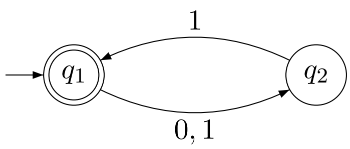
Aufgaben zum Tut am 01.03.2006

Thomas Pajor

28. Februar 2006

Aufgabe 1)

Gegeben sei ein DEA $\mathcal{A}$ über $\Sigma := \{0, 1\}$, der durch folgendes Übergangsdiagramm definiert wird:



Bestimmen Sie durch *systematische Konstruktion* einen regulären Ausdruck R derart, dass gilt $L(R) = L(\mathcal{A})$.

Lösung.

Wir wenden die induktive Konstruktion aus den Vorlesungsfolien auf den Automaten $\mathcal{A}$ an.

Da wir nur einen Endzustand haben, gilt $L(\mathcal{A}) = L^2_{q_1,q_1}$.

Für den Induktionsanfang gilt $L^0_{q_1,q_1} = L^0_{q_1,q_2} = \{\varepsilon\}$, denn es gibt kein Symbol $a \in \Sigma$ so, dass $\delta(q_i, a) = q_i$ für $i = 1, 2$ erfüllt ist. Weiterhin gilt $L^0_{q_1,q_2} = \{0, 1\}$ und $L^0_{q_2,q_1} = \{1\}$.

Als nächstes betrachten wir $L^1_{q_i,q_j}$:

- $L^1_{q_1,q_1} = L^0_{q_1,q_1} \cup (L^0_{q_1,q_1} \cdot (L^0_{q_1,q_1})^* \cdot L^0_{q_1,q_1}) = \{\varepsilon\}$
- $L^1_{q_1,q_2} = L^0_{q_1,q_2} \cup (L^0_{q_1,q_1} \cdot (L^0_{q_1,q_1})^* \cdot L^0_{q_1,q_2}) = \{0, 1\} \cup (\{\varepsilon\} \cdot \{\varepsilon\}^* \cdot \{0, 1\}) = \{0, 1\}$

- $L^1_{q_2,q_1} = L^0_{q_2,q_1} \cup (L^0_{q_2,q_1} \cdot (L^0_{q_1,q_1})^* \cdot L^0_{q_1,q_1}) = \{1\} \cup (\{1\} \cdot \{\varepsilon\}^* \cdot \{\varepsilon\}) = \{1\}$
- $L^1_{q_2,q_2} = L^0_{q_2,q_2} \cup (L^0_{q_2,q_1} \cdot (L^0_{q_1,q_1})^* \cdot L^0_{q_1,q_2}) = \{\varepsilon\} \cup (\{1\} \cdot \{\varepsilon\}^* \cdot \{0,1\}) = \{1\} \cdot \{0,1\}$

Somit folgt insgesamt

$$\begin{aligned} L(\mathcal{A}) &= L^2_{q_1,q_1} \\ &= L^1_{q_1,q_1} \cup (L^1_{q_1,q_2} \cdot (L^1_{q_2,q_2})^* \cdot L^1_{q_2,q_1}) \\ &= \{\varepsilon\} \cup (\{0,1\} \cdot (\{1\} \cdot \{0,1\})^* \cdot \{1\}) \\ &= (\{0,1\} \cdot \{1\})^* \end{aligned}$$

Der gesuchte Ausdruck R ist also

$$R = ((0 \cup 1)(1))^*$$

Aufgabe 2)

Das Problem HITTING SET ist wie folgt definiert:

Gegeben: Eine Menge $\mathcal{U}$ und eine Familie $M := \{M_1, \ldots, M_n\}$ von Teilmengen von $\mathcal{U}$ – oder $M \subseteq 2^{\mathcal{U}}$ mit $|M| = n$, sowie ein Parameter $K \in \mathbb{N}$.

Frage: Gibt es eine Teilmenge $H \subseteq \mathcal{U}$ mit $|H| = K$ derart, dass

$$\forall M_i \in M : |H \cap M_i| \geq 1?$$

Zeigen Sie: HITTING SET ist $\mathcal{NP}$–vollständig.

