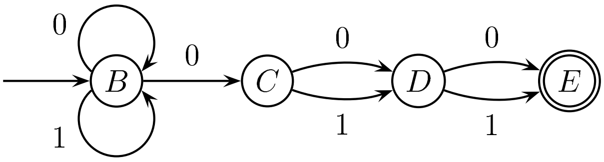
Prüfungsklausur Grundkurs Theoretische Informatik
3. + 4. Semester, Studienjahr 2000/2001 Leipzig, den 24.09.2001
Bitte pro Aufgabe ein separates Blatt! Namen und Matrikelnummer auf jedem Blatt angeben!

---

1) Der folgende Automat A hat Startzustand B und akzeptierenden Zustand E: $\boxed{6}$



- Beschreiben Sie $L(A)$ durch einen regulären Ausdruck.
- Erzeugen Sie aus A mittels Potenzmengenkonstruktion einen äquivalenten vollständigen deterministischen Automaten.
- Untersuchen Sie, ob dieser deterministische Automat minimal ist. (Antwort Ja/Nein mit kurzer Begründung genügt.)

---

2) Betrachten Sie die Sprache $L = \{w \in \Sigma^\star : |w|_0 \leq |w|_1\}$ über dem Alphabet $\Sigma = \{0, 1\}$. $\boxed{7}$
Dabei ist $|w|_x$ die Anzahl der Positionen in w, auf denen der Buchstabe x steht.

- Bestimmen Sie die Sprache $L' = L \cap 0^\star 1^\star$.
- Zeigen Sie, dass aus $L \in \mathrm{REG}$ auch $L' \in \mathrm{REG}$ folgt.
- Zeigen Sie mit einem Pumping-Lemma, dass $L' \notin \mathrm{REG}$.
- Welchen maximalen Chomsky-Typ hat L'?
- Geben Sie eine entsprechende Grammatik für L' an.

---

3) Zeigen Sie, dass jede aufzählbare unendliche Menge M von natürlichen Zahlen eine entscheidbare unendliche Teilmenge $T \subseteq M$ enthält. $\boxed{5}$
Folgen Sie diesem Ansatz: Wenn $m : \mathbb{N} \to M$ eine Funktion ist, die M aufzählt, also $M = \{m(0), m(1), \ldots\}$, dann wählen wir T als die Elemente einer monoton steigenden Teilfolge von $(m(0), m(1), \ldots)$. Beispiel: wenn $m(0) = 5$, $m(1) = 2$, $m(2) = 7$, $m(3) = 0$, $m(4) = 2$, $m(5) = 9$, dann $T = \{5, 7, 9, \ldots\}$.

- Beweisen Sie, dass die so konstruierte Menge T unendlich ist. Beachten Sie, dass (wie im Beispiel) die Funktion m nicht notwendig monoton oder injektiv ist.
- Begründen Sie, dass die so konstruierte Menge T entscheidbar ist, indem Sie einen Algorithmus angeben, der bei Eingabe x feststellt, ob $x \in T$.
- Beweisen Sie, dass Ihr Algorithmus sowohl (a) im Fall $x \in T$ als auch (b) im Fall $x \notin T$ hält und die richtige Antwort liefert.

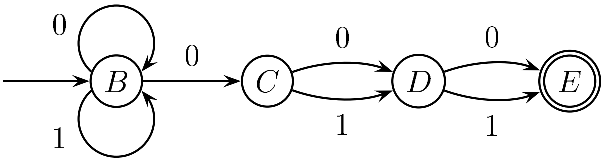
---

4) Reduktion: $\boxed{7}$

- Definieren Sie die Probleme *Handelsreisender* (traveling salesman, TSP) und *Hamiltonkreis* (HC) jeweils als Entscheidungsproblem (Paar von Eingabe und Frage).
- Definieren Sie die Menge NPH der NP-harten Probleme sowie die Menge NPC der NP-vollständigen Probleme. Schreiben Sie die Antwort *als Formel*, also zum Beispiel $\mathrm{NPH} = \{P : \forall Q \in \ldots\}$ Benutzen Sie dabei die Begriffe $\leq_P$ und NP (ohne diese zu definieren).
- Beweisen Sie, dass $\mathrm{TSP} \in \mathrm{NPC}$.
  Setzen Sie dabei $\mathrm{HC} \in \mathrm{NPC}$ als bekannt voraus. Sie müssen (unter anderem!) eine geeignete Reduktion durchführen, bei der Sie aus einem Graphen mit (bzw. ohne) Hamiltonkreis eine Kostenmatrix mit (bzw. ohne) preiswerter Rundreise konstruieren.

