LIEN ENTRE PLUS LONGUE SOUS-CHAINE COMMUNE ET DISTANCE DE LEVENSHTEIN

un peu Cochemore 1.3 p 242.

Pb Le problème de la plus longue sous chaine commune (PLSCC) est de trouver, étant données en entrée deux chaines = mots x et y, un mot z qui soit à la fois sous-chaine / sous-mot de x et sous-mot de y, et de longueur maximale.

Dans un premier temps on pourra ne chercher que la longueur maximale d'une sous-chaine commune.

⚠ sous mot ≠ facteur sous-mot ≃ suite extraite.

Déf Soit $x = x_1 \text{---} x_n$ un mot.
$z = z_1 \text{---} z_\ell$ est un sous mot de x
$\Leftrightarrow$ il existe une fonction strictement croissante φ de $[1..\ell]$ vers $[1..n]$ telle que $\forall i \in [1..\ell]$ $z_i = x_{\varphi(i)}$.
ou dit il existe une "extra" φ telle que $z = x_{\varphi(1)} \text{---} x_{\varphi(\ell)}$

NB : facteur ⇒ sous-mot

ex LABO est un sous mot de LABRADOR

ici $n=8$, $\ell=4$ et $\varphi = \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 2 \\ 3 \mapsto 3 \\ 4 \mapsto 8 \end{cases}$

mais LABO n'est pas un facteur de LABRADOR.

En revanche LAB, OR, ABRA sont des facteurs de LABRADOR et donc des sous-mots.

Alignement → SCC

Un alignement fournit naturellement une sous-chaine commune (SSC) il suffit de ne considérer que les lettres qui sont "support" d'une substitution triviale (change une lettre en elle même).

ex

x = BLABLABLA $n = 9$

y = ABRACADABRA $m = 12$

$\tilde{x}$: _ B L _ A B L A _ _ B L A

$\tilde{y}$: A B _ R A C _ A D A B R A

↳ BAABA est une SSC à x et y.

Cependant si un alignement ne contient pas de substitution non triviale, cela s'exprime encore plus facilement : il suffit de considérer les lettres qui ne sont pas-face à un tiret.

Considérons $ins = del = 1$ et $sub(a,b) = \begin{cases} 3 & \text{si } a \neq b \\ 0 & \text{si } a = b \end{cases}$ jusqu'à la fin de cette fiche.

Puisque la substitution* est strictement plus chère qu'une insertion et une suppression, un alignement minimal sera sans substitution*

* = non triviale. Sous entendu dans la suite.

eac

x = BLABLABLA $n = 9$

y = ABRACADABRA $m = 12$

$\tilde{x}$ _ B L _ A B _ L A _ _ B L _ A

$\tilde{y}$ A B _ R A _ C _ A D A B _ R A

1 0 1 1 0 1 1 1 0 1 1 0 1 1 0

→ BAABA, $L = 5$

↳ coût de l'alignement $\tilde{C}(x,y) = 4 + 6 = 10$

Nb del Nb ins

SCC -> alignement

A l'inverse une SCC z de x et y fournit un alignement de x et y.

On écrit $z = z_1 \cdots z_L$. Il existe, puisque z est sous-mot de x et de y, deux fonction φ et ψ, strictement croissantes, telles que $z = (x_{\varphi(i)})_{i \in [\![1..L]\!]} = (y_{\psi(i)})_{i \in [\![1..L]\!]}$.

Alors on peut décomposer $\begin{cases} x = u_0\, x_{\varphi(1)}\, u_1 \cdots x_{\varphi(L)}\, u_L \\ y = v_0\, y_{\psi(1)} \cdots y_{\psi(L)}\, v_L \end{cases}$ où

les (v_i) et les (u_i) sont des mots, éventuellement vides.

Alors on a l'alignement $\begin{cases} \tilde{x} = u_0 \,(\text{-}\text{-}\text{-})\, x_{\varphi(0)}\, u_1 \,(\text{-}\text{-}\text{-}) \cdots x_{\varphi(L)}\, u_L \,(\text{-}\text{-}\text{-}) \\ \tilde{y} = (\text{-}\text{-}) \, v_0 \, y_{\psi(0)} \,(\text{-}\text{-}\text{-})\, v_1 \cdots y_{\psi(L)} \,(\text{-}\text{-}\text{-})\, v_L \end{cases}$

NB: L'alignement n'est ni unique ni canonique.

ex

$x = BLABLABLA$ $\quad$ $y = ABRACADABRA$

φ : $z = BAABA$ $\quad$ ψ : $z = BAABA$

$u_0 = \varepsilon$, $u_1 = L$, $u_2 = BL$, $u_3 = \varepsilon$, $u_4 = L$, $u_5 = \varepsilon$

$v_0 = A$, $v_1 = R$, $v_2 = C$, $v_3 = DA$, $v_4 = R$, $v_5 = \varepsilon$

d'où l'alignement

$\tilde{x}$: _ B L _ A B L _ A _ _ B L _ A

$\tilde{y}$: A B _ R A _ _ C A D A B _ R A

Lev <-> L

Si z est une SSC de x et y "correspondant" à l'alignement $(\tilde{x}, \tilde{y})$ alors $\boxed{2|z| = |x| + |y| - C(\tilde{x}, \tilde{y})}$

Preuve Considérons N = le nombre total de symbole dans l'alignement $(\tilde{x}, \tilde{y})$ (ie sur les 2 lignes)

$N = |x| + |y| + N_{ins} + N_{del}$ // en comptant les lettres puis les tirets

$= 2(|z| + N_{ins} + N_{del})$ // en regardant comment chaque "colonne" est colorée

Donc $2|z| = |x| + |y| + N_{ins} - 2N_{ins} + N_{del} - 2N_{del} = |x| + |y| - \underbrace{(1 \times N_{del} + 1 \times N_{ins})}_{= C(\tilde{x}, \tilde{y})}$ (del, ins)

Cor $L(x,y) = \dfrac{|x|+|y| - Lev(x,y)}{2}$ où L désigne la longueur d'1 PLSCC

Preuve 1) $L(x,y) \geq \dfrac{|x|+|y| - Lev(x,y)}{2}$ $\quad n = |x|$, $m = |y|$

Pour chaque alignement de coût C il existe une PLSCC de longueur L où $2L = n+m-C$. Comme $L \leq L(x,y)$ pour déf. on a $n+m-C \leq 2L(x,y)$. En particulier pour un alignt de coût min on a $n+m-Lev(x,y) \leq 2L(x,y)$ soit $L(x,y) \geq \dfrac{n+m-Lev(x,y)}{2}$

2) $\dfrac{|x|+|y| - Lev(x,y)}{2} \geq L(x,y)$.

Etant donné une SCC de taille L on peut construire un alignt de cout C ou $2L = n+m-C$. Comme $C \geq Lev(x,y)$ on a $2L \leq n+m-Lev(x,y)$. En particulier pour une PLSCC on a $2L(x,y) \leq \dfrac{n+m-Lev(x,y)}{2}$

On conclut par double inégalité.

Cor Calculer la distance de Levenshtein revient à calculer la longueur d'une plus longue sous-chaine commune pourvu qu'on ait fixé $del = ins = 1$ et $sub(a,b) = \begin{cases} 2 \text{ si } a \neq b \\ 0 \text{ si } a = b \end{cases}$

De plus étant donné un alignemt réalisant la distance de Levenshtein on a une PLSCC, et réciproquemt étant donné une SCC de longueur $L(x,y)$, on a un alignemt de coût minimal.

minimiser $C(\tilde{x},\tilde{y}) \Leftrightarrow$ maximiser $|z|$