Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Volker Heun

SS 2000
Lösungsblatt 3
8. Juni 2000

Algorithmen der Bioinformatik

Abgabetermin: 30. Mai

Aufgabe 1

a) In der Vorlesung wurde eine Methode zur Konstruktion eines *verallgemeinerten Suffixbaums* zu einer Menge $M = \{w_1, w_2, \ldots, w_n\}$ von Strings vorgeführt. Dabei wurde ein künstliches Wort $w' = w_1\$_1 w_2\$_2 \cdots w_n\$_n$ betrachtet. Dieser Baum ist jedoch größer als notwendig. Geben Sie eine Methode zur Erzeugung eines kleineren Suffixbaums an, der alle Suffixe der Wörter von M enthält.

Lösungsvorschlag:
Statt einen Suffixbaum für das Wort $w' = w_1\$_1 w_2\$_2 \cdots w_n\$_n$ zu konstruieren, der stets zuviele Suffixe enthält und auch nach strenger Auffassung kein echter Suffixbaum ist (– die Suffixes sollten bijektiv den Blättern zugeordnet sein –), wollen wir einen Baum für genau die gewünschte Suffixmenge aufbauen. Zunächst eine naive $O(n^2)$-Implementierung: Erzeuge eine einzige mit $w_1\$_1$ beschriftete Kante. Dann betrachte das Suffix $w^1_{2\ldots|w_1|}\$_1$. Falls dieses mit w_1 ein leeres gemeinsames Präfix hat, so erzeuge eine neue Kante für $w^1_{2\ldots|w_1|}\$_1$. Sonst spalte die Kante für $w_1\$_1$ durch Einfügen eines neuen Knotens auf in zwei Kanten: Die "obere" Kante (u, v) ist mit dem Präfix beschriftet, und die "untere" Kante mit dem Suffix von $w_1\$_1$. Hänge an v eine neue Kante für das Suffix von $w^1_{2\ldots|w_1|}\$_1$ an. Nun haben wir einen Baum mit den ersten beiden Suffixes von $w_1\$_1$. Fahre so fort, bis ein vollständiger Suffixbaum für $w_1\$_1$ konstruiert ist. Nun betrachte das Wort $w_2\$_2$ und führe genau dieselbe Konstruktion aus, aber unter Weiterverwendung des bereits existierenden Baums. Dieser Baum wird dann Bätter haben, deren Kanten mit Strings der Form $x\$_1$ *und* $x\$_2$ beschriftet sind. Nun betrachte $w_3\$_3$, etc.
Diese Konstruktion kostet natürlich quadratische Zeit, kann aber mit dem Algorithmus von Ukkonen genauso in linearer Zeit ausgeführt werden wie die Konstruktion eines Suffixbaums für ein einzelnes Wort. Die hierbei verwendeten impliziten Suffixbäume lassen sich nämlich auf dieselbe Weise auf mehrere Strings verallgemeinern. Da dies in der Aufgabe jedoch nicht verlangt war, verzichten wir auf eine detaillierte Ausführung.

b) Konstruieren Sie nach Ihrer Methode den Suffixbaum für die Sprache

$$\{0101100011, 1101, 01010\}.$$

Lösungsvorschlag: Siehe Figur 1.

Aufgabe 2

Sei, wie in der Vorlesung definiert, $w : (\Sigma \cup \{-\})^2 \to \mathbf{R}_0^+$ die Kostenfunktion für Edit-Operationen eines Sequence Alignment, und seien die Gesamtkosten für eine Folge $S = s_1 s_2 \dots s_n$ von Edit-Operationen definiert als $w(S) = \sum_{i=1}^{n} w(s_i)$. Für zwei Wörter $x, y \in (\{\Sigma \cup \{-\})^*$ sei $d_w(x, y) := \min\{w(S) \mid x \Rightarrow_S y\}$.
Zeigen Sie: Ist die Funktion w eine Metrik, so ist auch d_w eine Metrik.

