The undecidability of the unrestricted modified edit distance

Vitus J. Leung*

Department of Information and Computer Science, University of California, Irvine, CA 92717, USA

Received February 1996
Communicated by M. Nivat

Abstract

We define the unrestricted modified edit distance based on the modified edit distance defined by Galil and Giancarlo (1989) where the cost of substring deletions and insertions are context-sensitive and the cost of character substitutions are context-free. The modified edit distance is the minimum cost of converting a string $X$ to a string $Y$ where the sequence of edit operations has the property that all substring deletions precede all character substitutions and all character substitutions precede all substring insertions. Note that the modified edit distance does not satisfy the triangle inequality. We show that the problem of finding the unrestricted modified edit distance which is the minimum cost over all edit sequences (without these constraints) of converting $X$ to $Y$ is undecidable.

1. Introduction

This paper is motivated by problems arising in molecular biology in which it is useful to quantify the similarity of two protein strings. Examples of such measures are the minimum mutation distance between two protein sequences, first advanced by Fitch [1], and the longest common subsequence and its dynamic programming solution, first used by Needleman and Wunsch [5]. The ideas of Fitch and Needleman and Wunsch were made mathematically rigorous by Sellers [7]. Independent of Sellers, Wagner and Fischer [9] arrived at the same dynamic programming solution to the edit distance problem. Waterman et al. [10] generalized the treatment of gap weights to include gaps of more than one sequence element in length. This generalization was first shown to be useful in geology by Smith and Waterman [8] and subsequently shown to be useful in molecular biology by Fitch and Smith [3]. Galil and Giancarlo [4] further generalized the treatment of gap weights to be context-sensitive in the modified edit distance.
distance. Galil and Giancarlo define the following three types of edit operations which can be used to convert a string $X$ to a string $Y$:

- A substring $x_i x_{i+1} \ldots x_j$ can be deleted from $X = x_1 x_2 \ldots x_m$ at a cost of $f_1(x_{i-1}, x_i) + f_2(x_i, x_{i+1}) + 3(j - i + 1)$ where $1 < i \leq j < m$.
- A character $x$ in $X$ can be changed to a character $y$ at a cost of $s(x, y)$.
- A substring $y_1 y_2 \ldots y_n$ can be inserted into $X$ between $x_i$ and $x_{i+1}$ at a cost of $f'_1(x_i, y_1) + f'_2(y_n, x_{i+1}) + g'(n)$ where $1 \leq i \leq m - 1$.

Note that the cost of substring deletions and insertions is context-sensitive and the cost of character substitutions is context-free. The cost of all operations is defined by the functions: $h : \Sigma^2 \rightarrow \mathbb{R}$, $g : I^+ \rightarrow \mathbb{R}$, $s : \Sigma^2 \rightarrow \mathbb{R}$, $f'_1 : \Sigma^2 \rightarrow \mathbb{R}$ and $g' : I^+ \rightarrow \mathbb{R}$, where $i \in \{1, 2\}$, $\mathbb{R}$ is the set of nonnegative reals and $I^+$ is the set of positive integers. The cost of an edit sequence is the total cost of all its operations.

Galil and Giancarlo define the modified edit distance from $X$ to $Y$ to be the minimum cost over all edit sequences that convert $X$ to $Y$ where substrings are deleted from $X$ first, then characters in $Y$ are substituted for characters in $X$, and finally substrings from $Y$ are inserted into $X$. They use a simple recurrence relation to determine the modified edit distance from $X$ to $Y$ in polynomial time. As with the edit distance problem, 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 obtaining the minimum cost over all edit sequences without restricting the order of operations, the unrestricted modified edit distance (UMED), is undecidable. We prove this by showing that a problem of Thue proved to be undecidable by Post [6] can be reduced to the UMED.

Thue's problem is specified in Section 2. Section 3 shows that Thue's problem can be reduced to the UMED, establishing that the UMED 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 $T = (\Sigma_T, P)$, where $\Sigma_T$ is a finite alphabet and $P$ is a set of $n$ pairs of strings over $\Sigma_T$. We have

$$P = \{(A_{1,1}, A_{2,1}), (A_{1,2}, A_{2,2}), \ldots, (A_{1,n}, A_{2,n})\},$$

where $A_{i,j} \in \Sigma_T^*$ for all $1 \leq i \leq 2$ and $1 \leq j \leq n$. Two strings $\alpha, \beta \in \Sigma_T^*$ are said to be similar in $T$ if $\beta$ can be obtained from $\alpha$ by replacing a substring $A_{1,j}$ (or $A_{2,j}$) of $\alpha$ by its corresponding $A_{2,j}$ (or respectively $A_{1,j}$) in $P$. Clearly, if $\alpha$ and $\beta$ are similar in $T$, $\beta$ and $\alpha$ are also similar in $T$. Finally, $\alpha$ and $\beta$ are said to be equivalent in $T$ if there is a finite sequence $\gamma_1, \gamma_2, \ldots, \gamma_m \in \Sigma_T^*$ such that $\alpha = \gamma_1$, $\gamma_m = \beta$ and $\gamma_i$ is similar to $\gamma_{i+1}$ for all $1 \leq i \leq m - 1$. Thue's problem is to decide, given two strings $\alpha$ and $\beta$ over $\Sigma_T$, whether or not $\alpha$ and $\beta$ are equivalent in $T$. In [6] Post proved that Thue's problem is undecidable.
3. The reduction

