## Grundlagen der theoretischen Informatik, SS 2005 — 4. Übungsblatt

Abgabe bis Dienstag, 17. Mai 2005, 10:15 Uhr

27. Konstruieren Sie für den DEA $M = (Q = \{a, b, c\}, \Sigma = \{0, 1\}, \delta, a, \{b\})$ einen regulären Ausdruck, der die von $M$ akzeptierte Sprache beschreibt. Die Übergangsfunktion $\delta$ ist in der nebenstehenden Tabelle angegeben.

| $\delta$ | 0 | 1 |
|---|---|---|
| $a$ | $a$ | $b$ |
| $b$ | $c$ | $b$ |
| $c$ | $b$ | $a$ |

    Wenden Sie den in der Vorlesung vorgestellten Algorithmus von Kleene auf die Zustände einmal in der Reihenfolge $a, b, c$ und dann in der Reihenfolge $c, a, b$ an. Vereinfachen Sie dabei die Zwischenergebnisse, so gut es geht. Versuchen Sie auch, „direkt“ durch Betrachten des Zustandsdiagramms und Nachdenken einen möglichst einfachen regulären Ausdruck zu finden.

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

    (a) Es sind auch Zahlen in Exponentialnotation wie zum Beispiel `0.54e-7` $= 0{,}54 \times 10^{-7}$ zugelassen.
    (b) Bei Eingabe von `*` wird die nächste Zahl nicht addiert, sondern multipliziert.
    (c) Bei Eingabe von `=` wird das bisherige Ergebnis ausgedruckt.
    (d) 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 die Wechselwirkung zwischen den verschiedenen Arten von Kommentaren. Ergänzen Sie zunächst die obige Aufgabenspezifikation *in Worten*, sodass das Verhalten des Programmes eindeutig daraus hervorgeht. Ihr Programm sollte dann dieser Spezifikation entsprechen.

29. Konstruieren Sie einen regulären Ausdruck für die Sprache $L_6$ von Aufgabe 23f.

30. (Zusatzaufgabe) Konstruieren Sie DEAs und NEAs für folgende Sprachen über $\{0, 1\}$:

    (a) die Wörter, deren vierter Buchstabe eine 0 ist;
    (b) die Wörter, deren viertletzter Buchstabe eine 0 ist.

31. Definieren Sie formal die Nachfolgerrelation $xqy \vdash x'q'y'$ für Konfigurationen einer Turingmaschine. ($q, q' \in Q$, $x \in \{\varepsilon\} \cup (\Gamma - \{B\})\Gamma^*$, $y \in \{\varepsilon\} \cup \Gamma^*(\Gamma - \{B\})$.)

32. (10 Punkte) Multiplikation. Konstruieren Sie eine Turingmaschine, die zwei „unäre“ Zahlen multipliziert. Bei Eingabe von $0^m\#0^n$ soll die Ausgabe $0^{mn}$ berechnet werden.

33. Binäre Multiplikation.
    In der Vorlesung wurde eine Turingmaschine vorgestellt, die die *Summe* zweier Zahlen in Binärdarstellung berechnet. Bei Eingabe von $\mathrm{bin}(x)$#$\mathrm{bin}(y)$\$ und Start im Zustand $q_0$ hält die Maschine im Zustand $q_F$ mit der Ausgabe $\mathrm{bin}(x+y)$#$\mathrm{bin}(y)$\$ und dem Kopf auf dem ersten Eingabesymbol. (Die Notation $\mathrm{bin}(x) \in \{0, 1\}^*$ ist eine Darstellung von $x \in \mathbb{N}$ in Binärdarstellung.) Der Bandinhalt rechts vom \$-Zeichen bleibt dabei unverändert stehen.

    Erweitern Sie diese Maschine zu einer Maschine, die zwei Binärzahlen *multipliziert*: Bei Eingabe von $\mathrm{bin}(x)$\$$\mathrm{bin}(y)$ soll die Maschine die Ausgabe $\mathrm{bin}(x \cdot y)$ berechnen.

    Ihre Maschine soll die Addiermaschine aus der Vorlesung als „Unterprogramm“ benutzen, indem Sie an geeigneten Stellen in den Zustand $q_0$ übergeht. Der Zustand $q_F$ ist dann kein Haltezustand mehr, sondern die Rechnung wird danach fortgesetzt.

    *Beschreiben Sie Ihren Algorithmus zunächst in Worten.*

[1] `http://www.inf.fu-berlin.de/lehre/SS05/GTI/flex/summe.flex`