

The Structure of Forward, Reverse, and Transverse Path Graphs in The Pattern Recognition Algorithms of Sellers

Lewis Lasser

Louis D'Alotto

Department of Mathematics and Computer Science
York College/CUNY
Jamaica, New York 11451
llasser@york.cuny.edu

Department of Mathematics and Computer Science
York College/CUNY
Jamaica, New York 11451
dalotto@york.cuny.edu

Abstract—In [3], [4], [5] Sellers develops a dynamic programming pattern matching algorithm that generates forward, reverse, and transverse path graphs that determine the best resemblance (lowest cost) of a smaller string pattern inside a larger. In this paper we study the properties and structure of these graphs. We show that these path graphs can be decomposed into a small number of distinct block types, that are used to analyze graph structure. It is also shown that an exact pattern match results in a disconnected transverse path graph.

keywords: Pattern Recognition, Connected Graph, Path Graph, Block Type, Evolutionary Distance.

I. INTRODUCTION

The pattern recognition problem of finding an interval \mathbf{i} in a large sequence, $\mathbf{a} = a_1a_2\dots a_n$, that best resembles a smaller sequence, $\mathbf{b} = b_1b_2\dots b_m$, was studied by Sellers, see [3], [4], [5]. In his work he described a dynamic programming pattern recognition algorithm that searches for a best resemblance (lowest cost).

String \mathbf{a} resembles pattern \mathbf{b} if there is an interval in \mathbf{a} that is either equal to \mathbf{b} or close enough to \mathbf{b} . The term *close enough* depends on the choice of a suitable metric. The notation $d(x, y)$, where x and y are arbitrary strings, will be used to represent such a metric.

The function $d(x, y)$ is called the *evolutionary distance*. The evolutionary distance is the length of the shortest path (lowest cost) of evolutionary steps between \mathbf{a} and \mathbf{b} . We assume, as in [3], that only certain types of evolutionary steps are allowed for strings (sequences) and that a cost can be associated with each step. We consider an evolutionary step as an *insertion*, *deletion*, or *mutation* of a single element in a sequence. We assign each such step a cost of

1. More formally, the evolutionary distance $d(\mathbf{a}, \mathbf{b})$ between any two sequences is defined as

$$\min \left\{ \sum_{i=1}^{m+n} d(a_i, b_i) \right\}$$

Where the minimum is taken over all pairs of sequences

$$a_1a_2\dots a_{n+m}$$

and

$$b_1b_2\dots b_{n+m}$$

with m null symbols (-) inserted in sequence \mathbf{a} and n null symbols inserted in sequence \mathbf{b} . The null symbols represent the deletion or insertion of a term with which it is aligned in the other sequence. This produces a metric alignment of sequences \mathbf{a} and \mathbf{b} . Thus the evolutionary distance is the lowest cost of evolutionary steps between \mathbf{a} and \mathbf{b} .

The algorithm of Sellers is described as follows. Given two finite sequences, $\mathbf{a} = a_1a_2\dots a_n$ and $\mathbf{b} = b_1b_2\dots b_m$, where $n > m$ and \mathbf{b} is the specified string pattern being sought in \mathbf{a} . The algorithm searches to identify intervals \mathbf{i} in \mathbf{a} that resemble \mathbf{b} and calculates their evolutionary distance from string \mathbf{b} . We use the notation $\mathbf{i} \subset \mathbf{a}$ to say that \mathbf{i} is an interval in \mathbf{a} . Hence $\mathbf{i} = a_p a_{p+1} \dots a_q$, where $1 \leq p \leq q \leq n$.

$D_f = (e(i, j))$ is called the $(n + 1) \times (m + 1)$ *forward distance matrix* and is computed inductively. To start constructing the matrix, the initial values of the first column are set to zero. That is, set $e(i, 0) = 0$ for $i = 0, 1, \dots, n$. Continuing, the values in the first row are set

$$e(0, j) = \sum_{h=1}^j d(-, b_h)$$

for $j = 1, 2, \dots, m$.

The rest of the forward matrix values are computed inductively by assigning the smallest of these three values:

$$\begin{aligned} & (e(i-1, j) + d(a_i, -)) \\ & (e(i-1, j-1) + d(a_i, b_j)) \\ & (e(i, j-1) + d(-, b_j)) \end{aligned}$$

to $e(i, j)$. These three values determine the forward neighborhood, see [1]. Note that $e(i, j)$ is the minimum total cost of the mutations, insertions, and deletions needed to transform the first string segment $b_1 b_2 \dots b_j$ of \mathbf{b} into any interval $\mathbf{i} \subset \mathbf{a}$ of any length that ends at a_i .