Let \( T = (\Sigma_T, P) \) be a Thue system. The reduction consists of constructing cost functions \( f_1, f_2, g, g', \) and \( s \) that simulate \( T \). Since in our construction, all insertions are reversible (i.e. \( f_1 = f_2', g = g' \) and \( s(x, y) = s(y, x) \), where \( x, y \in \Sigma \),) we only describe \( f_1 \) and \( g \). Now, let \( \alpha, \beta \in \Sigma_* \). Our construction will be such that if \( \alpha \) and \( \beta \) are equivalent in \( T \), the UMED between the two strings \( e\alpha \$ \) and \( e\beta \$ \) will be zero. Otherwise, the UMED between \( e\alpha \$ \) and \( e\beta \$ \) will be positive. The characters \( e \) and \$ are not in \( \Sigma_T \) and are used to simulate the replacement of a suffix or prefix of \( \alpha \).

The overall strategy of the construction is to use zero cost context-sensitive insertions and deletions to "pack" a substring \( A_{i,j} \) into a supercharacter representing \( A_{i,j} \), a zero cost context-free substitution to replace a supercharacter representing \( A_{1,j} \) (or \( A_{2,j} \)) by a supercharacter representing its corresponding \( A_{2,j} \) (or respectively \( A_{1,j} \)) in \( P \), and zero cost context-sensitive insertions and deletions to unpack a supercharacter representing \( A_{i,j} \). In the description of the construction, we will fix \( i \) and \( j \) and describe how to pack \( A_{i,j} \) into the supercharacter representing \( A_{i,j} \) and how to unpack the supercharacter back to \( A_{i,j} \). The construction is presented in phases. In phase \( l \), we describe the operations which allow for intermediate characters representing substrings of length \( l + 1 \) to be inserted in between intermediate characters representing substrings of length \( l \). Then we present the operations which allow for intermediate characters representing the length \( l \) substrings to be deleted in between intermediate characters representing the length \( l + 1 \) substrings. Let \( m \) denote the length of \( A_{i,j} \). Let \( A_{i,j} = a_1a_2 \ldots a_m \). Each supercharacter representing a substring of \( A_{i,j} \) is subscripted with "\( A_{i,j} \)" to indicate that it belongs to that substring. We will use characters \( \lambda \) and \( \rho \) to delimit the packing and unpacking of \( A_{i,j} \). Thus \( \lambda \) provides the left context for zero cost deletions of intermediate characters representing prefixes of \( A_{i,j} \). Similarly for \( \rho \) and suffixes of \( A_{i,j} \).

Let \( f_1(c, \lambda) = f_2(\lambda, a_1) = g(1) = 0 \), where \( c \in \Sigma_T \cup \{e\} \). These definitions allow the character \( \lambda \) to be inserted at zero cost between a character in \( \Sigma_T \cup \{e\} \) and the first character in \( A_{i,j} \). Let \( f_1(a_m, \rho) = f_2(\rho, d) = 0 \), where \( d \in \Sigma_T \cup \{\$\} \). These definitions allow the character \( \rho \) to be inserted at zero cost between the last character in \( A_{i,j} \) and a character in \( \Sigma_T \cup \{\$\} \). Let \( f_1(a_k, [a_k, a_{k+1}]_{A_{i,j}}) = f_2([a_k, a_{k+1}]_{A_{i,j}}, a_{k+1}) = 0 \), for \( 1 \leq k \leq m - 1 \). These definitions allow the intermediate character representing a pair of adjacent characters in \( A_{i,j} \) to be inserted at zero cost between those two characters. Let \( f_1(\lambda, [\lambda, a_1]_{A_{i,j}}) = f_2([\lambda, a_1]_{A_{i,j}}, a_1) = 0 \). These definitions allow the intermediate character representing \( \lambda \) and the first character in \( A_{i,j} \) to be inserted at zero cost between those two characters. Let \( f_1(a_m, [a_m, \rho]_{A_{i,j}}) = f_2([a_m, \rho]_{A_{i,j}}, \rho) = 0 \). These definitions allow the intermediate character representing the last character in \( A_{i,j} \) and \( \rho \) to be inserted at zero cost between those two characters. Let \( f_1([a_{k-1}, a_k]_{A_{i,j}}, a_k) = f_2(a_k, [a_k, a_{k+1}]_{A_{i,j}}) = 0 \), for all \( 2 \leq k \leq m - 1 \). These definitions allow a character \( a_k \) in \( A_{i,j} \) to be deleted at zero cost from between two intermediate characters. The intermediate character to the left represents the pair of adjacent characters in \( A_{i,j} \) ending with \( a_k \) and the intermediate character to the right represents the pair of adjacent characters in \( A_{i,j} \) beginning with \( a_k \). Let \( f_1([\lambda, a_1]_{A_{i,j}}, a_1) = f_2(a_1, [a_1, a_2]_{A_{i,j}}) = 0 \). These definitions allow the
first character in $A_{i,j}$ to be deleted at zero cost from between the intermediate character representing $\lambda$ and the first character in $A_{i,j}$ and the intermediate character representing the first two characters in $A_{i,j}$. Let $f_1([a_{m-1}, a_m]_{A_{i,j}}, a_m) = f_2(a_m, [a_m, \rho]_{A_{i,j}}) = 0$. These definitions allow the last character in $A_{i,j}$ to be deleted at zero cost from between the intermediate character representing the last two characters in $A_{i,j}$ and the intermediate character representing the last character in $A_{i,j}$ and $\rho$. Now, let $U, W \in \Sigma^*$. All of these definitions allow for the following zero cost transformations:

