modified edit distance

Author
Leung, Vitus J.

Publication Date
1994-11-11

Peer reviewed

THE UNDECIDABILITY OF THE MODIFIED EDIT DISTANCE

Vitus J. Leung

Department of Information and Computer Science
University of California, Irvine

Technical Report 94-45
November 11, 1994

Abstract

Given two strings, X and Y, over a finite alphabet Σ, the *modified edit distance* between X and Y is the minimal cost of an edit sequence that changes X into Y, where the cost of substituting a character in Y for a character in X is context free, and the cost of deleting a substring from X or inserting a substring from Y into X is somewhat context sensitive. The modified edit distance does not require that the minimum cost over all edit sequences where the cost of substituting a character in Σ for a character in a string is context free, the cost of deleting a substring from a string is somewhat context sensitive, and the cost of inserting a string Z into X to obtain a string X' is equivalent to the cost of deleting Z from X' to obtain X again. We show that if the minimum cost over all edit sequences must be obtained, the modified edit distance becomes undecidable.

THE UNDECIDABILITY OF THE MODIFIED EDIT DISTANCE

Vitus J. LEUNG

Department of Information and Computer Science, University of California, Irvine, CA 92717, U.S.A.

ABSTRACT

Given two strings, X and Y, over a finite alphabet Σ, the *modified edit distance* between X and Y is the minimal cost of an edit sequence that changes X into Y, where the cost of substituting a character in Y for a character in X is context free, and the cost of deleting a substring from X or inserting a substring from Y into X is somewhat context sensitive. The modified edit distance does not require that the minimum cost over all edit sequences where the cost of substituting a character in Σ for a character in a string is context free, the cost of deleting a substring from a string is somewhat context sensitive, and the cost of inserting a string Z into X to obtain a string X' is equivalent to the cost of deleting Z from X' to obtain X again. We show that if the minimum cost over all edit sequences must be obtained, the modified edit distance becomes undecidable.

1. Introduction

Galil and Giancarlo [1] [2] define the modified edit distance as follows. Given two strings over a finite alphabet Σ, $X=x_1x_2 \cdots x_m$ and $Y=y_1y_2 \cdots y_n$, the *modified edit distance* between X and Y is the minimal cost of an edit sequence that changes X into Y, where the cost of substituting a character in Y for a character in X is context free, and the cost of deleting a substring from X or inserting a substring from Y into X is somewhat context sensitive. Formally, the cost of deleting $x_{k+1}x_{k+2} \cdots x_i$ from X is $w_X(k,i)=f_1(x_k,x_{k+1})+f_2(x_i,x_{i+1})+g(i-k)$, where $1 \leq k < i < m$.

This cost consists of charges for breaking X between x_k and x_{k+1} and between x_i and x_{i+1} plus an additional charge that depends on the length of the substring from x_{k+1} to x_i. The cost of inserting $y_{k+1}y_{k+2}\cdots y_j$ into X is equivalent to the cost of deleting $y_{k+1}y_{k+2}\cdots y_j$ from Y, where $1 \leq k < j < n$. The cost of substituting y_j for x_i is $s(x_i, y_j)$, where $1 \leq i \leq m$ and $1 \leq j \leq n$. The cost of an edit sequence is the total cost of all its operations.

To compute the modified edit distance, Galil and Giancarlo consider the recurrence relation D_{XY} given in Figure 1. Note, however, that D_{XY} will find the minimal cost of an edit sequence where substrings are deleted from X first, then characters in Y are substituted for characters in X next, and finally substrings from Y are inserted into X last. Furthermore, the modified edit distance does not require that the minimum cost over all edit sequences where the cost of substituting a character in Σ for a character in a string is context free, the cost of deleting a substring from a string is somewhat context sensitive, and the cost of inserting a string Z into X to obtain a string X' is equivalent to the cost of deleting Z from X' to obtain X again. As with the classical edit distance problem [5] [7] [8] [9], a slight change in the definition of the modified edit distance can remove it from the class of problems solvable in polynomial time. We show that if the minimum cost over all edit sequences must be obtained, the modified edit distance becomes undecidable. A problem of Thue [3] [4] [6] proved to be undecidable by Post [6] can be reduced to such a modified edit distance. From this point onward, such a modified edit distance shall simply be referred to as the modified edit distance.

Thue's problem is specified in Section 2. Section 3 shows that Thue's problem can be reduced to the modified edit distance, establishing that the modified edit distance is undecidable. Finally, a related open problem is given in Section 4.

