2.10 编辑距离问题(EDP)

编辑距离问题(EDP)在[16,125页]中被称作“能用DP解决的不精确匹配问题”。它在[22, 284 - 286页]中也被称作“字串编辑问题”。[10,364-367页]还描述了这个问题的一些变种。

令$\sum$为一个有限的字母表。给定两个串$x \in \sum^m$和$y \in \sum^n$, 假定$x = x_1...x_m$且$y = y_1...y_n$。我们的任务就是用最少的编辑操作使x变成y。这里说的编辑操作有以下三种：

- 删除操作D:花费c(D)从串中删掉一个字母。
- 插入操作I:花费c(I)从$\sum$往串中插入一个字母。
- 复位操作(或者说是替换操作)R：花费c(R)用$\sum$中的一个字母把串中的一个字母换掉。

我们的目标是寻找一个编辑序列，让x以最小的花费变成y。这里我们定义编辑序列的费用为各操作的费用之和。
下面的递归式能够计算出x的前缀X_i变成y的前缀Y_j的最小花费$f(i,j)$。

$$f(i,j) = \begin{cases} jD & if\ i = 0 \\ iD & if\ j = 0 \\ \min\{f(i-1,j) + c(D), & \\ f(i,j-1) + c(I), f(i-1,j-1) + c(R)\} & if\ i > 0\ and\ j > 0 \end{cases}$$

这里费用函数c定义如下：

$$\begin{array}{ll} \text{c(D)=}c_D & \text{在任意位置删掉任意字母} \\ \text{c(I)=}c_I & \text{在任意位置插入任意字母} \\ \text{c(R)=}\begin{cases} 0 & if\ x_i = y_j \ \text{(字母相等)} \\ c_R & if\ x_i \neq y_j \ \text{(一个可行的复位)} \end{cases} & \end{array}$$

状态转移函数定义如下：

$$t(X_i, Y_j, D) = (X_{i-1}, Y_j)$$
$$t(X_i, Y_j, i) = (X_i, Y_{j-1})$$
$$t(X_i, Y_j, R) = (X_{i-1}, Y_{i-1})$$

我们的目标是计算$f(x,y)$，即编辑序列的最小花费。

考虑[16,223页]中的一个例子，x=”CAN”且y=”ANN”，插入花费$c_I = 1$，删除花费$c_D = 1$，复位花费$c_R = 1$。有以下几种花费f(x,y)=2（花费最小）的编辑序列:

- CAN $\vdash_R$ CNN $\vdash_R$ ANN,
- CAN $\vdash_D$ AN $\vdash_I$ ANN,
- CAN $\vdash_I$ CANN $\vdash_D$ ANN,

这个问题和2.23节点最长公共子序列问题(LCS)有很密切的联系。