\[
\epsilon U a_1 a_2 \cdots a_m W $\,$
\]

\[
\downarrow
\]

\[
\epsilon U a_1 [a_1, a_2]_{A_{i,j}} [a_2, a_3]_{A_{i,j}} \cdots [a_{m-1}, a_m]_{A_{i,j}} a_m \rho W $\,$
\]

\[
\downarrow
\]

\[
\epsilon U [\lambda, a_1]_{A_{i,j}} [a_1, a_2]_{A_{i,j}} [a_2, a_3]_{A_{i,j}} \cdots [a_{m-1}, a_m]_{A_{i,j}} a_m [a_m, \rho]_{A_{i,j}} \rho W $\,$
\]

\[
\downarrow
\]

\[
\epsilon U [\lambda, a_1]_{A_{i,j}} [a_1, a_2]_{A_{i,j}} [a_2, a_3]_{A_{i,j}} \cdots [a_{m-1}, a_m]_{A_{i,j}} [a_m, \rho]_{A_{i,j}} \rho W $\,$
\]

For each $l \in \{2, 3, \ldots, m - 1\}$ we have the following:

\[
f_1([a_k, a_{k+1}, \ldots, a_{k+l-1}]_{A_{i,j}}, [a_k, a_{k+1}, \ldots, a_{k+l}]_{A_{i,j}}) = f_2([a_k, a_{k+1}, \ldots, a_{k+l}]_{A_{i,j}}, [a_{k+1}, a_{k+2}, \ldots, a_{k+l}]_{A_{i,j}}) = 0
\]

for all $k \in \{1, 2, \ldots, m - l\}$. These definitions allow an intermediate character representing $l + 1$ consecutive characters in $A_{i,j}$ to be inserted at zero cost between the intermediate character representing the first $l$ of those characters and the intermediate character representing the last $l$ of those characters. For all $l \in \{2, 3, \ldots, m\}$ we have the following:

\[
f_1([\lambda, a_1, a_2, \ldots, a_{l-1}]_{A_{i,j}}, [\lambda, a_1, a_2, \ldots, a_l]_{A_{i,j}}) = f_2([\lambda, a_1, a_2, \ldots, a_l]_{A_{i,j}}, [a_1, a_2, \ldots, a_l]_{A_{i,j}}) = 0
\]

which allow the intermediate character $[\lambda, a_1, a_2, \ldots, a_l]_{A_{i,j}}$ to be inserted at zero cost in between $[\lambda, a_1, a_2, \ldots, a_{l-1}]_{A_{i,j}}$ to the left and $[a_1, a_2, \ldots, a_l]_{A_{i,j}}$ to the right. For all $l \in \{2, 3, \ldots, m\}$ we also have

\[
f_1([a_{m-l+1}, a_{m-l+2}, \ldots, a_m]_{A_{i,j}}, [a_{m-l+1}, a_{m-l+2}, \ldots, a_m, \rho]_{A_{i,j}}) = f_2([a_{m-l+1}, a_{m-l+2}, \ldots, a_m, \rho]_{A_{i,j}}, [a_{m-l+2}, a_{m-l+3}, \ldots, a_m, \rho]_{A_{i,j}}) = 0
\]

which allow the intermediate character $[a_{m-l+1}, a_{m-l+2}, \ldots, a_m, \rho]_{A_{i,j}}$ to be inserted at zero cost between $[a_{m-l+1}, a_{m-l+2}, \ldots, a_m]_{A_{i,j}}$ and $[a_{m-l+2}, a_{m-l+3}, \ldots, a_m, \rho]_{A_{i,j}}$. Now to delete the intermediate characters of length $l$, we have