Lösung.

***Hinweis**: Ich habe die Lösung diesmal von der Ausführlichkeit so gehalten, wie sie in der Klausur gefordert sein könnte.*

- HITTING SET $\in \mathcal{NP}$:

  Zu einer geratenen Lösung H kann wie folgt geprüft werden ob sie ein Hitting Set ist: Für jede Menge M_i prüfe ob es in H ein Element gibt, dass in M_i enthalten ist. Dies kann mit Aufwand $\mathcal{O}(|M| \cdot |\mathcal{U}| \cdot |H|)$ in polynomieller Zeit überprüft werden. Es gilt also HITTING SET $\in \mathcal{NP}$.

- HITTING SET ist $\mathcal{NP}$–hart:

  Wir zeigen: VERTEX COVER $\leq_p$ HITTING SET.

  Sei $I := (V, E, K)$ eine Instanz von VERTEX COVER. Konstruiere dazu wie folgt eine Instanz $I' := (M, \mathcal{U}, K')$ von HITTING SET:

– $\mathcal{U} := V$
– $M := \{\{u, v\} \in E\}$
– $K' := K$

Die Konstuktion ist offensichtlich polynomiell.

Wir zeigen jetzt:

$$I \text{ hat ein Vertex Cover der Größe } K \quad \Leftrightarrow \quad I' \text{ hat ein Hitting Set der Größe } K'$$

In I existiert genau dann ein Vertex Cover der Größe K, wenn es eine Teilmenge $V' \subseteq V$ der Größe K gibt, so dass für jede Kante $\{u, v\} \in E$ gilt: $u \in V' \;\vee\; v \in V'$. Nach Konstruktion hat genau dann auch I' ein Hitting Set der Größe $K' = K$, in dem wir die Menge $H = V'$ wählen. Genau dann gilt für jedes $\{u, v\} \in M$ dass $u \in H$ oder $v \in H$, und somit $|M_i \cap H| \geq 1$ für $i = 1, \ldots, n$[1].

$\Rightarrow$ HITTING SET ist $\mathcal{NP}$–hart.

$\Rightarrow$ HITTING SET ist $\mathcal{NP}$–vollständig. $\square$

Aufgabe 3)

Zeigen Sie: Das Halteproblem ist $\mathcal{NP}$–hart. Warum ist es nicht $\mathcal{NP}$–vollständig?

Lösung.

Wir reduzieren 3SAT auf das Halteproblem.

Sei $\mathcal{M}$ eine Turingmaschine, die als Eingabe eine 3SAT Instanz $I := (C, P := \{P_1, \ldots, P_n\})$ erhält wobei C eine Menge von Klauseln über P ist. $\mathcal{M}$ arbeite wie folgt:

(1) Schreibe einen Bitvektor X der Länge $|P|$ auf das Band und initialisiere ihn mit lauter Nullen. $\mathcal{M}$ behandle den Bitvektor X wie folgt: An der i–ten Stelle von X steht genau dann eine 1, wenn P_i als wahr interpretiert wird.

(2) Prüfe ob X eine erfüllende Variablenbelgung für I ist. Falls ja: stoppe. Sonst überschreibe X durch seinen lexikographischen Nachfolger und prüfe erneut.

(3) Falls $X = (1, \ldots, 1)$ gehe in eine Endlossschleife (Beispielsweise indem der Kopf unendlich lange nach Rechts gefahren wird).

[1] Die Formulierung „genau dann“ in jedem Satz klingt vielleicht etwas doof, aber das spart die Hin– und Rückrichtung getrennt zu behandeln. Diese Vorgehensweise bietet sich hier an, da die Reduktion so simpel ist, dass ihr Beweis in einem Gang erschlagen werden kann.

Betrachte nun eine 3SAT Instanz $I := (C, P)$. Wir konstruieren dazu eine Instanz $I' := (\langle\mathcal{T}\rangle, w)$ des Halteproblems wie folgt:

- $\langle\mathcal{T}\rangle := \langle\mathcal{M}\rangle$
- w ist die Menge der Klauseln sowie die aussagenlogischen Variablen in einer geeigneten Kodierung.