2. Thue's Problem

Post defines Thue's problem as follows. A *Thue system* is $T=(\Sigma_1,P)$, where Σ_1 is a finite alphabet and $P \subseteq \{(A_i,B_i) : A_i, B_i \in \Sigma_1^*, 1 \leq i \leq n, |A_i| \leq |B_i|\}$. Two strings $\alpha, \beta \in \Sigma_1^*$ are said to be

<table>
<tr><th colspan="2">D_{XY}</th></tr>
<tr><td>$D_{XY}[m,n]=D_{XY}[m-1,n-1]+s(x_m,y_n)$</td><td rowspan="5">$1<i<m$ and $1<j<n$</td></tr>
<tr><td>$D_{XY}[i,j]=\min\{D_{XY}[i-1,j-1]+s(x_i,y_j),$
$\min_{1\le k<i} D_{XY}[k,j]+w_X(k,i),$
$\min_{1\le k<j} D_{XY}[i,k]+w_Y(k,j)\}$</td></tr>
<tr><td>$D_{XY}[i,1]=D_{XY}[1,1]+w_X(1,i)$</td></tr>
<tr><td>$D_{XY}[1,j]=D_{XY}[1,1]+w_Y(1,j)$</td></tr>
<tr><td>$D_{XY}[1,1]=s(x_1,y_1)$</td></tr>
</table>

Figure 1

similar in T if β can be obtained from α by replacing a substring A_i or B_i of α by its corresponding B_i or A_i, respectively, in P. Clearly, if α and β are similar in T, β and α are similar in T. Finally, α and β are said to be *equivalent* in T if there is a finite sequence $\gamma_1,\gamma_2, \cdots ,\gamma_m \in \Sigma_1^*$ such that α and γ_1, γ_j and γ_{j+1}, $1\le j<m$, and γ_m and β are each similar in T. *Thue's problem* is determining whether or not α and β are equivalent in T. Post proved that Thue's problem is undecidable.

3. The Reduction

Let $T=(\Sigma_1,P)$ be a Thue system. The reduction consists of constructing cost functions f_1, f_2, g, and s that simulate T; the details are given below. Now, let $\alpha,\beta\in\Sigma_1^*$. Since a prefix or suffix A_i or B_i of α may need to be replaced by its corresponding B_i or A_i, respectively, in P, let ¢ and \$ be left and right end markers, respectively, not in Σ_1. Our construction will be such that if α

and β are equivalent in T, the modified edit distance between the two strings ¢α\$ and ¢β\$ will be zero. Otherwise, the modified edit distance between ¢α\$ and ¢β\$ will be positive.

The overall strategy of the construction is to use zero cost context sensitive insertions and deletions to "pack" a substring in Σ_1^* into a supercharacter, a zero cost context free substitution to replace a supercharacter representing A_i or B_i by a supercharacter representing its corresponding B_i or A_i, respectively, in P, and zero cost context sensitive insertions and deletions to "unpack" a supercharacter. For $U,W \in \Sigma_1^*$, $a,b \in \Sigma_1$, f_1 and f_2 as given in Figure 2, and $g(1)=0$, an example of how zero cost context sensitive insertions and deletions can be used to pack a substring in Σ_1^* into a supercharacter is summarized in Figure 3. Note, however, that f_1 and f_2 as given in Figure 2 enable zero cost context free insertions and deletions of any character in Σ_1 into and from, respectively, any position of any string in Σ_1^*. To remedy this situation, let λ and ρ be local left and right end markers, respectively, not in Σ_1 and the functions υ, ι, π, and σ, which can be thought of as limited union, intersection, prefix, and suffix operations be as given in Figures 4, 5, 6, and 7, respectively. Now, let Σ be the union of all the characters, end markers, and supercharacters used thus far and f_1 and f_2 be as given in Figure 8. Then, the example summarized in Figure 3 can be replaced by the example summarized in Figure 9. The zero cost context sensitive insertions and deletions alternately insert and delete the union and intersection of characters, with boundary conditions handled by local left and right end markers.