\[
f_1([a_k, a_{k+1}, \ldots, a_{k+l}]_{A_{i,j}}, [a_{k+1}, a_{k+2}, \ldots, a_{k+l}]_{A_{i,j}}) = f_2([a_{k+1}, a_{k+2}, \ldots, a_{k+l}]_{A_{i,j}}, [a_{k+1}, a_{k+2}, \ldots, a_{k+l}]_{A_{i,j}}) = 0
\]
for all \( l \in \{2,3,\ldots,m-2\} \) and \( k \in \{1,2,\ldots,m-l-1\} \). These definitions allow an intermediate character representing \( l \) consecutive characters in \( A_{i,j} \) to be deleted at zero cost from between two intermediate characters. The intermediate character to the left represents the \( l+1 \) consecutive characters in \( A_{i,j} \) ending with those \( l \) characters and the intermediate character to the right represents the \( l+1 \) consecutive characters in \( A_{i,j} \) beginning with those \( l \) characters. For all \( l \in \{2,3,\ldots,m\} \) we have

\[
f_1([\lambda, a_1, a_2, \ldots, a_{l-1}]_{A_{i,j}}) = f_2([\lambda, a_1, a_2, \ldots, a_{l-1}]_{A_{i,j}}, [\lambda, a_1, a_2, \ldots, a_l]_{A_{i,j}}) = 0,
\]

and for all \( l \subset \{2,3,\ldots,m-1\} \) we have

\[
f_1([\lambda, a_1, a_2, \ldots, a_l]_{A_{i,j}}) = f_2([a_1, a_2, \ldots, a_l]_{A_{i,j}}, [a_1, a_2, \ldots, a_{l+1}]_{A_{i,j}}) = 0.
\]

Thus, we can delete at zero cost the \([\lambda, a_1, a_2, \ldots, a_{l-1}]_{A_{i,j}}\) between the \( \lambda \) on the left and the \([\lambda, a_1, a_2, \ldots, a_l]_{A_{i,j}}\) on the right, and we can drop at zero cost the \([a_1, a_2, \ldots, a_l]_{A_{i,j}}\) between the \([\lambda, a_1, a_2, \ldots, a_l]_{A_{i,j}}\) on the left and the \([a_1, a_2, \ldots, a_{l+1}]_{A_{i,j}}\) on the right. For all \( l \in \{2,3,\ldots,m\} \) we have

\[
f_1([a_{m-l+1}, a_{m-l+2}, \ldots, a_{m}, \rho]_{A_{i,j}}, [a_{m-l+2}, a_{m-l+3}, \ldots, a_{m}, \rho]_{A_{i,j}}) = 0,
\]

and for all \( l \subset \{2,3,\ldots,m-1\} \) we have

\[
f_1([a_{m-l+1}, a_{m-l+2}, \ldots, a_{m}]_{A_{i,j}}, [a_{m-l+2}, a_{m-l+3}, \ldots, a_{m}, \rho]_{A_{i,j}}) = 0.
\]

Thus, we can delete at zero cost the \([a_{m-l+2}, a_{m-l+3}, \ldots, a_{m}, \rho]_{A_{i,j}}\) between the \([a_{m-l+1}, a_{m-l+2}, \ldots, a_{m}, \rho]_{A_{i,j}}\) on the left and the \( \rho \) on the right, and we can drop at zero cost the \([a_{m-l+1}, a_{m-l+2}, \ldots, a_{m}]_{A_{i,j}}\) between the \([a_{m-l}, a_{m-l+1}, \ldots, a_{m}]_{A_{i,j}}\) on the left and the \([a_{m-l+1}, a_{m-l+2}, \ldots, a_{m}, \rho]_{A_{i,j}}\) on the right. All of these definitions allow for the following zero cost transformations for each \( l = 2,3,\ldots,m-1 \):

\[
\begin{align*}
\varepsilon U\lambda[a_1, a_2, \ldots, a_{l-1}]_{A_{i,j}} & [a_1, a_2, \ldots, a_l]_{A_{i,j}} [a_2, a_3, \ldots, a_{l+1}]_{A_{i,j}} \cdots \\
[a_{m-l+1}, a_{m-l+2}, \ldots, a_m]_{A_{i,j}} & [a_{m-l+2}, a_{m-l+3}, \ldots, a_m, \rho]_{A_{i,j}} \rho W S \\
\downarrow \\
\varepsilon U\lambda[a_1, a_2, \ldots, a_{l-1}]_{A_{i,j}} & [a_1, a_2, \ldots, a_l]_{A_{i,j}} [a_1, a_2, \ldots, a_{l+1}]_{A_{i,j}} \\
[a_2, a_3, \ldots, a_{l+1}]_{A_{i,j}} & [a_2, a_3, \ldots, a_{l+2}]_{A_{i,j}} \cdots [a_{m-l+1}, a_{m-l+2}, \ldots, a_m]_{A_{i,j}} \\
[a_{m-l+1}, a_{m-l+2}, \ldots, a_m, \rho]_{A_{i,j}} & [a_{m-l+2}, a_{m-l+3}, \ldots, a_m, \rho]_{A_{i,j}} \rho W S \\
\downarrow \\
\varepsilon U\lambda[a_1, a_2, \ldots, a_l]_{A_{i,j}} & [a_1, a_2, \ldots, a_{l+1}]_{A_{i,j}} [a_2, a_3, \ldots, a_{l+2}]_{A_{i,j}} \\
[a_{m-l}, a_{m-l+1}, \ldots, a_m]_{A_{i,j}} & [a_{m-l+1}, a_{m-l+2}, \ldots, a_m, \rho]_{A_{i,j}} \rho W S.
\end{align*}
\]
For $m \geq 2$ after all $m$ stages have been applied the string will look like $cU[\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}[a_1, a_2, \ldots, a_m]_{A_{i,j}}[\lambda, a_2, \ldots, a_m, \rho]_{A_{i,j}}\rho W$. For $m \geq 2$ we have