The *reverse distance matrix* $D_r = (f(k, j))$ is constructed in a similar manner, however we start with the last terms of \mathbf{a} and \mathbf{b} and end with the first. Here we note that $f(k, j)$ is the minimum total cost of mutations, insertions, and deletions necessary to convert the string segment $b_k b_{k+1} \dots b_m$ into an interval $\mathbf{i} \subset \mathbf{a}$. The initial values of the matrix entries in the last column and row are $f(i, m+1) = 0$ for $i = 1, 2, \dots, n+1$ and

$$f(n+1, j) = \sum_{h=0}^{m-j} d(-, b_{m-h})$$

for $j = 1, 2, \dots, m$. The rest of the reverse matrix values are computed inductively by setting $f(i, j)$ equal to the smallest of these three values in the reverse neighborhood:

$$\begin{aligned} & (f(i+1, j) + d(a_i, -)) \\ & (f(i+1, j+1) + d(a_i, b_j)) \end{aligned}$$

and

$$(f(i, j+1) + d(-, b_j))$$

The algorithm determines both global and local resemblances. The above procedure determines all local resemblances. Looking only at the forward matrix determines only global resemblances. Given the evolutionary distance d , an interval \mathbf{i} in \mathbf{a} best resembles a short sequence \mathbf{b} *globally* iff $d(\mathbf{i}, \mathbf{b}) \leq d(\mathbf{j}, \mathbf{b})$ for all intervals \mathbf{j} in \mathbf{a} . Also, given an evolutionary distance d , an interval \mathbf{i} in \mathbf{a} best resembles \mathbf{b} *locally* iff $d(\mathbf{i}, \mathbf{b}) \leq d(\mathbf{h}, \mathbf{b})$ and $d(\mathbf{i}, \mathbf{b}) \leq d(\mathbf{j}, \mathbf{b})$ for all \mathbf{h} and \mathbf{j} , where $\mathbf{h} \subset \mathbf{i} \subset \mathbf{j} \subset \mathbf{a}$.

The *forward path graph* is constructed directly from D_f by connecting each pair of matrix entries, that are connected by an inductive step, with an edge and replacing each matrix entry with a vertex. Similarly, the *reverse path graph* is constructed from D_r . The *common path graph* is the intersection of vertices and

edges from both graphs. The *transverse path* is a succession of edges that connects the first column and the last column of a path graph. Global resemblances can be distinguished from local by examining the values of the last column in the matrix associated with the vertices of the transverse path graph. A minimal value corresponds to a global resemblance while any larger values correspond to local resemblances. If the transverse path graph is connected then all vertices in the last column of the associated matrix must have the same value. Throughout this paper we consider a graph connected if there is exactly one connected component after we disregard vertices of degree zero. A local match (resemblance) only occurs when a value larger than the minimal value is encountered. If only one value exists all matches must be global. Hence the graph property of being connected corresponds to all matches being global.

For $i \geq 1$ and $j \geq 1$, the *forward neighborhood* of vertex $e(i, j)$ is the set of three edges $e(i-1, j)$, $e(i-1, j-1)$ and $e(i, j-1)$. For $i = 0$ and $j \geq 1$ it consists of the single vertex $e(0, j-1)$ while for $j = 0$ and $i \geq 1$ it is just $e(i-1, 0)$. The vertex $e(0, 0)$ has empty neighborhood. The *forward neighbors* of a vertex $e(i, j)$ are those vertices in its forward neighborhood to which it is connected. For $i \leq n$ and $j \leq m$, the *reverse neighborhood* of vertex $f(i, j)$ is the set of three edges $f(i+1, j)$, $f(i+1, j+1)$ and $f(i, j+1)$. For $i = n+1$ and $j \leq m$ it consists of the single vertex $f(n+1, j+1)$ while for $j = m+1$ and $i \leq n$ it is just $f(i+1, m+1)$. The vertex $f(n+1, m+1)$ has empty neighborhood. The *reverse neighbors* of a vertex $f(i, j)$ are those vertices in its reverse neighborhood to which it is connected.

The lemmas and theorems presented in this paper demonstrate that the forward, reverse, common, and transverse path graphs can be decomposed into various block types. It is further interesting to see, that a good pattern match results in a disconnected transverse path graph.