Lösungsideen

1) Gegeben ist der Automat:

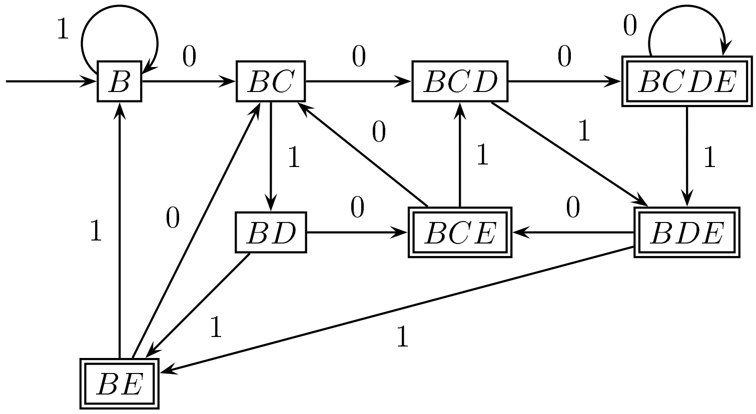


mit der Zustandsmenge $Q = \{B, C, D, E\}$ und dem Eingabealphabet $\Sigma = \{0, 1\}$.

- Es gilt $L(A) = (0 + 1)^\star 0(0 + 1)(0 + 1) = (0 + 1)^\star 0(0 + 1)^2$, also ist $L(A)$ die Sprache aller Wörter über $\{0, 1\}$, deren drittletztes Zeichen 0 ist (und die also auch mindestens drei Zeichen enthalten).
- Wenn $\delta : \Sigma \times Q \to 2^Q$ die Übergangsfunktion von A und $\hat{\delta} : \Sigma \times 2^Q \to 2^Q$ deren Erweiterung auf 2^Q ist, dann gilt:

$$
\begin{array}{rcl@{\qquad}rcl}
\hat{\delta}(0, \{B\}) & = & \{B, C\} & \hat{\delta}(1, \{B, C, D\}) & = & \{B, D, E\} \\
\hat{\delta}(1, \{B\}) & = & \{B\} & \hat{\delta}(0, \{B, D\}) & = & \{B, C, E\} \\
\hat{\delta}(0, \{C\}) & = & \{D\} & \hat{\delta}(1, \{B, D\}) & = & \{B, E\} \\
\hat{\delta}(1, \{C\}) & = & \{D\} & \hat{\delta}(0, \{B, C, D, E\}) & = & \{B, C, D, E\} \\
\hat{\delta}(0, \{D\}) & = & \{E\} & \hat{\delta}(1, \{B, C, D, E\}) & = & \{B, D, E\} \\
\hat{\delta}(1, \{D\}) & = & \{E\} & \hat{\delta}(0, \{B, D, E\}) & = & \{B, C, E\} \\
\hat{\delta}(0, \{E\}) & = & \emptyset & \hat{\delta}(1, \{B, D, E\}) & = & \{B, E\} \\
\hat{\delta}(1, \{E\}) & = & \emptyset & \hat{\delta}(0, \{B, C, E\}) & = & \{B, C\} \\
\hat{\delta}(0, \{B, C\}) & = & \{B, C, D\} & \hat{\delta}(1, \{B, C, E\}) & = & \{B, D\} \\
\hat{\delta}(1, \{B, C\}) & = & \{B, D\} & \hat{\delta}(0, \{B, E\}) & = & \{B, C\} \\
\hat{\delta}(0, \{B, C, D\}) & = & \{B, C, D, E\} & \hat{\delta}(1, \{B, E\}) & = & \{B\}
\end{array}
$$

  Damit kann der vollständige deterministische Automat einfach erzeugt werden. Das Transitionsdiagramm hat die Gestalt:



