Grundlagen der theoretischen Informatik, SS 2005 — 0. Übungsblatt

(freiwillige Vorübungen)

---

1. (0 Punkte) Die Programmiersprache SCHLEIFE hat die folgenden einfachen Regeln:

   - Es gibt einen unendlichen Vorrat an Variablen `X0`, `X1`, `X2`, `X3`, ..., die beliebige natürliche Zahlen (≥ 0) speichern können.
   - Es gibt die einfachen Anweisungen `X`i `= 0` (Inititalisierung), `X`i `= X`j (Kopieren), `X`i `= X`i `+ 1` (Hinaufzählen) und `X`i `= X`i `- 1` (Herunterzählen). (Beim Herunterzählenwird `X`i unverändert gelassen, wenn es den Wert 0 hat, weil die Variablen keine negativen Werte enthalten können.)
   - Außerdem gibt es eine *Schleifenanweisung*:

     `WIEDERHOLE X`i `MAL`
       Anweisungsfolge
     `SCHLEIFENENDE`

     Wenn die Variable `X`i zu Beginn der Ausführung den Wert n hat, wird die Schleife n-mal wiederholt, auch wenn `X`i zwischendurch geändert wird. Schleifen können auch geschachtelt sein.

   Es gibt keine Ein- und Ausgabeanweisungen. Die Eingabeparameter eines Programms stehen zu Beginn in den ersten Variablen `X0`, `X1`, ... Alle übrigen Variablen sind zu Beginn auf 0 gesetzt. Das Ergebnis des Programms ist der Wert, der am Ende in `X0` steht.

   Das folgende Programm berechnet zum Beispiel die Summe von zwei Zahlen `X0` und `X1`:

   ```
   WIEDERHOLE X1 MAL
     X0 = X0 + 1
   SCHLEIFENENDE
   ```

   (a) Schreiben Sie ein Programm zur Berechnung des Produktes von `X0` und `X1`.

   (b) Schreiben Sie ein Programm, dass den Rest der Division von `X1` durch `X0` berechnet. (Wenn `X0` $= 0$ ist soll das Ergebnis `X1` sein.)

   (c) Schreiben Sie ein Programm zur Berechnung von $2^{\texttt{X0}}$.

   (d) Schreiben Sie ein Programm, dass mit allen Variablen auf Null beginnt, aus höchstens $n = 2, 3, 4, 5, 6, 7$ Anweisungen besteht, und am Ende einen möglichst großen Wert in `X0` hat. (Dabei gilt ein `WIEDERHOLE-SCHLEIFENENDE`-Paar als eine einzige Anweisung.)

   (e) Die Programmiersprache SCHLEIFE++ hat zusätzlich bedingte Anweisungen:

     `IF X`i`>X`j `THEN`
       Anweisungsfolge
     `ENDIF`

     Wie kann mein ein solches Programm durch Eliminieren aller bedingten Anweisungen in ein SCHLEIFE-Programm übersetzen?

2. (0 Punkte) Für ein *Alphabet* Σ mit k Elementen ist Σ^* ist die Menge der Wörter (Folgen), die man aus den Buchstaben von Σ bilden kann.

   (a) Wieviele Wörter in Σ^* haben die Länge n (bestehen aus n Buchstaben)?

(b) Wieviele Palindrome in Σ^* haben die Länge n? (Ein Palindrom ist ein Wort, das von vorne und von hinten gelesen gleich ist.)

3. (0 Punkte) Durch wiederholte Anwendung des *Homomorphismus*

$$0 \to 01,\ 1 \to 0$$

auf das Ausgangswort 0 erhält man die Folge der Wörter

$$w_0 = 0,\ w_1 = 01,\ w_2 = 010,\ w_3 = 01001,\ w_4 = 01001010,\ w_5 = 0100101001001,\ \ldots$$

(a) Beschreiben Sie, was ein Homomorphismus ist, und was die Anwendunge eines Homomorphismus auf ein Wort (über einem gegebenen Alphabet) bedeutet. (Lassen Sie sich dabei von diesem Beispiel und dem Beispiel in der nächsten Aufgabe leiten, anstatt gleich im Lehrbuch nachzuschlagen.)

(b) Welche Länge hat das Wort w_n?

(c) Jedes Wort ist ein Anfangsstück (*Präfix*) des darauffolgenden Wortes, sodass die Folge gegen ein unendliches Wort w_∞ "konvergiert". Wieviele verschiedene Teilwörter (zusammenhängende Teilfolgen) der Längen $n = 1, 2, 3, 4, 5$ enthält w_∞?

(d) (*) Gegen welchen Wert konvergiert der Anteil der Einsen in w_n an der Gesamtzahl der Symbole?

4. (0 Punkte) Durch wiederholte Anwendung des Homomorphismus

$$0 \to 100,\ 1 \to 001$$

auf das Ausgangswort 0 erhält man die Folge der Wörter

$$w_0 = 0,\ w_1 = 100,\ w_2 = 001100100,\ w_3 = 100001001001100100001100100,\ \ldots$$

(a) Welche Länge hat das n-te Wort?

(b) Gegen welchen Wert konvergiert der Anteil der Einsen?

(c) Die Wörter, die mit 0 beginnen, konvergieren gegen ein unendliches Wort. Wieviele verschiedene Teilwörter der Längen $n = 1, 2, 3, 4, 5$ enthält dieses Wort?

(d) Im Gegensatz zur vorigen Aufgabe ist hier nicht jedes Wort ist eine Verlängerung des vorhergehenden Wortes. An welcher Eigenschaft des Homomorphismus mag das liegen?

Kriterien zur Erlangung eines Scheines:

1. In jeder Vorlesungswoche muss eine Übungsaufgabe gelöst und abgegeben werden. Insgesamt müssen 60 % der Gesamtpunktezahl der Übungen erreicht werden. Die Übungsaufgaben können in Zweiergruppen (oder einzeln) abgegeben werden.