$$f_1([\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}, [a_1, a_2, \ldots, a_m]_{A_{i,j}}) = f_2([a_1, a_2, \ldots, a_m]_{A_{i,j}}, [a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}) = 0$$

which will allow for the $[a_1, a_2, \ldots, a_m]_{A_{i,j}}$ character to be deleted at zero cost between the $[\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}$ character on the left and the $[a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}$ character on the right. Finally, we have

$$f_1([\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}, [\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}) = f_2([a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}, [a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}) = 0$$

which allow for the supercharacter $[\lambda, a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}$ to be inserted at zero cost between $[\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}$ on the left and $[a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}$ on the right, and for $m \geq 1$ we have

$$f_1([\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}, [\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}) = f_2([a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}, [\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}) = 0$$

and

$$f_1([\lambda, a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}, [a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}) = f_2([a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}, [a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}) = 0$$

Thus, we can delete at zero cost the $[\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}$ between the $\lambda$ on the left and the $[\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}$ on the right, and we can drop at zero cost the $[a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}$ from between the $[\lambda, a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}$ on the left and the $\rho$ on the right. All of these definitions allow for the following zero cost transformations:

$$cU[\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}[a_1, a_2, \ldots, a_m]_{A_{i,j}}[\lambda, a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}\rho W$$

$$\downarrow$$

$$cU[\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}[a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}\rho W$$

$$\downarrow$$

$$cU[\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}[\lambda, a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}[a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}\rho W$$

$$\downarrow$$

$$cU[\lambda, a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}\rho W$$

We need to address the special case where $A_{i,j} = \epsilon$. Let $f_2(\lambda, d) = 0$. This definition along with the previous definition of $f_1(c, \lambda) = 0$ allows $\lambda$ to be inserted at zero cost between a character in $\Sigma_T \cup \{c\}$ and a character in $\Sigma_T \cup \{$. Let $f_1(\lambda, \rho) = 0$. This definition along with the previous definition of $f_2(\rho, d) = 0$ allows $\rho$ to be inserted at zero cost between $\lambda$ and a character in $\Sigma_T \cup \{\}$ for $A_{i,j} = \epsilon$, let $f_1(\lambda, [\lambda, \rho]_{A_{i,j}}) = f_2([\lambda, \rho]_{A_{i,j}}, \rho) = 0$. These definitions allow $[\lambda, \rho]_{A_{i,j}}$ to be inserted at zero cost between $\lambda$ and $\rho$. The supercharacter $[\lambda, \rho]_{A_{i,j}}$ represents $A_{i,j} = \epsilon$. Now, let $\Sigma$ be the union of all the characters, intermediate characters and supercharacters used thus far and $[A_{i,j}]$ be the supercharacter representing $\lambda, A_{i,j}$ and $\rho$. 
Lemma 1. The UMED between $eU_{A_{i,j}}W$ and $eU_{[A_{i,j}]}pW$ is zero.

Proof. The UMED from $eU_{A_{i,j}}W$ to $eU_{[A_{i,j}]}pW$ is zero for $0 \leq m \leq 1$ from the definitions for $f_i$, $g$, and $[A_{i,j}]$ and for $m \geq 2$ from a straightforward induction. The UMED from $eU_{[A_{i,j}]}pW$ to $eU_{A_{i,j}}W$ is zero from the fact that $f_i = f'_i$ and $g = g'$. □

Now, let $s([A_{1,j}],[A_{2,j}]) = 0$. This definition allows $[A_{1,j}]$ to be replaced at zero cost by $[A_{2,j}]$.

Lemma 2. For $\alpha$ and $\beta$ similar in $T$, the UMED between $e\alpha\$ and $e\beta\$ is zero.