\mathbf{b} will always refer to the specified string and \mathbf{a} for the longer string in which we seek substrings locally and/or globally similar to \mathbf{b} . The set Σ will consist of all of our symbols together with the special null symbol $-$. We use the evolutionary distance $d(x, y) = 0$ if $x = y$ and 1 otherwise, for all $x, y \in \Sigma$. The value of 0 implies a direct match, while the value (cost) of 1 corresponds to a mutation, deletion, or insertion of a single term in the sequence. If $a \in \Sigma$ then a^n will mean a string of n consecutive a 's. The *interior* of the forward graph/matrix refers to all but the first row and column. For the reverse graph/matrix, the interior is all but the last row and column. The proofs of the results

depend on the analysis of the different “block types” of the forward and reverse matrices and their corresponding graphs. Where no confusion arises, we will use the term *block* to represent rectangular portions of the matrices as well as the path graphs.

In the illustrations that follow, it is frequently necessary to show the elements in either the long string **a** or the short string **b** that correspond to the portion of the matrix/graph being shown. These are indicated in one or more vertical lines along the extreme left margin (for the whole or the portion of the long string) or in one or more horizontal lines along the extreme top (for the whole or the portion of the short string). A star indicates an arbitrary symbol of the alphabet different from *a*.

The algorithm was implemented to generate and display graphically the forward, reverse, common, and transverse path graphs. The program was written in the C programming language and run under Mac OS X and Linux with the GD graphics library installed.

II. THE STRUCTURE OF FORWARD AND REVERSE PATH GRAPHS

There are eight fundamental “building blocks” from which the forward and reverse distance matrices and their respective path graphs are constructed (four for the forward and four for the reverse). The following lemmas describe each type for the forward matrix/graph case. Each has a counterpart for the reverse case.

Lemma 1: Suppose the rectangular portion of the forward distance matrix between rows i_1 and i_2 and columns j_1 and j_2 has every entry in the first column equal to p and entries in the first row, from left to right, equal to $p, p+1, \dots, p+s$, where $s = j_2 - j_1$. Suppose $a_i = a$ for $i_1 + 1 \leq i \leq i_2$ and $b_j \neq a$ for $j_1 + 1 \leq j \leq j_2$. Then every entry in the last column has value $p+s$ and the entries in the last row, from left to right, are $p, p+1, \dots, p+s$. Furthermore every vertex in the interior of the corresponding rectangular portion of the forward path graph is connected to exactly two neighbors, the one to its left and the one diagonally up and left.

Proof of Lemma 1: Figure 1 illustrates this lemma for $s = 4$ and $i_2 - i_1 = 8$. The values in the matrix are, as always, computed in an iterative fashion. First the missing values in the second row, from left to right, are found. This procedure is repeated for each row, from top to bottom. When a missing value $e(i, j)$ is computed we have $d(a_i, b_j) = 1$ because $a_i \neq b_j$. Furthermore it is easily seen that the values to its left and diagonally up and left are equal while the value above is one greater. Consequently, due to Seller’s construction, $e(i, j)$ is one more than the value to its

left and the forward path graph has the indicated edges incident with the corresponding vertex.

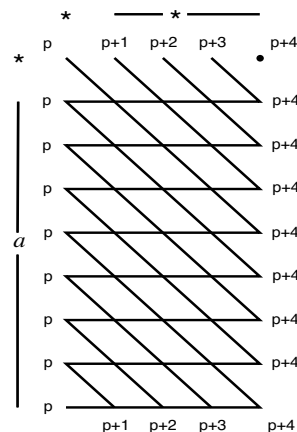


Fig. 1. Block Type 1 (BT1)

Lemma 2: Suppose the rectangular portion of the forward distance matrix between rows i_1 and i_2 and columns j_1 and j_2 has entries in the first column, from bottom to top, of $p, p+1, \dots, p+r$, where $r = i_2 - i_1$ and entries in the first row, from left to right, equal to $p+r, p+r+1, p+r+2, \dots, p+r+s$, where $s = j_2 - j_1$. Suppose $a_i = a$ for $i_1 + 1 \leq i \leq i_2$ and $b_j \neq a$ for $j_1 + 1 \leq j \leq j_2$. Then the entries in the last row, from left to right, are $p, p+1, \dots, p+s$ and the entries in the last column, from bottom to top, are $p+s, p+s+1, p+s+2, \dots, p+r+s$. Furthermore every vertex in the interior of the corresponding rectangular portion of the forward path graph is connected only to the neighbor to its left.

Proof of Lemma 2: Figure 2 illustrates this lemma for $r = s = 4$. The proof is a straightforward consequence of the construction.