2. Von jeweils zwei aufeinanderfolgenden Übungszetteln müssen mindestens je 30 % der Punkte erreicht werden. (Eine einmalige Ausnahme wird toleriert.)

3. Aktive Teilnahme an den wöchentlichen Tutorien und einmaliges Vorrechnen eines Übungsbeispiels pro Person, nach vorheriger Einteilung in einer Liste. Da die Aufgaben gemäß Punkt 1 dazu nicht ausreichen, gibt es auf den Übungsbättern zusätzliche Aufgaben für diesen Zweck (und auch einfach zum Üben).

4. 50 % der Punkte bei der Zwischenklausur und 50 % der Punkte bei der Abschlussklausur (oder bei der entsprechenden Nachklausur).

Der Schein wird benotet. Die Note ergibt sich aus den Klausuren.

Grundlagen der theoretischen Informatik, SS 2005 — 1. Übungsblatt

Abgabe bis Montag, 25. April 2005, 10:15 Uhr

---

5. (10 Punkte) Beschreiben Sie die Sprache L, die durch den regulären Ausdruck
$$0^*(0 + 10^*1)^*(\varepsilon + 0 + 00)$$
gegeben ist. (Alle Worte in L haben eine sehr charakteristische Eigenschaft. Welche?) Beweisen Sie Ihre Behauptung! Gibt es einen einfacheren regulären Ausdruck für diese Sprache?

6. (0 Punkte)
   (a) Geben Sie einen regulären Ausdruck an, der genau die Worte w über Σ mit der folgenden Eigenschaft beschreibt: „w enthält keine zwei unmittelbar aufeinanderfolgenden gleichen Symbole," und zwar für $\Sigma_2 = \{a, b\}$, $\Sigma_3 = \{a, b, c\}$, und für $\Sigma_4 = \{a, b, c, d\}$. Beweisen Sie, dass Ihr Ausdruck wirklich diese Sprache beschreibt!
   (b) Lösen Sie dieselbe Aufgabe für die folgende Eigenschaft: „Jedes Symbol in w steht neben einem gleichen Symbol." (Zum Beispiel hat das Wort $aaabbcccaa$ diese Eigenschaft.)
   (c) (schwierig) Wieviele Wörter der Länge n enthält die Sprache aus Aufgabe (b), für Σ_2 und Σ_3?

7. (0 Punkte)
   (a) Codieren Sie das Alphabet $\{a, b, \ldots, z\}$ durch Binärworte, so dass jedes codierte Wort wieder eindeutig decodierbar ist! (Eine ungeeignete Codierung wäre z.B. $a \mapsto 0, b \mapsto 01, c \mapsto 100$, etc. In diesem Fall könnte nämlich 0100 von ac oder von baa herstammen.) Zeigen Sie, dass Ihre Codierung die gewünschte Eigenschaft hat.
   (b) Das Alphabet $\Sigma = \{a, b\}$ sei durch $a \mapsto 0$ und $b \mapsto 010$ codiert. Zeigen Sie, dass diese Codierung eindeutig decodierbar ist!

8. (Zusatzfrage, 10 Punkte) Welche der folgenden Gleichungen gelten für alle regulären Ausdrücke?
   (a) $(AB)C = A(BC)$
   (b) $(A^*)^* = A^*$
   (c) $(A + B)^* = A^* + B^*$
   (d) $(A^*B^*)^* = (A + B)^*$

   Beweisen Sie Ihre Antworten.Wenn eine Beziehung nicht als Gleichung gilt, in welchen Fällen gilt dann eine Inklusionsbeziehung ($\subseteq$ oder $\supseteq$)?

9. (0 Punkte) Geben sie einen regulären Ausdruck an, der alle Wörter über $\Sigma = \{0, 1\}$ beschreibt, die 0101 nicht als Teilwort enthalten.

10. (0 Punkte) Konstruieren Sie einen regulären Ausdruck für die Sprache L_1 der Wörter über $\{0, 1\}$, bei denen auf jedes Vorkommen der Teilfolge 0011 unmittelbar die Teilfolge 1100 anschließt.

    Zum Beispiel ist $w_1 = 0001111100 \notin L_1$, $w_2 = 010110011110011111000 \in L_1$, $w_3 = 0010 \in L_1$, und $w_4 = 00111100011 \notin L_1$. Erklären Sie Ihren Ausdruck. (Ein formaler Korrektheitsbeweis ist nicht erforderlich.)

Grundlagen der theoretischen Informatik, SS 2005 — 2. Übungsblatt

Abgabe bis Montag, 2. Mai 2005, 10:15 Uhr

---

11. Kommutierende Wörter: Wieviele Paare von Wörter $a \in \Sigma^3$ und $b \in \Sigma^2$ mit der Eigenschaft $ab = ba$ gibt es? Wieviele Paare von Wörter $a \in \Sigma^6$ und $b \in \Sigma^9$ mit der Eigenschaft $ab = ba$ gibt es? Wie ist es allgemein für $a \in \Sigma^m$ und $b \in \Sigma^n$? Wodurch kann man solche Wortpaare charakterisieren?

12. Eine *Kongruenzrelation* $\sim$ auf einer Menge Σ^* von Wörtern ist eine Äquivalenzrelation mit folgender zusätzlicher Eigenschaft:
$$a \sim b \wedge c \sim d \implies ac \sim bd.$$
Welche der folgenden Relationen sind Kongruenzrelationen? Die Beziehung $a \sim b$ soll genau dann gelten, wenn folgende Bedingung erfüllt ist:
    (a) a und b haben die gleiche Anzahl von Nullen und die gleiche Anzahl von Einsen.
    (b) $a = b = \varepsilon$ oder $a \neq \varepsilon \neq b$ und a und b haben denselben Anfangsbuchstaben.
    (c) $a = \varepsilon$ oder $b = \varepsilon$ oder $a \neq \varepsilon \neq b$ und a und b haben denselben Anfangsbuchstaben.
    (d) das Teilwort 100 kommt in a und b gleich häufig vor.