- Dieser Automat ist minimal, wie man einfach nachrechnen kann. Das ist auch plausibel, da es 2^3 mögliche Wortendungen gibt, die alle im Automaten kodiert werden müssen. Gleiches gilt übrigens ganz allgemein für Sprachen über zweielementigen Alphabeten, deren n-letztes Symbol fixiert ist: deterministische vollständige Automaten zur Akzeption solcher Sprachen haben immer mindestens 2^n Zustände. Das beantwortet auch die Frage, ob bei Umwandlung eines nichtdeterministischen in einen deterministischen Automaten die Anzahl der Zustände exponentiell wachsen kann mit: ja. (Entsprechende nichtdeterministische Automaten kommen nach obigem Muster mit $n + 1$ Zuständen aus.)

2) Gegeben ist $L = \{w \in \Sigma^\star : |w|_0 \leq |w|_1\}$ mit $\Sigma = \{0, 1\}$.

- Es gilt $L' = L \cap 0^\star 1^\star = \{0^a 1^b : 0 \leq a \leq b\}$, also ist L' die Sprache aller der Wörter über Σ, die mindestens so viele Zeichen 1 wie Zeichen 0 enthalten und bei denen alle Zeichen 0 vor allen Zeichen 1 stehen.
- REG ist abgeschlossen bei Durchschnittsbildung, da es abgeschlossen ist bei Vereinigung (nach Definition von regulären Ausdrücken) und bei Komplementbildung (End- und Nicht-Endzustände des entsprechenden Automaten vertauschen) und weil $M \cap N = \overline{\overline{M} \cup \overline{N}}$. Damit und weil $0^\star 1^\star \in \mathrm{REG}$ folgt aus $L \in \mathrm{REG}$ auch $L' \in \mathrm{REG}$.
- Angenommen $L' \in \mathrm{REG}$. Dann gibt es nach dem Pumping-Lemma für reguläre Sprachen eine solche Zahl n, dass es für alle Wörter $w \in L$ mit $|w| \geq n$ eine Zerlegung $w = xyz$ gibt mit $|xy| \leq n$, $|y| > 0$ und $\forall i \in \mathbb{N} : xy^i z \in L$.
  Sei also dieses n bekannt und sei $w = 0^n 1^n$. Dann gelten $z = 1^n$ wegen $|xy| \leq n$, $y = 0^k$ mit $k > 0$ wegen $|y| > 0$ und $x = 0^{n-k}$ wegen $w = xyz$. Damit gilt $xy^2 z = 0^{n-k} 0^{2k} 1^n = 0^{n+k} 1^n \notin L'$. Das ist ein Widerspruch zu $xy^i z \in L$, die Annahme war also falsch und es gilt $L' \notin \mathrm{REG}$.
- L' hat einen maximalen Chomsky-Typ 2 und
- eine entsprechende Grammatik für L' lautet: $S \to 0S1 \mid S1 \mid \varepsilon$.

3) Präzisierung der Bildungsvorschrift für T:

1. Zähler r mit $r := 0$ initialisieren.
2. $m(r)$ in T aufnehmen.
3. Zuletzt aufgenommenes Element $m(r)$ in z speichern, $z := m(r)$.
4. Zähler r mit $r := r + 1$ erhöhen.
5. Wenn $m(r) > z$ gilt, dann weiter bei Punkt 2, sonst weiter bei Punkt 4.

Hier wird eine streng monoton wachsende Teilfolge aus $(m(0), m(1), \ldots)$ in T übernommen.