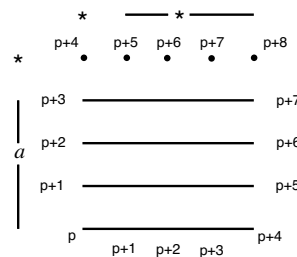


Fig. 2. Block Type 2 (BT2)

Lemma 3: Suppose the rectangular portion of the forward distance matrix between rows i_1 and i_2 and columns j_1 and j_2 has entries in the first column, from bottom to top, of $p, p+1, \dots, p+r$, where $r = i_2 - i_1$ and entries in the first row, from left to right, equal to $p+r, p+r+1, p+r+2, \dots, p+r+s$, where $s = j_2 - j_1$.

Suppose $a_i = a$ for $i_1 + 1 \leq i \leq i_2$ and $b_j = a$ for $j_1 + 1 \leq j \leq j_2$. Then the entries in the last row, from left to right, are $p, p + 1, \dots, p + s$ and the entries in the last column, from bottom to top, are $p + s, p + s + 1, p + s + 2, \dots, p + r + s$. Furthermore every vertex in the interior of the corresponding rectangular portion of the forward path graph is connected to exactly two neighbors: the neighbor to its left and diagonally up and left.

Proof of Lemma 3: Figure 3 illustrates this lemma for $r = s = 4$.

Again, this is a straightforward consequence of the way in which the matrix and graph are constructed.

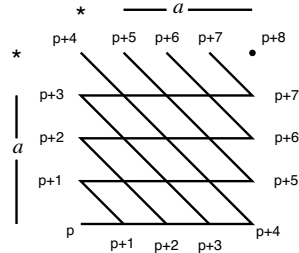


Fig. 3. Block Type 3 (BT3)

Lemma 4: Suppose the rectangular portion of the forward distance matrix between rows i_1 and i_2 and columns j_1 and j_2 has every entry in the first column equal to p and entries in the first row, from left to right, equal to $p, p + 1, \dots, p + s$, where $s = j_2 - j_1$. Suppose $a_i = a$ for $i_1 + 1 \leq i \leq i_2$ and $b_j = a$ for $j_1 + 1 \leq j \leq j_2$. Then every entry in the last row has value p and the entries in the last column, from bottom to top, consist of $r - s$ values of p followed by $p + 1, p + 2, \dots, p + s$. The edges of a vertex in the interior of the corresponding rectangular portion of the forward path graph depend on the relative row and column numbers. If $j - j_1 \geq i - i_1$ then the vertex is connected to exactly two neighbors: the one to its left and the one diagonally up and left. Otherwise the vertex is connected to the neighbor diagonally up and left.

Proof of Lemma 4: Figure 4 illustrates this lemma for $r = 8$ and $s = 4$.

Although more complex than lemmas 1 through 3 it is proved in the same fashion.

For future reference we refer to both the portion of the matrix and the portion of the graph in lemmas 1 through 4 as being of *block type one* through *four*, respectively. For graphs of block type four there is an area in the upper right corner that is connected with a number of disjoint paths running on diagonals below. The vertex where the triangular region intersects the last column has a matrix value of p . We call this vertex

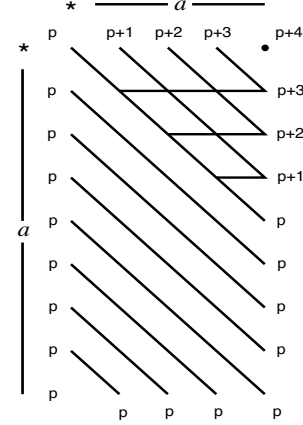


Fig. 4. Block Type 4 (BT4)

the *transition vertex* for a graph of block type four. All matrix values below the transition vertex have value p and those above, from bottom to top, have values $p + 1, p + 2, \dots, p + s$.

We now describe all possible relations between the four block types that can occur in a forward or reverse path graph. The next lemma shows that to the right of every block type one is a block type four. This is shown in Figure 5.

Lemma 5: Suppose a portion P_1 of the forward matrix/graph is of block type one. Suppose the values of the matrix in the first row, from left to right, are $p, p + 1, \dots, p + s$ and that this row of the matrix extends to the right with values $p + s + 1, p + s + 2, \dots, p + s + t$. Suppose every element of the short sequence \mathbf{b} above these extended matrix values is a . Then there is a portion P_2 of the forward matrix/graph of block type four that shares every matrix-value/vertex of the rightmost column of P_1 .