<table>
<tr><td colspan="3">f_1 and f_2</td></tr>
<tr><td>$f_1(a,[a,b])=0$</td><td>$f_2([a,b],b)=0$</td><td rowspan="3">$a,b \in \Sigma_1$, $c \in \Sigma_1 \bigcup \{¢\}$, and $d \in \Sigma_1 \bigcup \{\$\}$</td></tr>
<tr><td>$f_1(c,a)=0$</td><td>$f_2(a,[a,b])=0$</td></tr>
<tr><td>$f_1([a,b],b)=0$</td><td>$f_2(b,d)=0$</td></tr>
</table>

Figure 2

$$¢UabW\$$$

$$¢Ua[a,b]bW\$$$

$$¢U[a,b]W\$$$

Figure 3

υ	
$\upsilon(a,b)=[a,b]$	$a,b \in \Sigma_1$
$\upsilon(\lambda,a)=[\lambda,a]$	
$\upsilon(b,\rho)=[b,\rho]$	
$\upsilon([\lambda,a],[a,b])=[\lambda,a,b]$	
$\upsilon([a,b],[b,\rho])=[a,b,\rho]$	
$\upsilon([\lambda,a,b],[a,b,\rho])=[\lambda,a,b,\rho]$	

Figure 4

ι	
$\iota([\lambda,a],[a,b])=a$	$a,b \in \Sigma_1$
$\iota([a,b],[b,\rho])=b$	
$\iota([\lambda,a,b],[a,b,\rho])=[a,b]$	

Figure 5

π	
$\pi([\lambda,a,b])=[\lambda,a]$	$a,b\in\Sigma_1$
$\pi([\lambda,a,b,\rho])=[\lambda,a,b]$	

Figure 6

σ	
$\sigma([a,b,\rho])=[b,\rho]$	$a,b\in\Sigma_1$
$\sigma([\lambda,a,b,\rho])=[a,b,\rho]$	

Figure 7

Finally, supercharacters can be packed from left to right one character at a time, for up to $l=\max\{|B_i|:(A_i,B_i)\in P\}$ characters in Σ_1^*, and zero cost context free substitutions can replace one version of a character with another version of the same character to reduce the number of cases in one of the proofs below. Formally, let $\Sigma_k=(\{\lambda\}\times\Sigma_{k-1}\times\Sigma_1\times\{\rho\})$, $1<k\leq l$, and

$$\Sigma=\bigcup_{k=1}^{l}\Sigma_k\cup\{¢,\$,\lambda,\rho\}\cup\bigcup_{k=1}^{l-1}(\Sigma_k\times\Sigma_1)\cup\bigcup_{k=1}^{l-1}(\{\lambda\}\times\Sigma_k)\cup(\Sigma_1\times\{\rho\})\cup$$

$$\bigcup_{k=1}^{l-1}(\{\lambda\}\times\Sigma_k\times\Sigma_1)\cup\bigcup_{k=1}^{l-1}(\Sigma_k\times\Sigma_1\times\{\rho\})\cup\bigcup_{k=1}^{l-1}(\Sigma_k\times\{1\})\cup(\Sigma_1\times\{2\})\cup$$

$$\bigcup_{k=1}^{l-1}(\Sigma_k\times\Sigma_1\times\{1\})\cup\bigcup_{k=1}^{l-1}(\lambda\times\Sigma_k\times\Sigma_1\times\{1\})\cup\bigcup_{k=1}^{l-1}(\Sigma_k\times\Sigma_1\times\{\rho\}\times\{1\}).$$

Now, let the functions υ, ι, π, and σ be as given in Figures 10, 11, 12, and 13, respectively, and f_1 and f_2 be as given in Figure 14, with all undefined values of f_1, f_2, and g positive. Let the functions ξ and s be as given in Figure 15 and Figure 16, respectively, with all undefined values

<table>
<tr><th colspan="3">f_1 and f_2</th></tr>
<tr><td>$f_1(a,\lambda)=0$</td><td>$f_2(\lambda,b)=0$</td><td rowspan="6">$a \in \Sigma_1 \bigcup \{¢\}$, $b \in \Sigma_1$, $c \in \Sigma_1 \bigcup \{\$\}$, and $x,y \in \Sigma$</td></tr>
<tr><td>$f_1(b,\rho)=0$</td><td>$f_2(\rho,c)=0$</td></tr>
<tr><td>$f_1(x,\upsilon(x,y))=0$</td><td>$f_2(\upsilon(x,y),y)=0$</td></tr>
<tr><td>$f_1(x,\iota(x,y))=0$</td><td>$f_2(\iota(x,y),y)=0$</td></tr>
<tr><td>$f_1(\lambda,\pi(x))=0$</td><td>$f_2(\pi(x),x)=0$</td></tr>
<tr><td>$f_1(y,\sigma(y))=0$</td><td>$f_2(\sigma(y),\rho)=0$</td></tr>
</table>

Figure 8

$$¢UabW\$$$

$$¢U\lambda a[a,b]b\rho W\$$$

$$¢U\lambda[\lambda,a]a[a,b]b[b,\rho]\rho W\$$$

$$¢U\lambda[\lambda,a][a,b][b,\rho]\rho W\$$$

$$¢U\lambda[\lambda,a][\lambda,a,b][a,b][a,b,\rho][b,\rho]\rho W\$$$

$$¢U\lambda[\lambda,a,b][a,b,\rho]\rho W\$$$

$$¢U\lambda[\lambda,a,b][\lambda,a,b,\rho][a,b,\rho]\rho W\$$$

$$¢U\lambda[\lambda,a,b,\rho]\rho W\$$$

Figure 9

of s positive.