Proof. For some $U, V_1, V_2, W \in \Sigma^*$, we have $\alpha = UV_1W$, $\beta = UV_2W$, and $(V_1, V_2) \in P$ or $(V_2, V_1) \in P$. By Lemma 1, the UMED between $eU_{[V_1]}W$ and $eU_{[V_1]}pW$ is zero. By the definition for $s$, the UMED between $eU_{[V_1]}pW$ and $eU_{[V_2]}pW$ is zero. By Lemma 1, the UMED between $eU_{[V_2]}pW$ and $eUV_2W$ is zero. Therefore, the UMED between $e\alpha\$ and $e\beta\$ is zero. □

Theorem 1. For $\alpha$ and $\beta$ equivalent in $T$, the UMED between $e\alpha\$ and $e\beta\$ is zero.

Proof. The result follows from Lemma 2 and a straightforward induction. □

Now let all undefined values of $f_i, g$ and $s$ be positive. $X, Y \in \Sigma^*$ are edit-equivalent $X \equiv_e Y$ if $Y$ can be obtained from $X$ via zero cost insertions and deletions (no substitutions).

Lemma 3. $\equiv_e$ is an equivalence relation.

Proof. $X \equiv_e X$ from the definition of $\equiv_e$. If $X \equiv_e Y$, then $Y \equiv_e X$ from the fact that $f_i = f'_i$ and $g = g'$. If $X \equiv_e Y$ and $Y \equiv_e Z$, then $X \equiv_e Z$ from the definition of $\equiv_e$. Therefore, $\equiv_e$ is an equivalence relation. □

Let $X$ be elementary if $X \in (\Sigma_T \cup \{e, \$\})^*$ and let the components of $x \in \Sigma$ be the string in $(\Sigma_T \cup \{e, \$\})^*$ that $x$ represents. When $y$ immediately follows $x$ in $X$, let a prefix of the components of $y$ and a suffix of the components of $x$ 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 not equal to both the components of $x$ and the components of $y$. Let $\mathcal{E}(X)$ be the concatenation of the components of the characters in $X$, with the following exceptions. When $y$ immediately follows $x$ in $X$, the maximum proper overlap of $x$ and $y$ are excluded from the components of $y$, and for all $x$ in $X$, $\lambda$ and $\rho$ are excluded from the components of $x$. For

$$X = eU_{[A_{i,j}, \lambda, a_1, a_2, \ldots, a_{m-1}], \lambda, a_1, a_2, \ldots, a_m, \rho}_{A_{i,j}}$$
$$[a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}[a_2, a_3, \ldots, a_m, \rho]_{A_{i,j}}\rhoWS,$$
\(\lambda a_1 a_2 \cdots a_{m-1}\) is the overlap of \([\lambda, a_1, a_2, \ldots, a_{m-1}]_{A_{i,j}}\) and \([\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}\),
\(a_1 a_2 \cdots a_m\) is the overlap of \([\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}\) and \([a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}\) and \(a_2 a_3 \cdots a_m \rho\) is the overlap of \([a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}\) and \([a_2, a_3, \ldots, a_m, \rho]_{A_{i,j}}\). Therefore, \(\Xi(X) = \varnothing U A_{i,j} W S\).

Lemma 4. In each equivalence class there is at most one elementary string.

Proof. Since \(g(1) = 0\) and \(g(k) > 0\) for \(k \geq 2\), the only zero cost insertions and deletions are single character insertions and deletions. A character \(y\) can be inserted or deleted between \(x\) and \(z\) at zero cost if and only if \(f_1(x, y) = f_2(y, z) = 0\). For each \(y\), Figs. 1–4 give the corresponding \(x\) and \(z\) values such that \(f_1(x, y) = f_2(y, z) = 0\). In Figs. 1–4, all possible characters \(y\) are grouped into 24 cases. Consider the case of the intermediate character \([a_1, a_2]_{A_{i,j}}\). For \(m \geq 3\), this intermediate character can be inserted to the right of the character \(a_1\) or the intermediate character \([a_2, a_3]_{A_{i,j}}\) and to the left of the character \(a_2\) or the intermediate character \([a_1, a_2, a_3]_{A_{i,j}}\), resulting in four subcases.

For all combinations of \(x, y\) and \(z\) in Figs. 1–4, \(\Xi(xyz) = \Xi(xz)\). Therefore, if \(X \equiv y Y\), then \(\Xi(X) = \Xi(Y)\). Therefore, in each equivalence class there is at most one elementary string. \(\Box\)

Theorem 2. If the UMED between \(\varepsilon aS\) and \(\varepsilon \beta S\) is zero, \(\alpha\) and \(\beta\) are equivalent in \(T\).