13. Schreiben Sie ein Programm in Haskell (oder Java), das für einen endlichen Automaten (in einer geeigneten Darstellung) und ein gegebenes Wort entscheidet, ob es vom Automaten akzeptiert wird.

14. (a) (5 Punkte) Konstruieren Sie einen endlichen Automaten, der die durch 5 teilbaren Binärzahlen akzeptiert. Das höchstwertige Bit wird dabei zuerst eingelesen.
    (b) (5 Punkte) Konstruieren Sie einen deterministischen endlichen Automaten, der genau jene Wörter über dem Alphabet $\{a, b, c, d\}$ akzeptiert, die das Teilwort *abacabadabaca* enthalten.
    (c) (0 Punkte) Konstruieren Sie einen endlichen Automaten, der die Wörter akzeptiert, die das Teilwort *abacabadabaca nicht* enthalten.

    Erläutern Sie Ihre Automaten mit einigen Sätzen. Zeichnen Sie Zustandsdiagramme.

15. Konstruieren Sie einen äquivalenten deterministischen Automaten für den nichtdeterministischen endlichen Automaten mit $\Sigma = \{0, 1\}$, $Q = \{a, b, c\}$, $Q_0 = \{a\}$, $F = \{a\}$ und
$$\delta = \{(a, 0, a), (a, 0, b), (b, 1, b), (b, 1, c), (c, 0, c), (c, 0, a)\}.$$

16. (Zusatzaufgabe, 0 Punkte) Beweisen Sie, dass die von einem DEA akzeptierten Sprachen (die regulären Sprachen) unter Präfixbildung abgeschlossen sind: Wenn L eine solche Sprache ist, dann gibt es auch einen DEA, der
$$L' = \{\, x \mid \text{es gibt ein } y \text{ mit } xy \in L \,\}$$
akzeptiert.

17. Konstruieren Sie einen endlichen Automaten *ohne* ε-Übergänge für die Sprache
$$(a + ba)^+(b + ab)^* + (bb)^*a^+(bb)^*.$$

18. Für einen Zustand $q \in Q$ des DEA $M = (Q, \Sigma, \delta, q_0, F)$ sei die Sprache $L_q(M)$ so definiert:
$$L_q(M) = \{\, w \in \Sigma^* \mid \delta(q_0, w) = q \,\}$$
Beweisen Sie: Wenn $w_1, w_2 \in L_q(M)$ sind, dann gilt für alle $x \in \Sigma^*$:
$$w_1 x \in L(M) \iff w_2 x \in L(M)$$

Grundlagen der theoretischen Informatik, SS 2005 — 3. Übungsblatt

Abgabe bis Montag, 9. Mai 2005, 10:15 Uhr

19. (10 Punkte) L_1 sei eine reguläre Sprache über dem Alphabet $\{0,1\}$, und

$$L_2 = \{\, w_1 0 w_2 0 \dots 0 w_n \mid w = w_1 w_2 \dots w_n \in L_1,\ n \geq 1,\ w_i \in \{0,1\}\,\}$$

sei die Sprache, wo zwischen je zwei Buchstaben eine Null eingefügt ist. (Am Ende wird *keine* Null angehängt!) Beweisen Sie, dass L_2 ebenfalls regulär ist.

20. (a) Das Produkt von Automaten.
Konstruieren Sie zu zwei gegebenen DEAs $M^1 = (Q^1, \Sigma, \delta^1, q_0^1, F^1)$ und $M^2 = (Q^2, \Sigma, \delta^2, q_0^2, F^2)$ einen DEA für die Sprache $L(M^1) \cap L(M^2)$.
(b) Konstruieren Sie einen DEA für die Sprache $L(M^1) \cup L(M^2)$.

21. (a) Konstruieren Sie einen deterministischen endlichen Automaten, der die durch 5 teilbaren Binärzahlen akzeptiert. Diesmal soll das niederwertigste Bit als erstes gelesen werden.
(b) Konstruieren Sie endliche Automaten für die durch 6 teilbaren Dezimalzahlen, die (1) von der höchsten zur niedrigsten Stelle (2) von der niedrigsten zur höchsten Stelle gelesen werden.

22. Ist die Nerode-Relation einer Sprache L immer eine Kongruenzrelation? Ist die Nerode-Relation einer regulären Sprache L immer eine Kongruenzrelation?

23. Welche der folgenden Sprachen über $\Sigma = \{0,1\}$ sind regulär?
(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) $L_4 = \{\, x \mid$ die Anzahl der Nullen von x unterscheidet sich um höchstens um 1 von der Anzahl der Einsen $\}$
(e) $L_5 = \{\, x \mid$ in jedem Präfix von x unterscheidet sich die Anzahl der Nullen höchstens um 1 von der Anzahl der Einsen $\}$
(f) $L_6 = \{\, x \mid$ in jedem Präfix von x unterscheidet sich die Anzahl der Nullen höchstens um 2 von der Anzahl der Einsen $\}$

24. (Zusatzaufgabe, 0 Punkte) Für einen Homomorphismus $h\colon \Sigma^* \to \Delta^*$ und eine Sprache $L \subseteq \Sigma^*$ bezeichnet $h(L)$ die Sprache

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

Beweisen Sie, dass das Bild $h(L)$ einer regulären Sprache L unter einem Homomorphismus h regulär ist.