Proof of Lemma 5: Since P_1 is of block type one, the values in the matrix for P_1 in the rightmost column, from bottom to top, are all $p + s$. We now have a portion of the matrix (to the right of P_1) whose first column contains only the value $p + s$ and whose first row, from left to right, contains the values $p + s, p + s + 1, \dots, p + s + t$. Since all elements of the short sequence s above all but the first column are equal to a , lemma 4 identifies this portion of the forward matrix/graph to be of block type four.

The next lemma shows that to the right of every block type four are blocks of type two and one. This is shown in figure 6.

Lemma 6: Suppose a portion P_1 of the forward matrix/graph is of block type four. Suppose the values of the matrix in the first row, from left to right, are $p, p + 1, \dots, p + s$ and that this row of the matrix

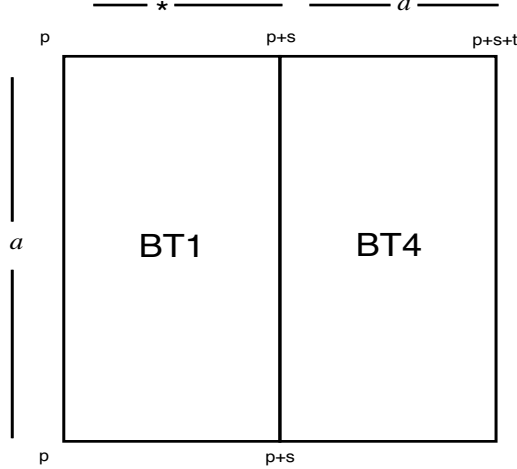


Fig. 5. BT1 with BT4 to its right

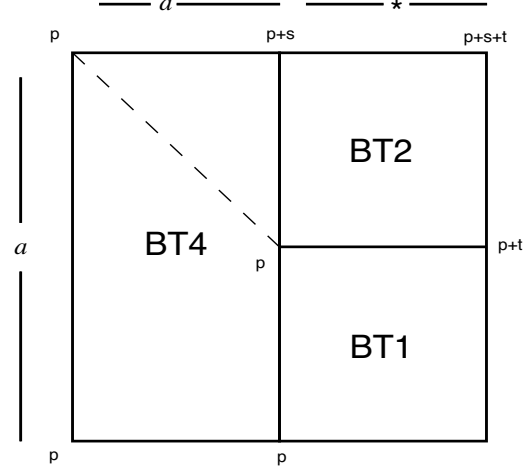


Fig. 6. BT4 with BT2 and BT1 to its right

extends to the right with values $p + s + 1, p + s + 2, \dots, p + s + t$. Suppose every element of the short sequence **b** above these extended matrix values differs from a . Then there is a portion P_2 of the forward matrix/graph of block type two that shares the matrix-values/vertices of the rightmost column of P_1 from the first vertex down to an including the transition vertex of P_1 and a portion P_3 of the forward matrix/graph of block type one that shares the matrix-values/vertices of the rightmost column of P_1 from the transition vertex of P_1 down to the last vertex.

Proof of Lemma 6: Lemma 5 provides the values of the matrix in the last column of P_1 . The transition vertex of P_1 divides these values up into two parts. The values on and above this vertex, together with those in the extended row yield a portion P_2 of block type two. Lemma 2 provides the values of the matrix in the last row of P_2 . Similarly the values on and below this vertex, together with those in the last row of P_2 yield a portion P_3 block type one.

The next lemma shows that to the right of every block type two is a block type three. See figure 7.

Lemma 7: Suppose a portion P_1 of the forward matrix/graph is of block type two. Suppose the values of the matrix in the first row, from left to right, are $p+r, p+r+1, \dots, p+r+s$ and that this row of the matrix extends to the right with values $p+r+s+1, p+r+s+2, \dots, p+r+s+t$. (The values in the first column, from bottom to top, are $p, p+1, \dots, p+r$.) Suppose every element of the short sequence s above these extended matrix values is a . Then there is a portion P_2 of the forward matrix/graph of block type three that shares every matrix-value/vertex of the rightmost column of P_1 .

Proof of Lemma 7: This lemma is proved in a way

similar to lemma 5.

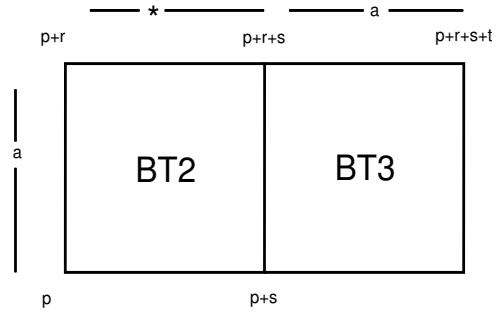


Fig. 7. BT2 with a BT3 to its right

The next lemma shows that to the right of every block type three is a block type two. See figure 8.