Proof. The minimum cost edit sequence that changes \(\varepsilon aS\) into \(\varepsilon \beta S\) consists of zero cost edit operations. The only zero cost operation which can change \(X\) to \(Y\) so that \(X\) and \(Y\) are not in the same equivalence class is replacing \([A_{i,j}]\) (or \([A_{2,j}]\)) with \([A_{2,j}]\) (or respectively \([A_{1,j}]\)). Without loss of generality, we consider the case where \([A_{1,j}]\) is replaced by \([A_{2,j}]\). If \([A_{1,j}]\) and \([A_{2,j}]\) are preceded by \(\lambda\) and followed by \(\rho\) in \(X\) and \(Y\), there is an elementary string in both equivalence classes. Otherwise, \([\lambda, a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}\) is replaced by \([A_{2,j}]\) before \([\lambda, a_1, a_2, \ldots, a_m]_{A_{i,j}}\), \([\lambda, a_1, a_2, \ldots, a_{m-1}]_{A_{i,j}}\), \([a_1, a_2, \ldots, a_m, \rho]_{A_{i,j}}\) or \([a_2, a_3, \ldots, a_m, \rho]_{A_{i,j}}\) are deleted. These characters cannot be involved in any zero cost edit operations with any characters before \(\lambda\) or after \(\rho\) until the replacement of \([A_{1,j}]\) with \([A_{2,j}]\) has been reversed.

Therefore, every replacement of \([A_{1,j}]\) (or \([A_{2,j}]\)) with \([A_{2,j}]\) (or respectively \([A_{1,j}]\)) gives a transition in the Thue system, and \(\alpha\) and \(\beta\) are equivalent in \(T\). \(\Box\)

By Theorems 1 and 2, \(\alpha\) and \(\beta\) are equivalent in \(T\), if and only if the UMED between \(\varepsilon aS\) and \(\varepsilon \beta S\) is zero.

4. Open problem

A necessary part of the above reduction is the presence of zero values for \(f_1, f'_1, g, g'\) and \(s(x, y)\), where \(x \neq y\). Therefore, the UMED does not satisfy the axioms of a metric. However, the UMED must satisfy the axioms of a metric in order to be useful in the construction of an evolutionary tree by the method of Fitch and Margoliash [2]. The
<table>
<thead>
<tr>
<th></th>
<th>(f_1(x,y) = f_2(y,z) = \emptyset, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x)</td>
<td>(y)</td>
</tr>
<tr>
<td>(c)</td>
<td>(\lambda)</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>(\rho)</td>
</tr>
<tr>
<td>(a_1)</td>
<td>([a_1,a_2]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([\lambda,a_1,a_2]<em>{A</em>{ij}})</td>
<td>([a_1,a_2,a_3]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>(a_k)</td>
<td>([a_k,a_k+1]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([a_{k-1},a_k,a_{k+1}]<em>{A</em>{ij}})</td>
<td>([a_k,a_k+1,a_{k+1}]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>(a_{m-1})</td>
<td>([a_{m-1},a_m]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([a_{m-2},a_{m-1},a_m]<em>{A</em>{ij}})</td>
<td>([a_{m-1},a_m,\rho]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>(a_1)</td>
<td>([a_1,a_2]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([\lambda,a_1,a_2]<em>{A</em>{ij}})</td>
<td>([a_1,a_2,\rho]<em>{A</em>{ij}})</td>
</tr>
</tbody>
</table>

Fig. 1.
\[ f_1(x, y) = f_2(y, z) = 0, \text{ II} \]

<table>
<thead>
<tr>
<th></th>
<th>(x)</th>
<th>(y)</th>
<th>(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\lambda)</td>
<td>([\lambda, a_1]<em>{A</em>{i,j}})</td>
<td>(a_1)</td>
<td>([\lambda, a_1, a_2]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>([\lambda, a_2]<em>{A</em>{i,j}})</td>
<td>(a_1)</td>
<td>([\lambda, a_1, \rho]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>(a_m)</td>
<td>([a_m, \rho]<em>{A</em>{i,j}})</td>
<td>(\rho)</td>
<td>([a_{m-1}, a_m, \rho]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>([a_{m-1}, a_m, \rho]<em>{A</em>{i,j}})</td>
<td>([a_{m-1}, a_m, \rho]<em>{A</em>{i,j}})</td>
<td>(\rho)</td>
<td>([a_{m-1}, a_m, \rho]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>([a_{k-1}, a_k]<em>{A</em>{i,j}})</td>
<td>(a_k)</td>
<td>([a_{k-1}, a_k]<em>{A</em>{i,j}})</td>
<td>([a_{k-1}, a_k]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>([\lambda, a_1]<em>{A</em>{i,j}})</td>
<td>(a_1)</td>
<td>([a_1, a_2]<em>{A</em>{i,j}})</td>
<td>([a_1, a_2]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>([a_{m-1}, a_m]<em>{A</em>{i,j}})</td>
<td>(a_m)</td>
<td>([a_{m-1}, a_m]<em>{A</em>{i,j}})</td>
<td>([a_{m-1}, a_m]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>([\lambda, a_1]<em>{A</em>{i,j}})</td>
<td>(a_1)</td>
<td>([a_1, \rho]<em>{A</em>{i,j}})</td>
<td>([a_1, \rho]<em>{A</em>{i,j}})</td>
</tr>
</tbody>
</table>