25. (*) Beweisen Sie: Wenn $L \in \Delta^*$ regulär ist und $h\colon \Sigma^* \to \Delta^*$ ein Homomorphismus ist, ist auch die folgende Sprache regulär:

$$h^{-1}(L) := \{\, x \in \Sigma^* \mid h(x) \in L \,\}$$

26. Welche der Sprachen $h_j(L_i)$ sind regulär, wenn L_i eine der Sprachen aus Aufgabe 23 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) $h_3(0) = 01$, $h_3(1) = \varepsilon$.

27. Konstruieren Sie den Minimalautomaten für den Automaten aus Aufgabe 15.

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 δ ist in der nebenstehenden Tabelle angegeben.

δ	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 Wechselworkung 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 $\text{bin}(x)\#\text{bin}(y)\$$ und Start im Zustand q_0 hält die Maschine im Zustand q_F mit der Ausgabe $\text{bin}(x+y)\#\text{bin}(y)\$$ und dem Kopf auf dem ersten Eingabesymbol. (Die Notation $\text{bin}(x) \in \{0, 1\}^*$ 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 $\text{bin}(x)\$\text{bin}(y)$ soll die Maschine die Ausgabe $\text{bin}(x \cdot y)$ 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`

Grundlagen der theoretischen Informatik, SS 2005 — 5. Übungsblatt

Abgabe bis Montag, 23. Mai 2005, 10:15 Uhr

34. (Zusatzaufgabe, 0 Punkte) Tabellenzugriff über Adressen, Simulation eines RAM-Speichers durch eine Turingmaschine.
Bei Eingabe des Wortes
$$x = x_1\#y_1 \,|\, x_2\#y_2 \,|\, \ldots \,|\, x_n\#y_n \,|\, x_0$$
mit $n \geq 0$, $x_k, y_k \in \{0,1\}^*$ soll die Ausgabe $x\#y$ berechnet werden. Dabei ist $y = y_i$ für den größten Index $i \leq n$ mit $x_i = x_0$, und $y = 0$, falls kein solches i existiert.

35. (a) Entwerfen Sie eine Turingmaschine, die die Sprache $\{\, 0^n 1^n 0^n \mid n \geq 0 \,\}$ akzeptiert.
    (b) Entwerfen Sie eine Turingmaschine für dieselbe Sprache, die mit dem Bandalphabet $\Gamma = \{0, 1, B\}$ auskommt.
    (Hinweis: Sie können ein größeres Bandalphabet „simulieren“, indem Sie es „binär kodieren“. Dabei wird jeweils eine feste Anzahl von Feldern des Bandes zu einer logischen Einheit zusammengefasst.)

36. Fleißige Biber (siehe auch Aufgabe 1d). Entwerfen Sie eine Turingmaschine mit 2 Zuständen (zusätzlich zum Haltezustand) und dem Bandalphabet $\Gamma = \{1, B\}$, die mit dem leeren Band beginnt, nach endlich vielen Schritten anhält, und dabei eine möglichst lange fortlaufende Folge von Einsen auf das Band schreibt.

37. Eine Funktion $f\colon \mathbb{N} \to \mathbb{N}$ heißt SCHLEIFE-berechenbar, wenn es ein Programm in der Programmiersprache SCHLEIFE aus Aufgabe 1 gibt, das bei Eingabe von n in der Variablen `X0` (und mit allen übrigen Variablen auf 0 gesetzt) mit dem Wert $f(n)$ in der Variablen `X0` terminiert.
    Beweisen Sie: Wenn die Funktionen f und g (durch Programme der Länge k_1 und k_2) SCHLEIFE-berechenbar sind, dann ist auch die Funktion $f \circ g\colon n \mapsto g(f(n))$ SCHLEIFE-berechenbar, und zwar durch ein Programm der Länge $k_1 + k_2$.

38. Überlegen Sie, wie man SCHLEIFE-berechenbare Funktionen $f\colon \mathbb{N} \to \mathbb{N}$ mit JAVA oder Haskell berechnen könnte. Diskutieren Sie, welche Möglichkeiten von JAVA oder Haskell über die Möglichkeiten der Programmiersprache SCHLEIFE wesentlich hinausgehen, sodass man offenbar nicht jedes JAVA- oder Haskell-Programm zu Berechnung einer Funktion in SCHLEIFE übersetzen kann. (Die Objektorientierung gehört nicht zu diesen wesentlichen Möglichkeiten.)

39. (Zusatzaufgabe, 5 Punkte) Wenn die Taste $\boxed{\texttt{-}}$ auf der Tastatur kaputt ist, steht der Befehl `X`i `= X`i `- 1` zum Herunterzählen in der Programmiersprache SCHLEIFE nicht zur Verfügung. Wie kann man diesen Befehl durch die übrigen Befehle simulieren?

40. (5 Punkte) Wie kann man für zwei gegebene reguläre Ausdrücke entscheiden, ob sie dieselbe Sprache darstellen? Skizzieren Sie einen Lösungsalgorithmus in groben Zügen. Dabei dürfen Sie Ergebnisse aus der Vorlesung und von anderen Übungsaufgaben verwenden.

41. Beweisen oder widerlegen Sie:
    (a) Das Komplement einer entscheidbaren Sprache ist entscheidbar.
    (b) Die Vereinigung zweier entscheidbarer Sprache ist entscheidbar.
    (c) Das Komplement einer unentscheidbaren Sprache ist unentscheidbar.
    (d) Die Vereinigung zweier unentscheidbarer Sprache ist unentscheidbar.

Grundlagen der theoretischen Informatik, SS 2005 — 6. Übungsblatt

Abgabe bis Montag, 30. Mai 2005, 10:15 Uhr

42. Welche der beiden folgenden Aussagen sind wahr? Beweisen oder widerlegen Sie sie.
    (a) Wenn L_1 und L_2 entscheidbar sind, dann ist auch $L_1 L_2$ entscheidbar.
    (b) Wenn L_1 und L_2 rekursiv aufzählbar sind, dann ist auch $L_1 L_2$ rekursiv aufzählbar.