- Angenommen T ist nicht unendlich, also $T = \{t_1, t_2, \ldots, t_k\}$. Nach Bildungsvorschrift bedeutet das $\exists n : \forall n' > n : m(n') \leq m(n) = t_k$, die Aufzählung m von M liefert also ab einer gewissen Stelle keine Elemente mehr, die grösser als t_k sind. Das bedeutet, dass t_k ein globales Maximum von m ist, da bei Bildung von T alle Elemente aus $(m(0), m(1), \ldots)$ betrachtet werden. Das heisst, dass Range m endlich ist. Aber es gilt Range $m = M$ und M ist unendlich, ein Widerspruch, also war die Annahme falsch und T ist unendlich.
- Ein Algorithmus, der für ein gegebenes x entscheidet, ob $x \in T$ gilt:
  1. Zähler r mit $r := 0$ initialisieren.
  2. $y := m(r)$ berechnen.
  3. Wenn $x = y$, dann "ja" ausgeben und beenden.
  4. Wenn $x < y$, dann "nein" ausgeben und beenden.
  5. Zähler r mit $r := r + 1$ erhöhen und weiter bei Punkt 2.
- (a) Sei $x \in T$. Dann gilt $x = m(k)$ und $k' < k \to m(k') < m(k)$ nach Bildungsvorschrift von T, also tritt für $r < k$ weder Bedingung 3 noch Bedingung 4 ein, für $r = k$ tritt Bedingung 3 ein, der Algorithmus gibt "ja" aus und terminiert.
  (b) Sei nun $x \notin T$. Dann gibt es k mit $m(k) > x$, da Range m unendlich. Sei k ausserdem minimal, also $k' < k \to m(k') \leq x$ (gibt es, da $\langle \mathbb{N}, < \rangle$ wohlgeordnet), dann tritt für $r < k$ die Bedingung 4 nie ein. Angenommen Bedingung 3 tritt für $r' < k$ ein, also $m(r') = x$. Sei r' ebenfalls minimal, also $r'' < r' \to m(r'') < m(r')$, dann ist $m(r') = x$ nach Bildungsvorschrift in T. Das ist ein Widerspruch zu $x \notin T$ und also tritt für $r < k$ weder Bedingung 3 noch Bedingung 4 ein, für $r = k$ tritt Bedingung 4 ein, der Algorithmus gibt "nein" aus und terminiert.

4) Reduktion:

- **TSP** Eingabe:
  - Gerichteter Graph $G = (V, E)$,
  - kantenbewertende Funktion $w : E \to \mathbb{N}$ und
  - eine Zahl $k \in \mathbb{N}$.

  Frage: Gibt es eine geschlossene kreuzungsfreie Rundreise durch alle $|V|$ Knoten mit Gesamtkosten $\leq k$? Also: Gibt es eine injektive Funktion $f : \{1, \ldots, |V|\} \to V$ mit

  $$\bigcup_{i=1}^{|V|} \{[f(i), f((i+1) \bmod |V|)]\} \subseteq E \text{ und } \sum_{i=1}^{|V|} w([f(i), f((i+1) \bmod |V|)]) \leq k?$$

  HC Eingabe: Gerichteter Graph $G = (V, E)$.

  Frage: Gibt es eine geschlossene kreuzungsfreie Rundreise durch alle $|V|$ Knoten? Also: Gibt es eine injektive Funktion $f : \{1, \ldots, |V|\} \to V$ mit

  $$\bigcup_{i=1}^{|V|} \{[f(i), f((i+1) \bmod |V|)]\} \subseteq E?$$

- **NPH** Menge der NP-harten Probleme: $\text{NPH} = \{P : \forall Q \in \text{NP} : Q \leq_p P\}$.

  NPC Menge der NP-vollständigen Probleme: $\text{NPC} = \text{NP} \cap \text{NPH}$.

- (i) Es gilt TSP $\in$ NP: Ein entsprechender nichtdeterministischer Algorithmus rät eine Rundreise und überprüft anschliessend die Kosten in einer Zeit proportional zu $|V|$.

  (ii) Es gilt HC $\leq_p$ TSP mit folgender Konstruktion: Sei $G = (V, E)$ die Eingabe für HC. Dann sei $G' = (V, V \times V)$ ein vollständiger Graph über den Knoten aus G und die kantenbewertende Funktion $w : V \times V \to \mathbb{N}$ sei definiert als

  $$w([x, y]) = \begin{cases} 1 & \text{falls } [x, y] \in E \text{ und} \\ 2 & \text{falls } [x, y] \notin E. \end{cases}$$

  Wird nun $(G', w, |V|)$ als Eingabe für TSP verwendet, dann lautet die Antwort genau dann "ja", wenn es einen Hamiltonkreis in G gibt.

  Aus (i) und (ii) und mit $HC \in$ NPC folgt TSP $\in$ NPC.