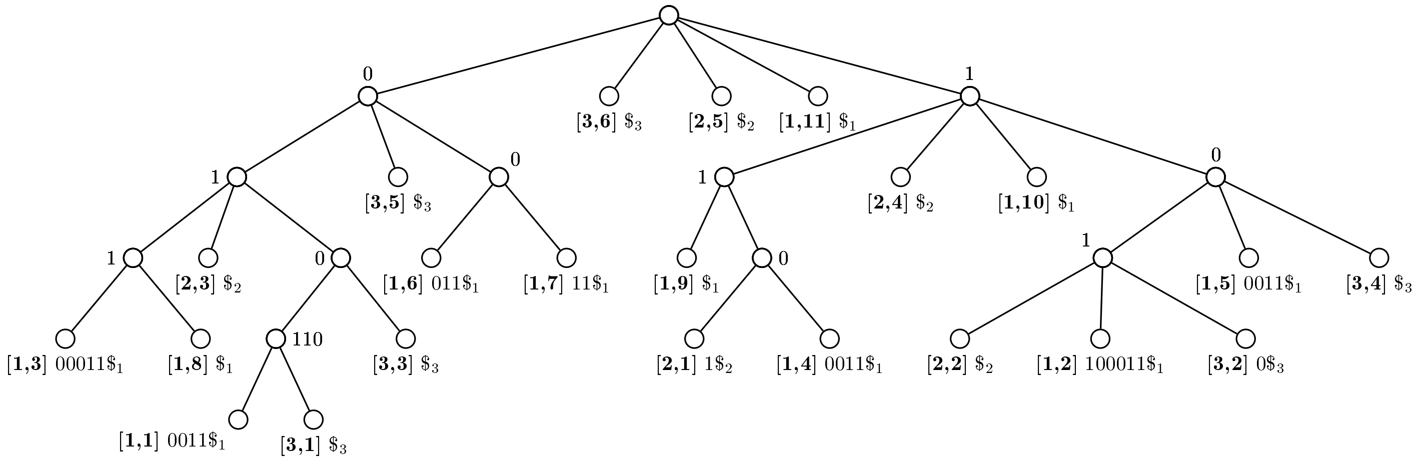
Lösungsvorschlag:
Eine Funktion $d : M^2 \longrightarrow \mathbf{R}_0^+$ ist genau dann eine *Metrik*, wenn für alle $x, y, z \in M$ gilt:

(i) $d(x, y) = 0$ g.d.w. $x = y$

(ii) $d(x, y) = d(y, x)$

(iii) $d(x, z) + d(z, y) \geq d(x, y)$.

Seien $z, x, y \in \Sigma^*$ beliebige Wörter. Bedingung (i) ergibt sich unmittelbar aus den Definitionen: Aus $d(x, y) = 0$ folgt die Existenz einer Edit-Sequenz S, wie x nach y überführt und $w(S) = 0$ erfüllt. Sei $S = s_1 \dots s_n$, dann ist wegen der Nichtnegativität der w-Werte klar, daß $w(s_i) = 0$ für alle i. Somit folgt, daß keines der s_i eine Änderung an einem Zeichen vornimmt, und damit $x = y$. Die Umkehrung ist trivial, ebenso wie die Symmetriebedingung (ii). Für die Dreiecksungleichung gehen wir nicht-konstruktiv vor und erhalten auf diese Weise einen extrem kurzen Beweis. Zu zeigen ist $d(x, z) + d(z, y) \geq d(x, y)$. Sei $S_{x,z}$ eine kostenminimale Editsequenz für $x \Rightarrow z$, und sei $S_{z,y}$ eine kostenminimale Editsequenz für $z \Rightarrow y$. Dann ist ihre Konkatenation (bzw. Komposition) $S_{x,z} S_{z,y}$ eine Editsequenz für $x \Rightarrow y$, aber nicht notwendig eine minimale. Daher gilt: $d(x, z) + d(z, y) = w(S_{x,z} S_{z,y}) \geq d(x, y)$.



Figur 1: Gemeinsamer Suffixbaum für die Sprache $\{0101100011, 1101, 01010\}$