Algorithmen und Programmieren III, WS 1999/2000 — 9. Übungsblatt

Abgabe bis Dienstag, 25. Januar 2000

44. (3 Punkte)

    (a) (3 Punkte) Berechnen Sie die Verschiebefunktion des *Fibonacci-Wortes*

    $$abaababaabaababaababaabaababaabaab.$$

    (b) (0 Punkte) Berechnen Sie die Verschiebefunktion der *Morse-Folge*

    $$0110100110010110100101100110100110010110011010010110100110010110.$$

45. (0 Punkte) Die verbesserte Verschiebefunktion zum Suchen von Zeichenketten ist folgendermaßen definiert:

    $$h[i] = \max \{ k \mid 1 \leq k < i,\ p_1 \ldots p_{k-1} = p_{i-k+1} \ldots p_{i-1} \text{ und } p_k \neq p_i \} \cup \{0\}$$

    (a) Berechnen Sie die verbesserte Verschiebefunktion der Muster aus der vorigen Aufgabe.

    (b) Zeigen Sie, dass beim Suchen mit der verbesserten Verschiebefunktion auf keinen Fall mehr Vergleiche der Form $p_i = s_j$ durchgeführt werden als mit der ursprünglichen Verschiebefunktion (ohne die Bedingung „$p_k \neq p_i$"). Gilt dies auch, wenn man den Aufwand an Vergleichen beim Berechnen der Verschiebefunktion mit berücksichtigt? Finden Sie ein Beispiel, bei dem tatsächlich weniger Vergleiche notwendig sind.

    (c) Schreiben Sie einen Algorithmus zum Berechnen der verbesserten Verschiebefunktion.

46. (5 Punkte) Erweitern Sie den Knuth–Morris–Pratt-Algorithmus so, dass er *alle* Stellen bestimmt, an denen das Muster im Text vorkommt.

47. (8 Punkte) Eine *Multimenge* ist etwas Ähnliches wie eine Menge, außer dass Elemente auch mehrfach vorkommen dürfen. Die Reihenfolge spielt keine Rolle. Zum Beispiel ist $\{\!\{a,b\}\!\} \neq \{\!\{a,a,b\}\!\} = \{\!\{a,b,a\}\!\} \neq \{\!\{a,a,a,b\}\!\}$.

    (a) (4 Punkte) Schreiben Sie eine Spezifikation für einen abstrakten Datentyp von Multimengen über der Grundmenge der ganzen Zahlen (`int`), die folgende Operationen unterstützt: Erzeugen einer leeren Multimenge; Einfügen und Streichen eines Elementes (dabei wird die Vielfachheit jeweils um 1 erhöht beziehungsweise erniedrigt); Feststellen der Vielfachkeit eines Elementes.

    (b) (4 Punkte) Geben Sie eine konkrete Darstellung (etwa als Java-Klasse `Multimenge`) an. Beschreiben Sie die Abstraktionsfunktion, sowie die Invarianten, die die gültigen Darstellungen charakterisieren. Geben Sie auch die Vorbedingungen für alle Operatiomen an. (Sie dürfen dabei vernünftige Einschränkungen für die verfügbaren Operationen machen.)

    (c) (0 Punkte) Implementieren Sie die Operationen.

    (d) (0 Punkte) Beweisen Sie die Korrektheit Ihrer Implementierung.

48. (0 Punkte) Schreiben Sie eine Spezifikation für *Intervallarithmetik*.
    Intervallarithmetik liefert verlässliche Ergebnisse, selbst wenn die Eingabedaten mit Messfehlern behaftet sind und zwischendurch Rechenfehler auftreten. Statt mit Zahlen rechnet man mit Intervallen $[a, b]$, die durch die arithmetischen Grundoperationen, $+$, $-$, $\times$ und $:$ miteinander verknüpft und miteinander verglichen werden können (durch $<$, $>$, $\leq$, usw.) Das Ergebnis ist wieder eine Intervall.