υ	
$\upsilon(a,b)=[a,b]$	$a \in \bigcup_{k=1}^{l-1} \Sigma_k$ and $b \in \Sigma_1$
$\upsilon(\lambda,a)=[\lambda,a]$	
$\upsilon(b,\rho)=[b,\rho]$	
$\upsilon([\lambda,a],[a,b])=[\lambda,a,b]$	
$\upsilon([a,b],[b,\rho])=[a,b,\rho]$	
$\upsilon([\lambda,a,b],[a,b,\rho])=[\lambda,a,b,\rho]$	

Figure 10

ι	
$\iota([\lambda,a],[a,b])=[a,1]$	$a \in \bigcup_{k=1}^{l-1} \Sigma_k$ and $b \in \Sigma_1$
$\iota([a,b],[b,\rho])=[b,2]$	
$\iota([\lambda,a,b],[a,b,\rho])=[a,b,1]$	

Figure 11

Lemma 1. For $U,W \in \Sigma_1^*$, $a \in \Sigma_k$, $1 \le k < l$, and $b \in \Sigma_1$, the modified edit distance between $\cent U\lambda abW\$$ and $\cent U\lambda[\lambda,a,b,\rho]W\$$ is zero.

Proof. A zero cost edit sequence from $\cent U\lambda abW\$$ to $\cent U\lambda[\lambda,a,b,\rho]W\$$ is summarized in Figure 17. Therefore, the modified edit distance between $\cent U\lambda abW\$$ and $\cent U\lambda[\lambda,a,b,\rho]W\$$ is zero.□

π	
$\pi([\lambda,a,b])=[\lambda,a]$	$a \in \bigcup_{k=1}^{l-1} \Sigma_k$ and $b \in \Sigma_1$
$\pi([\lambda,a,b,\rho])=[\lambda,a,b,1]$	

Figure 12

σ	
$\sigma([a,b,\rho])=[b,\rho]$	$a \in \bigcup_{k=1}^{l-1} \Sigma_k$ and $b \in \Sigma_1$
$\sigma([\lambda,a,b,\rho])=[a,b,\rho,1]$	

Figure 13

Lemma 2. For $U,W \in \Sigma_1^*$ and $a_1,a_2, \cdots ,a_k \in \Sigma_1$, $0 \leq k \leq l$, the modified edit distance between the two strings $¢U\lambda a_1 a_2 \cdots a_k W\$$ and $¢U\lambda\xi(a_1 a_2 \cdots a_k)W\$$ is zero.

Proof. For $0 \leq k \leq 1$, the result follows directly from the definitions for the functions f_1, f_2, g, and ξ. For $2 \leq k \leq l$, the result follows from Lemma 1 and a straight forward induction.□

Lemma 3. For α and β similar in T, the modified edit distance between $¢\alpha\$$ and $¢\beta\$$ is zero.

Proof. For some $U,V_1,V_2,W \in \Sigma_1^*$, we have $\alpha=UV_1W$, $\beta=UV_2W$, and $(V_1,V_2) \in P$ or $(V_2,V_1) \in P$. Starting from $¢UV_1W\$$, insert λ between U and V_1 to obtain $¢U\lambda V_1W\$$. Then, by Lemma 2, the modified edit distance between $¢U\lambda V_1W\$$ and $¢U\lambda\xi(V_1)W\$$ is zero. Then, substitute $\xi(V_2)$ for $\xi(V_1)$ to obtain $¢U\lambda\xi(V_2)W\$$. Then, by Lemma 2, the modified edit distance between $¢U\lambda\xi(V_2)W\$$ and $¢U\lambda V_2W\$$ is zero. Finally, delete λ from between U and V_2 to obtain $¢UV_2W\$$. The cost of this edit sequence is zero. Therefore, the modified edit distance between $¢\alpha\$$ and

f_1 and f_2		
$f_1(a,\lambda)=0$	$f_2(\lambda,b)=0$	$a \in \Sigma_1 \bigcup \{¢\}$, $b \in \Sigma_1 \bigcup \{\$\}$, $c \in \bigcup_{k=1}^{l-1} \Sigma_k \bigcup \{\lambda\}$, and $x,y \in \Sigma$
$f_1(c,\rho)=0$	$f_2(\rho,b)=0$	
$f_1(x,\upsilon(x,y))=0$	$f_2(\upsilon(x,y),y)=0$	
$f_1(x,\iota(x,y))=0$	$f_2(\iota(x,y),y)=0$	
$f_1(\lambda,\pi(x))=0$	$f_2(\pi(x),x)=0$	
$f_1(y,\sigma(y))=0$	$f_2(\sigma(y),\rho)=0$	

