

### Lecture Notes: Edit Distance

Given two strings  $x$  and  $y$ , we want to develop a dynamic programming algorithm to find the edit distance and the corresponding edit operations.

**Definition.** The *edit distance* between two strings  $x = x_1 \dots x_m$  and  $y = y_1 \dots y_n$  is the minimum number of edit operations needed to transform  $x$  into  $y$ . Possible edit operations are

- *insert*( $x, i, a$ ): insert  $a$  after the  $i$ -th character of  $x$  ( $x_1 x_2 \dots x_i a x_{i+1} \dots x_n$ )
- *delete*( $x, i$ ): delete the  $i$ -th character of  $x$  ( $x_1 x_2 \dots x_{i-1} x_{i+1} \dots x_n$ )
- *modify*( $x, i, a$ ): replace the  $i$ -th character of  $x$  with  $a$  ( $x_1 x_2 \dots x_{i-1} a x_{i+1} \dots x_n$ )

For example, if  $x = aabab$  and  $y = babb$ , then  $x$  can be transformed to  $y$  by *insert*( $x, 0, b$ ), *delete*( $x, 2$ ), *delete*( $x, 4$ ).

**Recursive Solution.** Let us define  $M[i, j]$  to be the minimum number of operations to transform  $x_1 x_2 \dots x_i$  into  $y_1 \dots y_j$  where  $0 \leq i \leq m$  and  $0 \leq j \leq n$ .  $M[0, j] = j$  because the only way to transform the empty string to  $y_1 \dots y_j$  is to add  $j$  characters. Similarly,  $M[i, 0] = i$ . For all other cases, there are three possible choices to transform  $x_1 \dots x_i$  to  $y_1 \dots y_j$ .

- **Case 1:** put  $y_j$  in the end to make  $x_1 \dots x_i y_j$  and then transform  $x_1 \dots x_i$  to  $y_1 \dots y_{j-1}$ .
- **Case 2:** remove  $x_i$  to make  $x_1 \dots x_{i-1}$  and then transform  $x_1 \dots x_{i-1}$  to  $y_1 \dots y_j$ .
- **Case 3:** change  $x_i$  to  $y_j$  (if they are different) which makes  $x_1 \dots x_{i-1} y_j$  and then transform  $x_1 \dots x_{i-1}$  to  $y_1 \dots y_{j-1}$ .

Therefore, the content of matrix  $M[i, j]$  can be formalized recursively as follows.

$$M(i, j) = \begin{cases} j & \text{if } i = 0 \\ i & \text{if } j = 0 \\ \min(M[i, j-1] + 1, M[i-1, j] + 1, M[i-1, j-1] + \text{change}(x_i, y_j)) & \text{Otherwise} \end{cases}$$

where  $\text{change}(x_i, y_j) = 1$  if  $x_i \neq y_j$  and  $\text{change}(x_i, y_j) = 0$  otherwise. The table has  $\Theta(mn)$  entries, each one computable in constant time. To obtain the corresponding edit operations, one can construct an auxiliary table  $OP[i, j]$ , which specifies what is the first operation needed to optimally transform  $x_1 \dots x_i$  to  $y_1 \dots y_j$ . The running time remains the same.

**Example.**  $x = aabab$  and  $y = babb$