Grundlagen der theoretischen Informatik, SS 2001 — 5. Übungsblatt

Abgabe bis Dienstag, 5. Juni 2001

---

26. (4 Punkte) Welche der folgenden Sprachen über $\Sigma = \{0, 1\}$ sind regulär? beweisen Sie Ihre Antworten.

    (a) Die Sprache $L_1 = \{ x \mid x = x^R \}$ der Palindrome

    (b) $L_2 = \{ 0^{i^2} \mid i \geq 1 \}$

    (c) $L_3 = \{ x \mid x$ enthält mindestens so viele Nullen wie Einsen $\}$

    (d) $L_4 = \{ x \mid$ jeder Präfix von x enthält mindestens so viele Nullen wie Einsen $\}$

    (e) (2 Punkte) $L_5 = \{ x \mid$ die Anzahl der Nullen von x unterscheidet sich um höchstens um 1 von der Anzahl der Einsen $\}$

    (f) (2 Punkte) $L_6 = \{ x \mid$ in jedem Präfix von x unterscheidet sich die Anzahl der Nullen höchstens um 1 von der Anzahl der Einsen $\}$

    Ein *Präfix* ist dabei ein Anfangsstück eines Wortes: x ist Präfix von y, wenn $y = xu$ für ein $u \in \Sigma^*$ ist.

27. (0 Punkte) Es seien Σ und Δ zwei Alphabete. Ein *Homomorphismus* $h\colon \Sigma^* \to \Delta^*$ ist durch eine beliebige Funktion $h\colon \Sigma \to \Delta^*$ gegeben, die folgendermaßen auf Σ^* erweitert wird:

    $$h(x_1 x_2 \ldots x_n) := h(x_1)h(x_2)\ldots h(x_n), \text{ für } n \geq 0,\ x_1, \ldots, x_n \in \Sigma$$

    Ein Homomorphismus ist durch folgende Eigenschaften charakterisiert:

    $$h(\varepsilon) = \varepsilon, \text{ und } h(x \cdot y) = h(x) \cdot h(y), \text{ für alle } x, y \in \Sigma^*$$

    Für eine *Sprache* $L \subseteq \Sigma^*$ bezeichnet $h(L)$ die Sprache

    $$h(L) := \{ h(x) \mid x \in L \}.$$

28. (4 Punkte) Welche der Sprachen $h_j(L_i)$ sind regulär, wenn L_i eine der Sprachen aus Aufgabe 26 ist und der Homomorphismus h_j folgendermaßen gegeben ist?

    (a) $h_1(0) = a$, $h_1(1) = b$, $\Delta = \{a, b\}$.

    (b) $h_2(0) = 1$, $h_2(1) = 1$.

    (c) $h_3(0) = 01$, $h_3(1) = \varepsilon$.

    Es gibt je 2 Punkte für die Sprachen $h_3(L_4)$ und $h_2(L_1)$.

29. (8 Punkte) Erweitern Sie das in der Vorlesung vorgestellte `lex`-Programm[1] zur Addition aller in der Eingabe vorkommenden Zahlen um folgende Funktionen:

    (a) Bei Eingabe des Zeichens `*` oder `-` wird die nächste Zahl nicht addiert, sondern multipliziert beziehungsweise subtrahiert. Bei Eingabe von `=` wird das bisherige Ergebnis ausgedruckt.

    (b) Es werden zusätzlich auch Kommentare im Stil von Haskell verstanden: Alles zwischen `--` und dem Ende der Zeile wird ignoriert.

    Die obigen Angaben lassen einige Fragen offen, zum Beispiel den Unterschied zwischen Subtraktion (`5 - 3`) und einem negativen Vorzeichen (`5 * -3`). Ergänzen Sie zunächst die obige Aufgabenspezifikation *in Worten*, sodass das Verhalten des Programmes eindeutig daraus hervorgeht. Ihr Programm sollte dann dieser Spezifikation entsprechen.

---

[1] `http://www.inf.fu-berlin.de/~rote/Lere/Grundlagen%20der%20theoretischen%20Informatik/flex/summe.flex`