Figure 14

ξ	
$\xi(\epsilon)=\rho$	$a,a_1,a_2,\cdots,a_k \in \Sigma_1$ and $1<k\leq l$
$\xi(a)=a$	
$\xi(a_1a_2\cdots a_k)=[\lambda,\xi(a_1a_2\cdots a_{k-1}),a_k,\rho]$	

Figure 15

$¢\beta\$$ is zero.□

Theorem 1. For α and β equivalent in T, the modified edit distance between $¢\alpha\$$ and $¢\beta\$$ is zero.

s	
$s(a,[a,1])=s([a,1],a)=0$	$a \in \bigcup_{k=1}^{l-1} \Sigma_k$, $b \in \Sigma_1$ and $(A_i,B_i) \in P$
$s(b,[b,2])=s([b,2],b)=0$	
$s([a,b],[a,b,1])=s([a,b,1],[a,b])=0$	
$s([\lambda,a,b],[\lambda,a,b,1])=s([\lambda,a,b,1],[\lambda,a,b])=0$	
$s([a,b,\rho],[a,b,\rho,1])=s([a,b,\rho,1],[a,b,\rho])=0$	
$s(\xi(A_i),\xi(B_i))=s(\xi(B_i),\xi(A_i))=0$	

Figure 16

Proof. The result follows from Lemma 3 and a straight forward induction.□

Now let the *diagram* of $X \in \Sigma^*$ be an ordered tree constructed as follows. X is the root. The children of a supercharacter x are its components in the same order they appear in x, with the exception of the components 1 and 2 which are ignored. End markers and characters in Σ_1 are leaves. See Figure 18 for the diagram of $X = ¢\lambda[\lambda,a][\lambda,a,b][a,b,\rho][b,\rho]\rho\$$, $a,b \in \Sigma_1$.

When y immediately follows x in X, let y or an ordered set of consecutive children of y, beginning with its left most child, and x or an ordered set of consecutive children of x, ending with its right most child, that are identical be called an *overlap* of x and y. Now, let a *proper* overlap of x and y be an overlap of x and y that is a proper subset of the children of either x or y. Let $\Xi(X)$ be the string of characters in Σ_1 contained in the leaves of the diagram of X in left-to-right order, with the following exception. The leaves of the maximum proper overlap of x and y are ignored in the subtree rooted at y.

$$\text{¢}U\lambda abW\$$$

$$\text{¢}U\lambda a[a,b]b\rho W\$$$

$$\text{¢}U\lambda[\lambda,a]a[a,b]b[b,\rho]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a][a,1][a,b][b,2][b,\rho]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a][a,b][b,\rho]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a][\lambda,a,b][a,b][a,b,\rho][b,\rho]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a][\lambda,a,b][a,b,1][a,b,\rho][b,\rho]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a,b][a,b,\rho]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a,b][\lambda,a,b,\rho][a,b,\rho]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a,b,1][\lambda,a,b,\rho][a,b,\rho,1]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a,b,\rho]\rho W\$$$