43. (10 Punkte) $D = \{ w_i \mid M_i \text{ akzeptiert } w_i \text{ nicht} \}$ bezeichnet die Diagonalsprache und $\bar{D}$ ihr Komplement.
    (a) Zeigen Sie, dass die Sprache $\bar{D}$ rekursiv aufzählbar ist.
    (b) Zeigen Sie, dass die Sprache $0D \cup 1\bar{D}$ nicht rekursiv aufzählbar ist.

44. (2 Zusatzpunkte) Eine andere Konstruktion einer unentscheidbaren Sprache.

$$U' := \{ \langle M \rangle w \mid M \text{ akzeptiert das Wort } \langle M \rangle w \text{ nicht} \}$$

    (Der Beweis geht ähnlich wie bei der Diagonalsprache D.)

45. Welche der folgenden Sprachen sind rekursiv, welche sind rekursiv aufzählbar?
    (a) $L_1 = \{ 0^k \langle M \rangle w \mid M \text{ akzeptiert } w \text{ nach höchstens } k \text{ Schritten.} \}$
    (b) $L_2 = \{ 0^k \langle M \rangle w \mid M \text{ hat } w \text{ nach } k \text{ Schritten noch nicht akzeptiert.} \}$
    (c) $L_3 = \Sigma^* - (L_1 \cup L_2)$
    (d) $L_4 = \{ 0^k \langle M \rangle w \mid M \text{ akzeptiert } w \text{ nach mehr als } k \text{ Schritten.} \}$

46. Beweisen Sie, dass die Sprache $L = \{ 0^i \mid i \geq 0, \text{ das Teilwort } 0^i 123 \text{ kommt in der Dezimaldarstellung von } \pi \text{ unendlich oft vor} \}$ entscheidbar ist. (Sie müssen dazu nichts über π wissen, nur logisch überlegen können. Wie kann L aussehen?)

47. Welche der folgenden Aussagen sind wahr, und welche sind falsch? Welche Aussagen sind trivial, und welche sind absurd?
    (a) Jede Teilmenge einer rekursiven Sprache ist rekursiv.
    (b) Jede Teilmenge einer rekursiv aufzählbaren Sprache ist rekursiv aufzählbar.
    (c) Die Vereinigung zweier entscheidbarer Sprachen ist entscheidbar.
    (d) Die Vereinigung zweier rekursiv aufzählbarer Sprachen ist rekursiv aufzählbar.
    (e) Die Schnittmenge zweier entscheidbarer Sprachen ist entscheidbar.
    (f) Die Schnittmenge zweier rekursiv aufzählbarer Sprachen ist rekursiv aufzählbar.
    (g) Die Schnittmenge einer rekursiv aufzählbaren Sprache und einer entscheidbaren Sprache ist rekursiv aufzählbar.
    (h) Jede reguläre Sprache ist rekursiv aufzählbar.
    (i) Jede Teilmenge einer regulären Sprache ist regulär.

48. Zeigen Sie:
    (a) Die rekursiven Sprachen sind unter Differenz abgeschlossen. (Das heißt: Mit L_1 und L_2 ist auch $L_1 - L_2$ immer rekursiv.)
    (b) Die rekursiv aufzählbaren Sprachen sind unter Differenz nicht abgeschlossen.

Grundlagen der theoretischen Informatik, SS 2005 — 7. Übungsblatt

Abgabe bis Montag, 13. Juni 2005, 10:15 Uhr

---

49. Ist es entscheidbar, ob eine Turingmaschine das leere Wort akzeptiert? Ist es entscheidbar, ob eine Turingmaschine alle Wörter (also Σ^*) akzeptiert?

50. Lösen Sie das Post'sche Korrespondenzproblem über einem einelementigen Alphabet.

51. Lösen Sie das Post'sche Korrespondenzproblem für folgende Eingaben:

    (a) $x_1, \ldots, x_5 = abb, a, bab, baba, aba$; $y_1, \ldots, y_5 = bbab, aa, ab, aa, a$.

    (b) $x_1, \ldots, x_5 = a, baa, abab, aabb, abb$; $y_1, \ldots, y_5 = ab, ba, bbaa, bbba, bba$.

    (c) $x_1, \ldots, x_{10} = d, de, deh, deh, deh, deha, deha, wa, wa, wah$; $y_1, \ldots, y_{10} = add, e, de, d, deha, eh, d, w, wah, wa$; Welches sind die zwei kürzestes Lösungen?

    (d) (zum Tüfteln für alle, denen das Post'sche Korrespondenzproblem zu leicht vorkommt) $x_1, \ldots, x_4 = 001, 01, 01, 10$; $y_1, \ldots, y_4 = 0, 011, 101, 001$.

52. Zeigen Sie, dass es unendlich viele Lösungen gibt, wenn das Post'sche Korrespondenzproblem eine Lösung hat.

53. (a) (5 Punkte) Die algorithmische Reduzierbarkeit zwischen Problemen ist transitive: Aus $A < B$ und $B < C$ folgt $A < C$. Ist sie eine Ordnungsrelation?

    (b) (5 Punkte) Beweisen Sie: Wenn A ein beliebiges entscheidbares Problem ist, so ist es auf das Halteproblem H reduzierbar: $A < H$.

54. Wie kann man jede Turingmaschine *ohne Zeitverlust* durch eine Turingmaschine simulieren, die sich nie über das erste Eingabezeichen nach links hinausbewegt?

55. (5 Punkte) Wenn eine Turingmaschine $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$ ein Wort w in t Schritten akzeptiert und sie dabei nur die ersten s Felder des Bandes besucht und sich nie über das erste Eingabezeichen nach links hinausbewegt, dann ist $t \leq |Q| \cdot s \cdot |\Gamma|^s$.

