

Minimum Edit Distance

給定兩字串 $A = a_1 a_2 \dots a_i \dots a_n$, $B = b_1 b_2 \dots b_j \dots b_m$. A 轉至 B
並定義三種運算

1. 刪除 a_i - 字元。符號記作 (\uparrow)
2. 在 a_i 後 插入 b_j 。符號記作 (\leftarrow)
3. a_i 替換 成 b_j 。符號記作 (\bowtie)

符號 (\wedge) 代表不變

Recursion:

令 $c[i, j]$ 為 A_i, B_j 最短運算序列的長度

$$c[i, j] = \min \begin{cases} (\uparrow) c[i-1, j] + 1, & \text{if } a_i \neq b_j \\ (\leftarrow) c[i, j-1] + 1, & \text{if } a_i \neq b_j \\ (\bowtie) c[i-1, j-1] + 1, & \text{if } a_i \neq b_j \\ (\wedge) c[i-1, j-1], & \text{if } a_i = b_j \end{cases}$$

$$c[i, 0] = (\uparrow) i$$

$$c[0, j] = (\leftarrow) j$$

Minimum Edit Distance 演算法

Input: $A = \langle a_1, a_2, \dots, a_n \rangle$, $B = \langle b_1, b_2, \dots, b_m \rangle$ // A convert to B

Output: length of Minimum Edit Distance

// $C[i, j]$ 為 A_i 和 B_j 序列長度的最經運算子序列的長度

1. $n \leftarrow \text{length}(A)$;
2. $m \leftarrow \text{length}(B)$;
3. for $i \leftarrow 1$ to n
4. $C[i, 0] \leftarrow i$; // 刪除 A_i
5. for $j \leftarrow 1$ to m
6. $C[0, j] \leftarrow j$; // 插入 B_j
7. for $i \leftarrow 1$ to n
8. for $j \leftarrow 1$ to m
9. if $a_i \neq b_j$
10. $C[i, j] = \min(C[i-1, j] + 1, C[i, j-1] + 1, C[i-1, j-1] + 1);$
11. else
12. $C[i, j] = C[i-1, j-1]$;
13. return $C[n, m]$;