$$\text{¢}U\lambda[\lambda,a,b,\rho]W\$$$

Figure 17

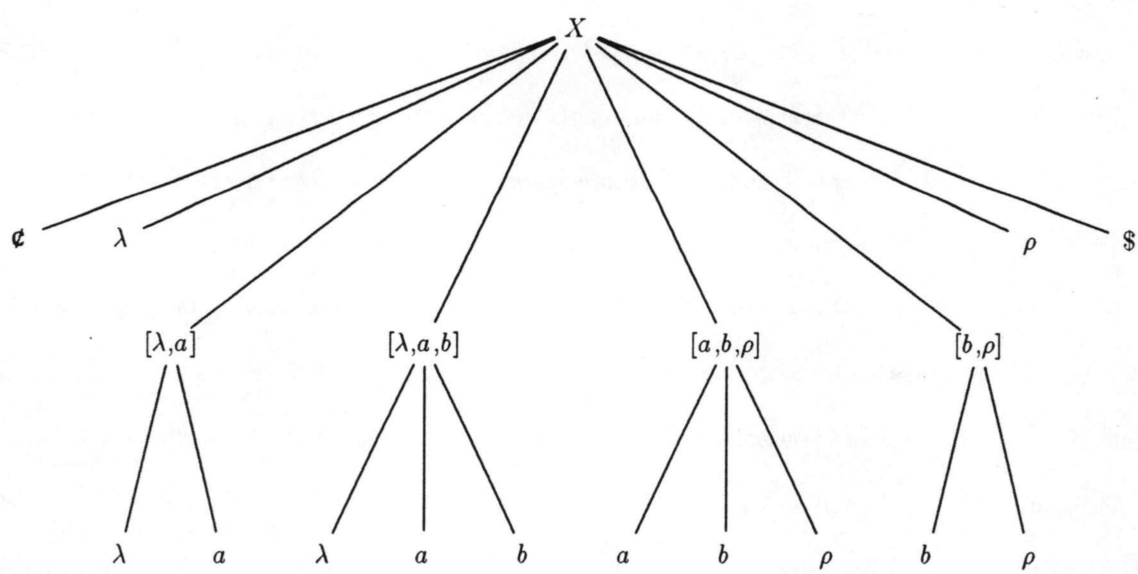

Figure 18

For the example of Figure 18, (λ,a) is the overlap of $[\lambda,a]$ and $[\lambda,a,b]$, (a,b) is the overlap of $[\lambda,a,b]$ and $[a,b,\rho]$, and (b,ρ) is the overlap of $[a,b,\rho]$ and $[b,\rho]$. Therefore, $\Xi(X)=ab$.

Theorem 2. If $\alpha,\beta\in\Sigma_1^*$ and the modified edit distance between $¢\alpha\$$ and $¢\beta\$$ is zero, α and β are equivalent in T.

Proof. The minimum cost edit sequence that changes $¢\alpha\$$ into $¢\beta\$$ consists of zero cost edit operations. Since $g(1)=0$ and $g(k)>0$, $k\neq 1$, x can be inserted or deleted from between y and z at zero cost if and only if $f_1(y,x)=f_2(x,z)=0$. For each $x\in\Sigma$, Figures 19 and 20 give the corresponding y and z values such that $f_1(y,x)=f_2(x,z)=0$. All the possible y and z combinations for each x in Figures 19 and 20 yield the zero cost insertions and deletions given in Figures 21 and 22, respectively. For $X \rightleftarrows X'$ in Figures 21 and 22, $\Xi(X)=\Xi(X')$.

<table>
<tr><td colspan="4">$f_1(y,x)=f_2(x,z)=0$, I</td></tr>
<tr><td>x</td><td>y</td><td>z</td><td rowspan="8">$a\in\Sigma_1$, $b\in\bigcup_{k=1}^{l-1}\Sigma_k$,
$c\in\Sigma_1\bigcup\{¢\}$, $d\in\Sigma_1\bigcup\{\$\}$,
and $e\in\bigcup_{k=1}^{l-1}\Sigma_k\bigcup\{\lambda\}$</td></tr>
<tr><td>a</td><td></td><td></td></tr>
<tr><td>$[\lambda,b,a,\rho]$</td><td>$[\lambda,b,a]$</td><td>$[b,a,\rho]$</td></tr>
<tr><td>$¢$</td><td></td><td></td></tr>
<tr><td>$\$$</td><td></td><td></td></tr>
<tr><td>λ</td><td>c</td><td>d</td></tr>
<tr><td>ρ</td><td>e</td><td>d</td></tr>
<tr><td>$[b,a]$</td><td>b</td><td>a</td></tr>
</table>

Figure 19

$f_1(y,x)=f_2(x,z)=0$, II			
x	y	z	
$[\lambda,a]$	λ	a	
		$[\lambda,a,b]$	
$[b,\rho]$	b	ρ	
	$[a,b,\rho]$		
$[\lambda,a,b]$	$[\lambda,a]$	$[a,b]$	$a \in \bigcup_{k=1}^{l-1} \Sigma_k$ and $b \in \Sigma_1$
$[a,b,\rho]$	$[a,b]$	$[b,\rho]$	
$[a,1]$	$[\lambda,a]$	$[a,b]$	
$[b,2]$	$[a,b]$	$[b,\rho]$	
$[a,b,1]$	$[\lambda,a,b]$	$[a,b,\rho]$	
$[\lambda,a,b,1]$	λ	$[\lambda,a,b,\rho]$	
$[a,b,\rho,1]$	$[\lambda,a,b,\rho]$	ρ	