56. Betrachten Sie die „Fleißige-Biber-Funktion“ für Programme in der Programmiersprache SCHLEIFE aus Aufgabe 1d, eingeschränkt auf Programme, in denen die Schleifenanweisungen höchstens bis zu einer vorgegebenen Tiefe k geschachtelt sind.

    (a) Beweisen Sie, dass diese Funktion berechenbar ist.

    (b) Beweisen Sie, dass diese Funktion für ein festes $k \geq 1$ nicht mit einem SCHLEIFE-Programm der Schachtelungstiefe k berechnet werden kann.

    (c) Was ist die Fleißige-Biber-Funktion für $k = 0$ (Programme ohne Schleife)? Was ist das kürzeste Programm zur Berechnung dieser Funktion?

57. Entwerfen Sie eine kontextsensitive Grammatik für die Sprache $\{\, a^n b^n c^n \mid n \geq 1 \,\}$.

58. Der erste Satz dieser Aufgabe ist ein Beispiel eines grammatikalisch korrekten Satzes. Analysieren Sie diesen Satz gemäß der nachstehenden rudimentären Grammatik:

    S $\to$ NG VG, NG $\to$ NG Att, NG $\to$ Art NG$'$, NG $\to$ NG$'$, NG$'$ $\to$ Adj NG$'$, NG$'$ $\to$ N, NG $\to$ Pro, Att $\to$ NG, Att $\to$ Präp NG, VG $\to$ V, VG $\to$ V Obj, VG $\to$ V Pr, VG $\to$ VG Erg, Obj $\to$ NG, Erg $\to$ Präp NG, Pr $\to$ Adj, Pr $\to$ NG, Adj $\to$ Adv Adj.

    (S = Satz, NG = Nominalgruppe, NG$'$ = Nominalgruppe ohne Artikel, VG = Verbgruppe, Att = Attribut, Art = Artikel, Adj = Adjektiv, Adv = Adverb, N = Hauptwort (Nomen), Pro = Pronomen, Obj = Objekt, Erg = Modalobjekt, Pr = Prädikat.)

    Warum passt der zweite Satz dieser Aufgabe nicht zu dieser Grammatik?

Grundlagen der theoretischen Informatik, SS 2005 — 8. Übungsblatt

Abgabe bis Montag, 20. Juni 2005, 10:15 Uhr

59. (a) (10 Punkte) Geben Sie eine kontextfreie Grammatik für arithmetische Ausdrücke über dem Alphabet $\{+, *, (,), Z\}$ an. Dabei sollen nur Ausdrücke *ohne überflüssige Klammern* erzeugt werden, d. h., solche Audrücke, die nach den üblichen mathematischen Vorrangregeln einen anderen Wert bekommen, wenn man Klammern weglässt. Zum Beispiel sind die Klammern in $Z * (Z + Z)$ notwendig, weil man den Ausdruck ohne Klammern $Z*Z+Z$ als $(Z*Z)+Z$ lesen würden. Aus diesem Grund sind auch die Klammern im Ausdruck $(Z * Z) + Z$ überflüssig (redundant).

    (b) (0 Punkte)
    Lösen Sie die Aufgabe mit dem erweiterten Alphabet $\{+, -, *, \div, (,), Z\}$.

60. Die Grammatik

$$\begin{aligned}
S &\to T \mid T+T \mid T-T \\
T &\to F \mid F*F \mid F \div F \\
F &\to V \mid (S) \\
V &\to Z \mid ZV \\
Z &\to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}$$

für arithmetische Ausdrücke mit konstanten ganzzahlichen Operanden ist mehrdeutig, weil zum Beispiel für den Ausdruck $3-4+5$ aus der Grammatik nicht hervorgeht, ob $4 + 5$ oder $3 - 4$ zuerst berechnet werden soll. Die Mehrdeutigkeit wird durch die gängige Regel aufgelöst, dass Operatoren gleicher Stufe von links nach rechts ausgewertet werden.

Schreiben Sie eine äquivalente eindeutige Grammatik, die diese Regel widerspiegelt.

61. Konstruieren Sie eine kontextfreie Grammatik für die Sprache der Wörter über $\{0, 1\}$, die *kein* Palindrom sind:

$$L = \{ w \mid w \neq w^R \}$$

62. (a) Beweisen Sie, dass die Grammatik G mit den Regeln $S \to aS \mid SS \mid aSbS \mid \varepsilon$ genau die Wörter über $\{a, b\}$ erzeugt, in denen jeder Präfix mindestens so viele a's wie b's enthält.

    (b) Schreiben Sie ein entsprechendes Gleichungssystem für die vom der Grammatik akzeptierte Sprache L_S auf.

    (c) $L_S = \{a, b\}^*$ ist eine Lösung dieses Gleichungssystems. Finden Sie eine Lösung, die von $L(G)$ und $\{a, b\}^*$ verschieden ist.

    (d) Schreiben Sie eine Grammatik ohne ε-Regeln für die Sprache $L(G) - \{\varepsilon\}$.

63. Wandeln Sie die folgende Grammatik in Chomsky-Normalform um. Prüfen Sie dann mit dem CYK-Algorithmus nach, dass das Wort $a + a * a * a$ von dieser Grammatik erzeugt wird, und geben Sie einen Syntaxbaum *und* eine Linksableitung (für die umgewandelte Grammatik) an.

$$\begin{aligned}
S &\to S * A \mid A - A \\
A &\to A * A \mid a
\end{aligned}$$

Grundlagen der theoretischen Informatik, SS 2005 — 9. Übungsblatt

Abgabe bis Montag, 27. Juni 2005, 10:15 Uhr

64. (a) L_1 sei eine kontextfreie Sprache über dem Alphabet $\{0, 1\}$, und

$$L_2 = \{\, w_1 0 w_2 0 \dots w_n 0 \mid w_1 w_2 \dots w_n \in L_1,\ n \geq 0 \,\}$$

sei die Sprache, wo nach jedem Buchstaben jedes Wortes eine Null eingefügt ist. Beweisen Sie, dass L_2 ebenfalls kontextfrei ist.

