Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Volker Heun

Wintersemester 2001/2002
Übungsblatt 4
14. November 2001

Algorithmische Bioinformatik

Abgabetermin: Donnerstag, den 22. November (15⁰⁰ Uhr) in der Übung

Aufgabe 1

Konstruiere einen Suffix-Trie und einen Suffix-Tree für *abaabababab*\$.

Hinweis: *Die als freiwillig gekennzeichneten Aufgaben werden nicht bewertet und sind für die Klausurzulassung ohne Bedeutung.*

Freiwillige Aufgabe 1

Sei Σ ein endliches Alphabet und sei $* \notin \Sigma$ ein so genanntes *dont-care*-Symbol. Ferner sei $s \in (\Sigma \cup \{*\})^+$ und $t \in \Sigma^+$ mit $|s| = m$ und $|t| = n$ sowie $k \in \mathbb{N}$ eine Konstante, die unabhängig von n und m ist.

Entwerfe einen Algorithmus, der feststellt, ob s ein Teilwort von t matched, d.h., ob es ein $i \in [0 : n - m]$ gibt, so dass für alle $j \in [0 : m - 1]$ entweder $s_j = *$ oder $s_j = t_{i+j}$ ist. Wenn das *dont-care*-Symbol dabei genau k-mal in s auftritt, soll der Algorithmus dabei maximal $O(k \cdot n + m)$ Vergleiche ausführen.

Hinweis: Versuche den Algorithmus von Knuth-Morris-Pratt geeignet zu modifizieren.

Freiwillige Aufgabe 2

Sei $v : \Sigma^m \to \mathbb{N}_0$ eine Bewertungsfunktion von Zeichenketten der Länge m über dem Alphabet $\{0, 1\}$ vermöge $v : w_1 \cdots w_m \mapsto \sum_{i=0}^{m-1} w_{m-i} 2^i$.

Zeige, das die folgende Variante eines numerischen Algorithmus zur Textsuche korrekt ist, und zeige, wie man die Berechnungen von $v(s)$ und insbesondere von $v(t_{i,m})$ effizient implementiert. Hierbei ist $t_{i,m} = t_i \cdots t_{i+m-1}$.

```
IS_SUBSTRING(char t[], int n, char s[], int m)
{   /* Es gelte s,t ∈ {0,1}^+ */
    for (int i = 0; i ≤ n − m; i++)
    {
        if (v(s) == v(t_{i,m})) return true ;
    }
    return false ;
}
```