Figure 20

The zero cost substitutions are given in Figure 23. For $X \rightleftarrows X'$ in Figure 23, $\Xi(X) \neq \Xi(X')$ if and only if $X=¢U\xi(A_i)W\$$ and $X'=¢U\xi(B_i)W\$$, where $U,W \in \Sigma^*$ and $(A_i,B_i) \in P$. If $\xi(A_i)$ and $\xi(B_i)$ do not have proper overlaps of characters in Σ_1 with the last character in U and the first character in W, then the change between $\Xi(¢U\xi(A_i)W\$)$ and $\Xi(¢U\xi(B_i)W\$)$ replaces the

<table>
<tr><th colspan="2">Zero Cost Insertions and Deletions, I</th></tr>
<tr><td>$\text{¢}U[\lambda,a,b][a,b,\rho]W\$ \rightleftarrows \text{¢}U[\lambda,a,b][\lambda,a,b,\rho][a,b,\rho]W\$$</td><td rowspan="4">$U,W \in \Sigma^*$, $a \in \bigcup_{k=1}^{l-1} \Sigma_k$,
$b \in \Sigma_1$, $c \in \Sigma_1 \bigcup \{\text{¢}\}$,
$d \in \Sigma_1 \bigcup \{\$\}$, and
$e \in \bigcup_{k=1}^{l-1} \Sigma_k \bigcup \{\lambda\}$</td></tr>
<tr><td>$\text{¢}UcdW\$ \rightleftarrows \text{¢}Uc\lambda dW\$$</td></tr>
<tr><td>$\text{¢}UedW\$ \rightleftarrows \text{¢}Ue\rho dW\$$</td></tr>
<tr><td>$\text{¢}UabW\$ \rightleftarrows \text{¢}Ua[a,b]bW\$$</td></tr>
</table>

Figure 21

substring A_i by its corresponding B_i, respectively, in P. Otherwise, for each $a \in \Sigma_k$, $1 \leq k \leq l$, Figure 24 gives the corresponding x and y values such that $\Xi(xa) \neq \Xi(x)\Xi(a)$ and $\Xi(ay) \neq \Xi(a)\Xi(y)$ after insertions, deletions, or substitutions $X \rightleftarrows X'$ such that $\Xi(X) = \Xi(X')$. For all possible a, x, and y combinations in Figure 24, there does not exist

$$b \in \bigcup_{k=1}^{l} \Sigma_k \bigcup \{\rho\} \bigcup \bigcup_{k=1}^{l-1} (\Sigma_k \times \{1\}) \bigcup (\Sigma_1 \times \{2\}),$$

and $z \in \Sigma$ such that $\Xi(b) \neq \Xi(a)$ and $f_2(x,b)=0$, $f_1(x,z)=f_2(z,b)=0$, $f_1(x,b)=0$, $f_2(b,y)=0$, $f_1(b,z)=f_2(z,y)=0$, or $f_1(b,y)=0$. Therefore, α and β are equivalent in T.□

By Theorems 1 and 2, α and β are equivalent in T, if and only if the modified edit distance between ¢α\$ and ¢β\$ is zero.

4. Open Problem

A necessary part of the above reduction is the presence of zero values for f_1, f_2, g, and $s(a,b)$, where $a \neq b$. The complexity of the modified edit distance without zero values for f_1, f_2, g, and

<table>
<tr><th colspan="2">Zero Cost Insertions and Deletions, II</th></tr>
<tr><td>$\text{¢}U\lambda aW\$\rightleftarrows\text{¢}U\lambda[\lambda,a]aW\$$</td><td rowspan="11">$U,W\in\Sigma^*$, $a\in\bigcup_{k=1}^{l-1}\Sigma_k$,
and $b\in\Sigma_1$</td></tr>
<tr><td>$\text{¢}U\lambda[\lambda,a,b]W\$\rightleftarrows\text{¢}U\lambda[\lambda,a][\lambda,a,b]W\$$</td></tr>
<tr><td>$\text{¢}Ub\rho W\$\rightleftarrows\text{¢}Ub[b,\rho]\rho W\$$</td></tr>
<tr><td>$\text{¢}U[a,b,\rho]\rho W\$\rightleftarrows\text{¢}U[a,b,\rho][b,\rho]\rho W\$$</td></tr>
<tr><td>$\text{¢}U[\lambda,a][a,b]W\$\rightleftarrows\text{¢}U[\lambda,a][\lambda,a,b][a,b]W\$$</td></tr>
<tr><td>$\text{¢}U[a,b][b,\rho]W\$\rightleftarrows\text{¢}U[a,b][a,b,\rho][b,\rho]W\$$</td></tr>
<tr><td>$\text{¢}U[\lambda,a][a,b]W\$\rightleftarrows\text{¢}U[\lambda,a][a,1][a,b]W\$$</td></tr>
<tr><td>$\text{¢}U[a,b][b,\rho]W\$\rightleftarrows\text{¢}U[a,b][b,2][b,\rho]W\$$</td></tr>
<tr><td>$\text{¢}U[\lambda,a,b][a,b,\rho]W\$\rightleftarrows\text{¢}U[\lambda,a,b][a,b,1][a,b,\rho]W\$$</td></tr>
<tr><td>$\text{¢}U\lambda[\lambda,a,b,\rho]W\$\rightleftarrows\text{¢}U\lambda[\lambda,a,b,1][\lambda,a,b,\rho]W\$$</td></tr>
<tr><td>$\text{¢}U[\lambda,a,b,\rho]\rho W\$\rightleftarrows\text{¢}U[\lambda,a,b,\rho][a,b,\rho,1]\rho W\$$</td></tr>
</table>