(b) Lösen Sie die gleiche Aufgabe für die Sprache

$$L_3 = \{\, w_1 0 w_2 1 w_3 0 w_4 1 w_5 0 \dots \mid w_1 w_2 \dots w_n \in L_1,\ n \geq 0 \,\},$$

wo nach jedem Buchstaben abwechselnd eine Null und eine Eins eingefügt wird.

65. Wenn man beim Beweis des Pumpinglemmas den *längsten* Pfad im Ableitungsbaum betrachtet, kann man die stärkere Aussage herleiten, dass es eine Zerlegung $w = xyzuv$ mit der *Längenbeschränkung* $|yzu| \leq n_0$ gibt, sodass all Wörter $xy^i zu^i v \in L$ sind.

66. (10 Punkte) Beweisen Sie, dass die folgenden Sprachen *über dem Alphabet* $\{a, b, c\}$ nicht kontextfrei sind.

(a) $L_1' = \{\, w \mid w$ enthält doppelt so viele a's wie b's und doppelt so viele b's wie c's $\}$

(b) $L_2 = \{\, a^i b^j c^k \mid i > j \geq k \geq 1 \,\}$

67. Schreiben Sie eine kontextfreie Grammatik für die regulären Ausdrücke über dem Alphabet $\{0, 1\}$. Sie können annehmen, dass die Ausdrücke voll geklammert sind, entsprechend der ursprünglichen Definition regulärer Ausdrücke, zum Beispiel

$$(((0 + (1)^*)) \cdot ((0 + (0 \cdot 1)) + \varepsilon))$$

68. Kontextfreie Sprachen sind unter Umkehrung (Spiegelung) abgeschlossen.

69. Entfernen Sie überflüssige Variablen und Regeln aus dieser Grammatik:

$S \to AB \mid CA$	$B \to BC \mid AB \mid BDE$	$D \to ADEba \mid ACA$
$A \to a$	$C \to aB \mid c$	$E \to DD \mid BA$

70. (0 Punkte) Übersetzen Sie den folgenden vereinfachten Ausschnitt aus der Syntax der Programmiersprache Java aus der EBNF (extended Backus-Naur-Form) in entsprechende kontextfreie Regeln:

TryBlock	::= **try** Block Catches | **try** Block [Catches] **finally** Block
Catches	::= CatchClause { CatchClause }
CatchClause	::= **catch** **(** Exception Identifier **)** Block
Exception	::= Identifier

Die Terminalsymbole sind hier fett gedruckt. Sie können die Variablen durch passende einbuchstabige Namen abkürzen. (Wozu dient eigentlich die Variable „Exception“, die man doch genausogut zusammen mit der Regel „Exception ::= Identifier“ eliminieren könnte?)

71. (a) Bleibt eine eindeutige Grammatik bei der Transformation in Chomsky-Normalform eindeutig?

(b) G sei eine Grammatik in Chomsky-Normalform, die eine endliche Sprache erzeugt. Finden Sie eine Schranke für das längste Wort in $L(G)$.

(c) Wie lang kann die Ableitung eines Wortes w bei einer Grammatik in Chomsky-Normalform höchstens sein?

Grundlagen der theoretischen Informatik, SS 2005 — 10. Übungsblatt

Abgabe bis Montag, 4. Juli 2005, 10:15 Uhr

---

72. (a) Zeigen Sie, dass die Sprache

$$L_1 = \{\, w \in \{a, b\}^* \mid w \text{ enthält gleich viele } a\text{'s wie } b\text{'s}\,\}$$

deterministisch kontextfrei ist.

(b) Geben Sie eine kontextfreie Grammatik für die Sprache L_1 an.

(c) Geben Sie eine kontextfreie Grammatik für die Sprache

$$L_2 = \{\, w \mid w \text{ enthält doppelt so viele } a\text{'s wie } b\text{'s}\,\}$$

über dem Alphabet $\Sigma = \{a, b\}$ an.

(d) Lösen Sie dieselbe Aufgabe über dem Alphabet $\{a, b, c\}$.

73. (a) Schreiben Sie eine kontextfreie Grammatik für die durch den regulären Ausdruck

$$(((0 + (1)^*)) \cdot ((0 + (0 \cdot 1)) + \varepsilon))$$

definierte Sprache.

(b) Wie kann man allgemein für einen regulären Ausdruck eine entsprechende kontextfreie Grammatik erzeugen?

74. (a) (6 Punkte) Wie kann man entscheiden, ob eine kontextfreie Grammatik eine endliche Sprache erzeugt?

(b) (4 Punkte) Kann man für eine gegebene kontextfreie Grammatik G und eine durch einen endlichen Automaten gegebene reguläre Sprache L_1 entscheiden, ob $L(G) = L_1$ ist?

(c) (0 Punkte) Wie kann man für eine gegebene kontextfreie Grammatik G und eine durch einen endlichen Automaten gegebene reguläre Sprache L_1 entscheiden, ob $L(G) \subseteq L_1$ ist? (Hier müssen Sie mehrere bekannte Ergebnisse kombinieren.)

75. Das Post'sche Korrespondenzproblem und kontextfreie Sprachen.

(a) Für gegebene $x_1, \ldots, x_k \in \Sigma^*$ ist die Sprache $\{\, a_{i_n} a_{i_{n-1}} \ldots a_{i_1} x_{i_1} x_{i_2} \ldots x_{i_n} \mid n \geq 1,\ 1 \leq i_j \leq k \,\}$ kontextfrei, wobei $a_1, a_2, \ldots, a_k$ neue Buchstaben sind, die nicht zu Σ gehören. (Versuchen Sie, mit möglichst wenigen Regeln auszukommen.)