Lemma 8: Suppose a portion P_1 of the forward matrix/graph is of block type three. Suppose the values of the matrix in the first row, from left to right, are $p+r, p+r+1, \dots, p+r+s$ and that this row of the matrix extends to the right with values $p+r+s+1, p+r+s+2, \dots, p+r+s+t$. (The values in the first column, from bottom to top, are $p, p+1, \dots, p+r$.) Suppose every element of the short sequence **b** above these extended matrix values differs from a . Then there is a portion P_2 of the forward matrix/graph of block type two that shares every matrix-value/vertex of the rightmost column of P_1 .

Proof of Lemma 8: This proof is similar to lemma 5.

The long sequence **a** consists of an arbitrary but finite number of a 's. The short sequence **b** can contain arbitrary characters. A *partition* of a sequence v is a sequence of subsequences such that every element of v lies in some subsequence, every subsequence consists

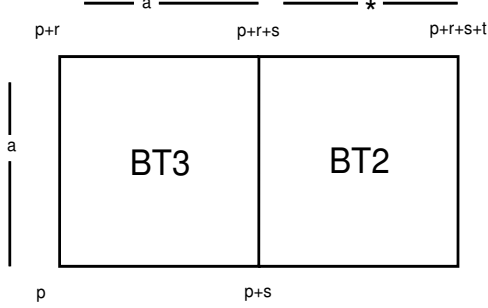


Fig. 8. BT3 with a BT2 to its right

either entirely of a 's or entirely of symbols different from a , and each subsequence is as long as possible. If s is partitioned into subsequences s_i and s_1 consists entirely of a 's then s_i consists entirely of a 's for odd i and entirely of symbols different from a for i even. If the s_i 's are concatenated in order from left to right then the original sequence s is obtained.

We now define the *main diagonal* (main band) of a matrix/graph. The upper bound of this band consists of all entries whose row number equals its column number. The lower bound is defined similarly, except we start at the element in the last row and column and move up and to the left along a diagonal. All elements on and between these diagonals constitute the main diagonal.

Theorem 1: The forward path graph can be decomposed into parts each of which is of one of the four block types.

Proof of Theorem 1: It is important to note that the parts overlap along their common borders. For example, if a block type two has a block type three to its right then the entries in the last column of the matrix/graph of the block on the left are identical to those in the first column of the block on the right. A decomposition in the sense of this theorem is not the same as a partition.

Partition the short sequence \mathbf{b} into subsequences s_1, s_2, \dots, s_z . It makes no difference whether s_1 consists only of a 's or only of symbols different from a . Without loss of generality assume the former, see Figure 9.

The first column of the matrix contains only zeros. The first row starts with a zero and increments by one until the last column is reached.

By lemma 4 there is a block of type four that stretches vertically the entire height of the matrix/graph and stretches horizontally from the first column up to and including the column corresponding to the last symbol in s_1 . We also know the numerical values of the matrix for this column by the lemma.

Since s_1 contains only a 's, no symbol in s_2 is an a . By lemma 6 we know the portion of the matrix/graph under s_2 starts with a block type two with a block type one below. Note that we have defined a vertical strip within the matrix/graph and that the block type two has height equal to its width while block type one extends all the way to the bottom row.

Since s_2 contains no a 's, s_3 contains only a 's. By lemma 7 we have a block type three to the right of the block type two in the second vertical strip. Similarly, by lemma 5 we have a block type four to the right of the block type one in the second vertical strip. The two blocks we have just described form the third vertical strip.

No element of s_4 is an a . By lemma 8 there is a block type two to the right of the block type three in vertical strip three. By lemma 6 we have block types two and one to the right of block type four in vertical strip three. These three blocks (in order from top to bottom: two, two, and one) define the fourth vertical strip.

From the four vertical strips two facts may be derived. First, for those s_i 's composed only of a 's there can only be block types three and four. All block type threes are above the main diagonal and the single block type four begins above the main diagonal and continues until the last row. Second, for the remaining s_i 's there can only be blocks types two and one. All block type twos are above the main diagonal and the single block type one begins above the main diagonal and continues until the last row.

Due to the alternating nature of the s_i 's and lemmas 5 through 8 this pattern will continue for as long as there are s_i 's.