Figure 22

$s(a,b)$, where $a\neq b$, remains open.

Zero Cost Substitutions	
$\cent UaW\$ \rightleftarrows \cent U[a,1]W\$$	$U,W\in\Sigma^*$, $a\in\bigcup_{k=1}^{l-1}\Sigma_k$, $b\in\Sigma_1$, and $(A_i,B_i)\in P$
$\cent UbW\$ \rightleftarrows \cent U[b,2]W\$$	
$\cent U[a,b]W\$ \rightleftarrows \cent U[a,b,1]W\$$	
$\cent U[\lambda,a,b]W\$ \rightleftarrows \cent U[\lambda,a,b,1]W\$$	
$\cent U[a,b,\rho]W\$ \rightleftarrows \cent U[a,b,\rho,1]W\$$	
$\cent U\xi(A_i)W\$ \rightleftarrows \cent U\xi(B_i)W\$$	

Figure 23

Acknowledgement

The author would like to thank Professor George S. Lueker for introducing him to this problem and Professor David Eppstein for participating in early discussions of this problem.

References

1. D. Eppstein, Z. Galil, and R. Giancarlo, Speeding up dynamic programming, *Proc. 29th Annual Symposium on Foundations of Computer Science*, (1988) 488-496.
2. Z. Galil and R. Giancarlo, Speeding up dynamic programming with applications to molecular biology, *Theoret. Comput. Sci.* **64** (1989), 107-118.

<table>
<tr><td colspan="4">$\Xi(xa) \neq \Xi(x)\Xi(a)$ and $\Xi(ay) \neq \Xi(a)\Xi(y)$</td></tr>
<tr><td>a</td><td>x</td><td>y</td><td rowspan="8">$c \in \bigcup_{k=1}^{l-1} \Sigma_k$ and $b,d \in \Sigma_1$</td></tr>
<tr><td rowspan="3">b</td><td>$[c,b]$</td><td>$[b,d]$</td></tr>
<tr><td>$[\lambda,b]$</td><td>$[b,\rho]$</td></tr>
<tr><td>$[c,b,1]$</td><td>$[b,d,1]$</td></tr>
<tr><td rowspan="4">$[\lambda,c,b,\rho]$</td><td>$[\lambda,c,b]$</td><td>$[c,b,\rho]$</td></tr>
<tr><td>$[\lambda,[\lambda,c,b,\rho]]$</td><td>$[[\lambda,c,b,\rho],d]$</td></tr>
<tr><td rowspan="2">$[\lambda,c,b,1]$</td><td>$[c,b,\rho,1]$</td></tr>
<tr><td>$[[\lambda,c,b,\rho],d,1]$</td></tr>
</table>

Figure 24

3. H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, (Prentice-Hall, Englewood Cliffs, NJ, 1981).

4. H. R. Lewis and C. H. Papadimitriou, Symmetric space-bounded computation, *Theoret. Comput. Sci.* **19** (1982), 161-187.

5. D. Maier, The complexity of some problems on subsequences and supersequences, *J. ACM* **25** (1978), 322-336.

6. E. L. Post, Recursive unsolvability of a problem of Thue, *J. Symbolic Logic* **12** (1947), 1-11.

7. D. Sankoff, Minimal mutation trees of sequences, *SIAM J. Appl. Math.* **28** (1975), 35-42.

8. R. A. Wagner and M. J. Fischer, The string-to-string correction problem, *J. ACM* **21** (1974), 168-173.

9. W. J. Wilbur and D. J. Lipman, The context dependent comparison of biological sequences *SIAM J. Appl. Math.* **44** (1984), 557-567.