Die Transformation ist offensichtlich polynomiell.

Nach Konstruktion von $\mathcal{M}$ hält die Turingmaschine, wenn die als Eingabe übergebene 3SAT Instanz eine erfüllende Belegung hat, denn es werden in Schritt (2) alle möglichen Variablenbelegungen für P überprüft. Hat I keine erfüllende Belegung so geht $\mathcal{M}$ in Schritt (3) in eine Endlosschleife, und terminiert nicht. Wir konnten also zeigen dass das Halteproblem $\mathcal{NP}$–hart ist. $\square$

Das Halteproblem ist jedoch nicht $\mathcal{NP}$–vollständig, da dafür zusätzlich verlangt wird, dass es in $\mathcal{NP}$ liegt. Dies ist aber nicht gegeben, da das Halteproblem nicht entscheidbar ist. Dies ist also ein Beispiel für ein $\mathcal{NP}$–hartes Problem, das aber *nicht* $\mathcal{NP}$–vollständig ist.

Aufgabe 4)

Gegeben sei das Problem BIN PACKING eingeschränkt auf Bins der Größe 1:

Gegeben: Eine Menge $M := \{a_1, \ldots, a_n\}$ von Elementen, sowie eine Gewichtsfunktion $w : M \to [0, 1)$.

Frage: Wieviele Bins B_j der Größe 1 sind mindestens nötig um die Elemente alle zu verpacken?

Betrachte dazu den folgenden Algorithmus „NEXT–FIT“:

Algorithm 1 NEXT–FIT

$K := 1$
$B_K := \{a_1\}$
for $i = 2, \ldots, n$ **do**
    if $w(a_i) > 1 - \sum_{a \in B_K} w(a)$ **then**
        $K := K + 1$
    $B_K := B_K \cup \{a_i\}$
return K

Zeigen Sie: NEXT–FIT liefert eine Faktor–2–Approximation für BIN PACKING.

Lösung.

Betrachte eine Instanz I von BIN PACKING. NF$(I) = K$ bezeichne dabei die Anzahl Bins die der Algorithmus NEXT–FIT für die Instanz I berechnet.

Wir definieren uns zunächst das Gewicht eines Bins durch die Summe der Elemente die es enthält, also

$$w(B_j) := \sum_{a \in B_j} w(a)$$

Dann gilt folgende Ungleichung für alle $j = 1, \ldots, K-1$:

$$w(B_j) + w(B_{j+1}) \geq 1$$

da, der Algorithmus sonst die Elemente aus B_{j+1} noch in B_j gepackt hätte. Es folgt nun wegen

$$\underbrace{w(B_1) + w(B_2)}_{>1} + \underbrace{w(B_3) + w(B_4)}_{>1} + \ldots + w(B_K)$$

dass

$$\sum_{j=1}^{K} w(B_j) \quad > \quad \frac{K}{2} \qquad \text{falls } k \text{ gerade}$$

$$\sum_{j=1}^{K} w(B_j) \quad > \quad \frac{K-1}{2} \qquad \text{falls } k \text{ ungerade}$$

Eine Optimale Lösung $\mathrm{OPT}(I)$ muss aber zwangsläufig mindestens so viele Bins belegen wie die Elemente aufsummiert an Gewicht tragen, da sonst manche Elemente nicht verpackt worden wären, also

$$\mathrm{OPT}(I) \geq \sum_{j=1}^{K} w(B_j)$$

Damit folgt insgesamt

$$\mathrm{OPT}(I) \geq \sum_{j=1}^{K} w(B_j) > \frac{K-1}{2}$$
$$\Leftrightarrow \mathrm{OPT}(I) > \frac{K-1}{2}$$
$$\Leftrightarrow 2 \cdot \mathrm{OPT}(I) + 1 > K$$
$$\Leftrightarrow 2 \cdot \mathrm{OPT}(I) + 1 > \text{NF}(I)$$

$\square$