For the forward distance matrix the values in the first column are set to zero and those of the first row are, from left to right, $0, 1, 2, \dots, n$, where n is the length of the short sequence \mathbf{b} . Values in the interior of the matrix are found by looking at values above, to the left, and diagonally up and left from the current element. For the reverse distance matrix the values in the last column are set to zero and those of the last row are, from right to left, $0, 1, 2, \dots, n$. Values in the interior of the matrix are found by looking at values below, to the right, and diagonally down and right from the current element. The procedure to determine a value in the interior of the matrix is identical in the forward and reverse cases when these orientation issues are taken into account. Consequently, everything we have done so far for the forward case carries over to the reverse case provided orientation issues are handled properly.

Geometrically, each of the four block types must

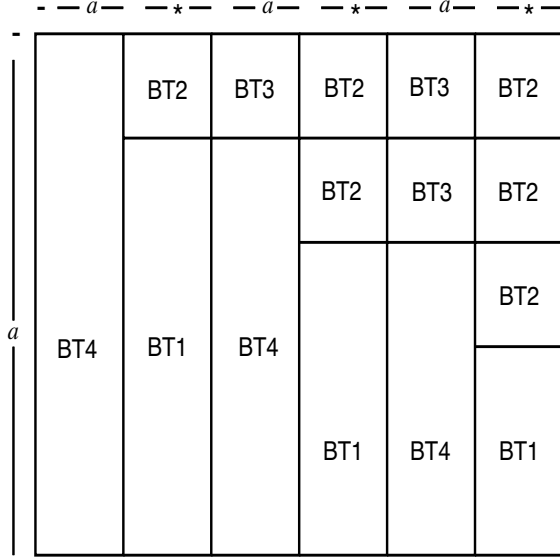


Fig. 9. Block structure of a forward matrix/graph

be rotated about their centers by 180 degrees. If block type i is transformed this way we will refer to it as a *rotated* block type i .

Lemmas 1 through 8 can now be rephrased for the reverse case. For example, lemma 6 stated that to the right of every block type four are blocks of type two and one (with type two above type one and type one extending to the bottom of the matrix/graph). Its counterpart states that to the left of every rotated block type four are rotated block types two and one (with type two below type one and type one extending to the top of the matrix/graph).

With these conventions we have the counterpart for theorem 1 for the case of the reverse matrix/graph.

Theorem 2: The reverse path graph can be decomposed into parts each of which is of one of the four rotated block types.

Proof of Theorem 2: Similar in nature to that of Theorem 1, and hence omitted.

In the proof of Theorem 1 the blocks are organized into vertical strips. The blocks are also organized into vertical strips for the reverse matrix/graph. The labeling of the vertices in the forward and reverse matrices and graphs are shifts of one another, i.e. the dash was moved to the other side of the short and long sequences and their symbols were shifted accordingly. It is easy to see that these shifts force the alignments of the vertical strips we have defined for the forward and reverse cases. (Although Sellers did not utilize

such strips in his papers, the notion of alignments of this sort were used in his proofs that the algorithms for finding local and global matches were correct.) Since the strips are common to the forward and reverse matrices/graphs we may refer to them in the common and transverse graphs as well.

III. CONNECTEDNESS IN THE COMMON AND TRANSVERSE PATH GRAPHS

Lemma 9: In each vertical strip of the common graph for which the corresponding portion of the short sequence \mathbf{b} consists only of a 's, each interior vertex is connected only to its neighbor up and to the left.

Proof of Lemma 9: Figure 10 illustrates the structure of the vertical strip. We will refer to this as a block type five (BT5 for short).

The common graph contains exactly those edges common to the forward and reverse path graphs. In the forward path graph we will have one or more block type threes above a single block type four that extends downwards for the remainder of the vertical height. In the reverse path graph we will have one or more rotated block type threes below a single rotated block type four that extends upwards for the remainder of the vertical height. Every one of these four block types have all possible diagonal edges. A block type four will have some horizontal edges but they are all above the main diagonal. Similarly a rotated block type four will have some horizontal edges as well but they are all below the main diagonal. Consequently no horizontal edge can appear in both the forward and reverse path graphs, and hence can not appear in the common graph. Since all diagonal edges appear in both the forward and reverse graphs they must appear in the common graph.

