Complément du cours “Édition d’arborescences”, M2R UMINR337,
Sèverine Bérard, nov. 2007.

Algorithme de Zhang et Shasha (complet)

Afin de calculer la distance d’édition entre deux arborescences ordonnées étiquetées, il est nécessaire de calculer auparavant la distance entre certaines paires de sous-arborescences ainsi qu’entre certaines paires de sous-forêts ordonnées.

Dans les algorithmes qui suivent, les sommets des arborescences sont ordonnées selon l’ordre suffixe gauche-droite.

Rappel des notations

- $T[i..j]$ représente la forêt ordonnée de T induite par les sommets numérotés de i à j inclus ;
- $t[i_1]$, …, $t[i_{n_i}]$ représentent les sommets fils de $t[i]$;
- $l(i)$ représente le numéro dans l’ordre suffixe gauche-droite de la première feuille (celle la plus à gauche) du sous-arbre enraciné en $t[i]$;
- la distance entre deux arborescences ordonnées étiquetées $T_1[i]$ et $T_2[j]$ est notée $treedist(i, j)$;
- La distance d’édition entre deux sous-forêts ordonnées $T_1[i'..i]$ et $T_2[j'..j]$ est notée $forestdist(T_1[i'..i], T_2[j'..j])$.

Ces notations ainsi que les algorithmes qui suivent sont reproduit d’après la thèse de Laurent Tichit [2], à l’origine issus de [1].

Algorithme 1: Calcul de la distance d’édition entre T_1 et T_2.

Données : Deux arborescences ordonnées et étiquetées : T_1 et T_2.
Résultat : $treedist(T_1[i], T_2[j])$ où $1 \leq i \leq |T_1|$ et $1 \leq j \leq |T_2|$.

```
/* Pré-traitement */
Calcul de l() ;
Calcul de LR_keyroots(T1) ;
Calcul de LR_keyroots(T2) ;
/* Récurrence principale */
pour s := 1 à |LR_keyroots(T1)| faire
    pour t := 1 à |LR_keyroots(T2)| faire
        i := LR_keyroots(T1)[s] ;
        j := LR_keyroots(T2)[t] ;
        Calcul de treedist(i, j) ;
```

Algorithme 2: $treedist()$

Données : i, j, T_1, T_2, $l()$ et les $LR\_keyroots$.

Résultat : $treedist(T_1[s], T_2[t])$ où $s \in desc(i)$ et $t \in desc(j)$ avec $l(s) = l(i)$ et $l(t) = l(j)$.

/* *Initialisation* */
$forestdist(\epsilon, \epsilon) = 0$;
pour $i_1 := l(i)$ *à* i **faire**
  $forestdist(T_1[l(i)..i_1], \epsilon] = forestdist(T_1[l(i)..i_1 - 1], \epsilon] + \gamma(t_1[i_1], -)$;
pour $j_1 := l(j)$ *à* j **faire**
  $forestdist(\epsilon, T_2[l(j)..j_1]] = forestdist(\epsilon, T_2[l(j)..j_1 - 1]] + \gamma(-, t_2[j_1])$;
/* *Remplissage* */
pour $i_1 := l(i)$ *à* i **faire**
  pour $j_1 := l(j)$ *à* j **faire**
    si $l(i_1) = l(i)$ *et* $l(j_1) = l(j)$ **alors**
      $$forestdist(T_1[l(i)..i_1], T_2[l(j)..j_1]) = \min \begin{cases} forestdist(l(i)..i_1 - 1 \ , l(j)..j_1) & + \ \gamma(t_1[i_1], -) \\ forestdist(l(i)..i_1 \ , l(j)..j_1 - 1) & + \ \gamma(-, t_2[j_1]) \\ forestdist(l(i)..i_1 - 1 \ , l(j)..j_1 - 1) & + \ \gamma(t_1[i_1], t_2[j_1]) \end{cases} ;$$
      $treedist(i_1, j_1) = forestdist(T_1[l(i)..i_1], T_2[l(j)..j_1])$;
      /* *Stocker les valeurs de treedist dans un tableau permanent* */
    sinon
      $$forestdist(T_1[l(i)..i_1], T_2[l(j)..j_1]) = \min \begin{cases} forestdist(l(i)..i_1 - 1 \ , l(j)..j_1) & + \ \gamma(t_1[i_1], -) \\ forestdist(l(i)..i_1 \ , l(j)..j_1 - 1) & + \ \gamma(-, t_2[j_1]) \\ forestdist(l(i)..l(i_1) - 1 \ , l(j)..l(j_1) - 1) & + \ treedist(i, j) \end{cases} ;$$

Références

[1] K. Zhang , D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems", *SIAM Journal on Computing*, vol. 18 num. 6, p.1245-1262, déc. 1989

[2] L. Tichit, "Algorithmique des structures biologiques : l'édition d'arborescences pour la comparaison des structures secondaires d'ARNs". Thèse de l'université de Bordeaux1, sept. 2003.