(b) Zu jedem Post'schen Korrespondenzproblem kann man zwei kontextfreie Grammatiken G_1 und G_2 konstruieren, sodass das Post'schen Korrespondenzproblem genau dann eine Lösung hat, wenn $L(G_1) \cap L(G_2) \neq \emptyset$.
Was folgt daraus für das entsprechende Entscheidungsproblem für kontextfreie Grammatiken?

76. Überprüfen Sie mit dem CYK-Algorithmus, ob das Wort $w = cdccd$ von der folgenden Grammatik erzeugt wird, und geben Sie gegebenfalls eine Ableitung für w an.

$$\begin{array}{lll} S \to AB \mid CD \mid AS & B \to SB \mid d & D \to CD \mid d \\ A \to AS \mid DC & C \to CA \mid c & \end{array}$$

77. Zeigen Sie, dass die Chomsky-Hierarchie echt ist: Geben Sie eine Typ-0-Sprache an, die keine Typ-1-Sprache ist. Geben Sie eine Typ-1-Sprache an, die keine Typ-2-Sprache ist. Geben Sie eine Typ-2-Sprache an, die keine Typ-3-Sprache ist. (Und geben Sie eine Sprache an, die keine Typ-0-Sprache ist.)

78. Beschrieben Sie, wie man zu einem regulären Ausdruck einen regulären Ausdruck für die komplementäre Sprache findet.

Grundlagen der theoretischen Informatik, SS 2005 — 11. Übungsblatt

freiwillige Abgabe bis Montag, 11. Juli 2005, 10:15 Uhr

79. (a) Wandeln Sie den folgenden Kellerautomaten in einen äquivalenten Kellerautomaten mit nur einem einzigen Zustand um. $\delta = \{(q_0, 0, \gamma \mid q_1, Z_0\gamma), (q_0, 1, \gamma \mid q_1, Z_1\gamma), (q_1, 0, \gamma \mid q_0, Z_0\gamma), (q_1, 1, \gamma \mid q_1, Z_1Z_1), (q_0, 1, Z_1 \mid q_0, \varepsilon), (q_0, 0, Z_0 \mid q_0, \varepsilon)\}$. Dabei steht γ jeweils für ein beliebiges Kellersymbol $\gamma \in \Gamma = \{Z_0, Z_1\}$.

    (b) Konstruieren Sie eine entsprechende kontextfreie Grammatik.

80. Beweisen Sie, dass die kontextfreien Sprachen abgeschlossen gegenüber der Vereinigung mit regulären Sprachen sind.

81. (a) Erweitern Sie das Haskell-Programm aus der Vorlesung zur Simulation eines deterministischen Zweiweg-Kellerautomaten und zur Berechnung der Entladefunktion[1] so, dass es die Anzahl der Schritte ausgibt, die der Automat bis zum Leeren des Kellers macht.

    (b) Erweitern Sie das Programm so, dass es auch dann terminiert, wenn der Automat in eine unendliche Schleife gerät.[2]

82. Betrachten Sie die deterministischen Zweiweg-Kellerautomaten mit dem Eingabealphabet $\Sigma = \{a\}$, dem Kelleralphabet $\Gamma = \{0, 1, Z_0\}$ und der Übergangsfunktion $\delta\colon Q \times (\Sigma \cup \{\%, \$\}) \times \Gamma \to Q \times \Gamma^* \times \{L, N, R\}$.

    $\delta(q_0, a, \gamma) = (q_0, 0\gamma, R)$, für $\gamma \in \Gamma$.
    $\delta(q_0, \$, \gamma) = (q_1, \gamma, N)$, für $\gamma \in \Gamma$.
    $\delta(q_1, x, 1) = (q_1, \varepsilon, L)$, für $x = 0$ oder $x = \$$.
    $\delta(q_1, x, 0) = (q_0, 1, N)$, für $x = 0$ oder $x = \$$.
    $\delta(q_1, x, Z_0) = (q_1, \varepsilon, N)$, für $x = 0$ oder $x = \$$.

    Die übrigen Werte von δ (für $x = \%$) sind beliebig.

    (a) Beschreiben Sie das Verhalten des Automaten bei Eingabe des Wortes a^n, für $n = 0, 1, 2, 3$ und für allgemeines n.

    (b) Bestimmen Sie die Entladefunktion $E\colon Q \times \{0, \dots, n+1\} \times \Gamma \to Q \times \{0, \dots, n+1\}$ bei Eingabe des Wortes a^n.

83. (a) Beschreiben Sie formal, wie man einen nichtdeterministischen Zweiweg-Kellerautomaten, der mit akzeptierenden Zuständen arbeitet, in einen äquivalenten Zweiweg-Kellerautomaten umwandelt, der mit leerem Keller akzeptiert.

    (b) Beschreiben Sie die Transformation in die umgekehrte Richtung.

84. Mit deterministischen Zweiweg-Kellerautomaten kann man alle deterministisch kontextfreien Sprachen akzeptieren.

85. (4 Zusatzpunkte) Konstruieren Sie (a) eine Grammatik, (b) einen Kellerautomaten für die Sprache $\{ ww^R \mid w \in \{0,1\}^* \}$.

86. Beweisen Sie: Wenn eine Sprache von einem deterministischen Kellerautomaten (mit akzeptierenden Zuständen) akzeptiert wird, dann gibt es eine eindeutige kontextfreie Sprache für diese Grammatik.

87. Sind die Sprachen, die von deterministischen Zweiweg-Kellerautomaten akzeptiert werden, abgeschlossen gegenüber Umkehrung (Spiegelbild)?

[1] `http://www.inf.fu-berlin.de/lehre/SS05/GTI/2Weg-PDA-Teilwort.hs`
[2] `http://www.inf.fu-berlin.de/lehre/SS05/GTI/2Weg-PDA-Zaehler.hs`