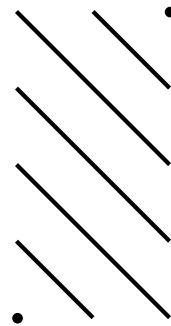


Fig. 10. Block Type 5 (BT5)

Lemma 10: Suppose P is a vertical strip of the Common Graph for which the corresponding portion of the short sequence s contains no a 's. Then P can be divided into three horizontal strips P_1 , P_2 , and P_3 , to be described below, where P_2 is always present and

P_1 and P_3 may or may not be present. Each vertex in P_1 , except for those in the first column, is connected only to its neighbor to the left. The same holds for P_3 . Each vertex in P_2 , except for those in the first row or first column, is connected to exactly two vertices: that to the left and the one diagonally up and to the left.

Proof of Lemma 10: Figure 11 illustrates the structure of the vertical strip. We will refer to this as a block type six, or BT6.

The proof is done in the same way as in Lemma 9, by intersecting BT1's and BT2's in the forward path graph and rotated BT1 and BT2 of the reverse path graph.

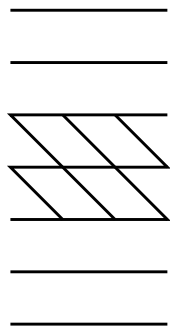


Fig. 11. Block Type 6 (BT6)

Theorem 3: The common graph can be decomposed into an alternating series of block types five and six.

Proof of Theorem 3: As usual, partition the short sequence \mathbf{b} into subsequences s_i . Each subsequence containing only a 's give rise to a block type five. Otherwise we have a block type six. Since the s_i 's alternate by definition, the block types must alternate.

We now state and prove our main result.

Theorem 4: Suppose the long sequence \mathbf{a} contains only the symbol a and the short sequence \mathbf{b} is arbitrary. The transverse graph is connected if and only if \mathbf{b} contains at least one symbol different from a .

Proof of Theorem 4: We need only eliminate from the common graph any edges that do not lie on some path beginning at the first column and ending at the last column.

Every block type five contains every possible diagonal path. Every block type six contains every possible horizontal path (as well as some diagonal edges, which we will ignore for now). These block types alternate. If we were to choose an arbitrary diagonal edge in a block type five or an arbitrary horizontal edge in a block type six, then this edge will be part of a path of maximal length formed by combining edges from the various vertical strips.

Suppose s contains at least one symbol different from a . Then there will be at least one block type six. Choose an arbitrary horizontal edge from a block type six that is common to the blocks P_1 and P_2 from the statement of lemma 10. If no such edge exists then P_1 does not exist and we choose any edge in the first row of P_2 . In either case we have an edge which will belong to some path as described in the previous paragraph. Call this path the *upper critical path*. We can define a *lower critical path* by choosing an edge between P_2 and P_3 , if one exists, or an edge from the last row of P_2 otherwise.

Any path above the critical path will fail to reach the first column. Similarly any path below the critical path will fail to reach the last column.

Now consider the diagonal edges of block types six that we have ignored until now. These edges will link the two critical paths and any paths between them and will not link any path above the upper critical path nor any path below the lower critical path. Consequently all paths above the upper critical path and all paths below the lower critical path will fail to reach either the first or last column and will have no edges connecting them to the central region that reaches both the first and last column. These paths will be absent from the transverse graph. The remaining paths, together with the diagonal edges of block types six, will form a single component that will be present in the transverse graph. Hence the transverse graph will be connected.

We now consider the case when \mathbf{b} is made up of only a 's. In this case there are no block types six. In fact, the entire graph is a single block type five and the Common Graph consists only of paths like those in Figure 10. It is clear that the Transverse Graph contains at least two paths that are connected by any additional edges, i.e. we have at least two components. Hence the transverse graph will be disconnected.

REFERENCES

- [1] D'Alotto, L., Lasser, L., "Disconnectedness in Forward and Reverse Path Graphs for Pattern Matching," *Proceedings of the International Conference on Artificial Intelligence*, Vol. 1, 302-305, Las Vegas, NV, (2004).
- [2] Sankoff, D., Kruskal, J., *Time Warps, String Edits, and Macromolecules, The Theory and Practice of Sequence Comparison*, CSLI Publications, (1999).
- [3] Sellers, P. H., "The Theory and Computation of Evolutionary Distances; Pattern Recognition," *J. Algorithms* **1**, 359-373, (1980).
- [4] Sellers, P. H., "Pattern recognition in genetic sequences," *Proc. Natl. Acad. Sci. USA* **76**, 3041, (1979).
- [5] Sellers, P. H., "On the Theory and Computation of Evolutionary Distances," *SIAM J. Appl. Math.* **26**, 787-793, (1974).
- [6] West, D. B., *Introduction to Graph Theory*, 2nd ed., Prentice Hall, NJ (2001).