Fig. 2.
\[ f_1(x,y) = f_2(y,x) = 0, \text{ III} \]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>([a_1, a_2, \ldots, a_{i-1}]<em>{A</em>{ij}})</td>
<td>([a_1, a_2, \ldots, a_l]<em>{A</em>{ij}})</td>
<td>([a_2, a_3, \ldots, a_l]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([\lambda, a_1, a_2, \ldots, a_l]<em>{A</em>{ij}})</td>
<td>([a_1, a_2, \ldots, a_{i+1}]<em>{A</em>{ij}})</td>
<td>([a_1, a_2, \ldots, a_{l+1}]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([a_k, a_{k+1}, \ldots, a_{k+l-1}]<em>{A</em>{ij}})</td>
<td>([a_k, a_{k+1}, \ldots, a_{k+l}]<em>{A</em>{ij}})</td>
<td>([a_{k+l}, a_{k+2}, \ldots, a_{k+l+1}]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([a_{k-l}, a_{k-l+1}, \ldots, a_{k+l}]<em>{A</em>{ij}})</td>
<td>([a_{k-l}, a_{k-l+2}, \ldots, a_{k+l}]<em>{A</em>{ij}})</td>
<td>([a_{k-l+1}, a_{k-l+2}, \ldots, a_{k+l}]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([a_{k-l}, a_{m-l+1}, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([a_{m-l+1}, a_{m-l+2}, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([a_{m-l+1}, a_{m-l+2}, \ldots, a_m]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([a_{m-l}, a_{m-l+1}, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([a_{m-l+1}, a_{m-l+2}, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([a_{m-l+1}, a_{m-l+2}, \ldots, a_m]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([\lambda, a_1, a_2, \ldots, a_{m-1}]<em>{A</em>{ij}})</td>
<td>([a_1, a_2, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([a_3, a_3, \ldots, a_m]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([\lambda, a_1, a_2, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([a_1, a_2, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([a_1, a_2, \ldots, a_m]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([\lambda, a_1, a_2, \ldots, a_{l-1}]<em>{A</em>{ij}})</td>
<td>([\lambda, a_1, a_2, \ldots, a_l]<em>{A</em>{ij}})</td>
<td>([a_1, a_2, \ldots, a_l]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>([\lambda, a_1, a_2, \ldots, a_{l+1}]<em>{A</em>{ij}})</td>
<td>([\lambda, a_1, a_2, \ldots, a_{l+1}]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>([\lambda, a_1, a_2, \ldots, a_{m-1}]<em>{A</em>{ij}})</td>
<td>([\lambda, a_1, a_2, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([a_1, a_2, \ldots, a_m]<em>{A</em>{ij}})</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>([\lambda, a_1, a_2, \ldots, a_m]<em>{A</em>{ij}})</td>
<td>([\lambda, a_1, a_2, \ldots, a_m]<em>{A</em>{ij}})</td>
</tr>
</tbody>
</table>

Fig. 3.
\[ f_1(x,y) = f_2(y,z) = 0, \text{ IV} \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x)</td>
<td>(y)</td>
<td>(z)</td>
</tr>
<tr>
<td>([a_{m-l}, a_{m-l+1}, \ldots, a_{m}]<em>{A</em>{i,j}})</td>
<td>([a_{m-l}, a_{m-l+1}, \ldots, a_{m}, \rho]<em>{A</em>{i,j}})</td>
<td>([a_{m-l+1}, a_{m-l+2}, \ldots, a_{m}, \rho]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>([a_{m-l-1}, a_{m-l}, \ldots, a_{m}, \rho]<em>{A</em>{i,j}})</td>
<td>(\rho)</td>
<td></td>
</tr>
<tr>
<td>([a_1, a_2, \ldots, a_m]<em>{A</em>{i,j}})</td>
<td>([a_1, a_2, \ldots, a_m, \rho]<em>{A</em>{i,j}})</td>
<td>([a_2, a_3, \ldots, a_m, \rho]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>([\lambda, a_1, a_2, \ldots, a_m, \rho]<em>{A</em>{i,j}})</td>
<td>(\rho)</td>
<td></td>
</tr>
<tr>
<td>([\lambda, a_1, a_2, \ldots, a_m]<em>{A</em>{i,j}})</td>
<td>([\lambda, a_1, a_2, \ldots, a_m, \rho]<em>{A</em>{i,j}})</td>
<td>([a_1, a_2, \ldots, a_m, \rho]<em>{A</em>{i,j}})</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>([\lambda, \rho]<em>{A</em>{i,j}})</td>
<td>(\rho)</td>
</tr>
</tbody>
</table>

Fig. 4.

UMED without zero values for \(f_i, f_i', g, g'\) and \(s(x, y)\), where \(x \neq y\), is decidable, but its complexity remains open.

Acknowledgements

The author would like to thank George Lueker for the introduction to this problem, David Eppstein for participation in early discussions on this problem and Sandy Irani for suggestions which greatly improved the presentation of the material in this paper.

References


