Technische Universität Chemnitz Sommersemester 2002
Professur Theoretische Informatik
Prof. Dr. A. Goerdt, Dipl.-Inf. F. Schädlich, Dipl.-Inf. A. Lanka

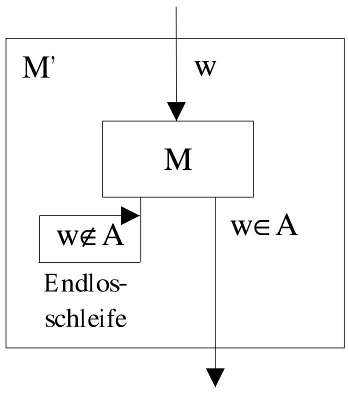
13. Übung „Theoretische Informatik II"
Lösungsvorschläge

Aufgabe 1

PCP ist $\mathcal{NP}$-hart, d. h. es muß $A \leq_p PCP$ für alle Sprachen $A \in NP$ gelten.

Sei $A \in \mathcal{NP}$ beliebig, d. h. A ist entscheidbar und M ist eine nichtdeterministische Turingmaschine, die A akzeptiert. D. h. für ein Wort $x \in A$ stoppt diese Maschine nach polynomieller Zeit in einem Endzustand. Für $x \notin A$ wird kein Endzustand erreicht.

Diese Maschine M wird nun so in eine Maschine M' überführt, so daß M' genau dann stoppt, wenn $x \in A$ ist. Diese Überführung kann leicht konstruiert werden. Zunächst überlegt man sich, wie man eine für eine beliebige Eingabe nicht stoppende Turingmaschine implementieren kann – z. B. durch das Erzeugen einer Endlosschleife. Nun erweitert man die Überführungsfunktion von M so, daß ein nichtdeterministischer Übergang von jedem Nichtendzustand zu der Endlosschleife möglich ist. Diese neue Maschine M' stoppt nun genau dann, wenn $x \in A$ ist.



Die nichtdeterministische Maschine wird nun in ein MPCP (siehe Aufgabe 12.4) übersetzt. Es gilt nun, daß das MPCP genau dann eine Lösung hat, wenn M' stoppt. $\Rightarrow\ A \leq_p MPCP \leq_p PCP \Rightarrow\ A \leq_p PCP$

Das PCP ist unentscheidbar, alle Probleme in $\mathcal{NP}$ sind aber entscheidbar. $\Rightarrow$ PCP $\notin \mathcal{NP}$

Aufgabe 2

Wir wollen $3 - \text{SAT} \leq_p$ unabhängige Menge zeigen. Hierzu ist es erforderlich, daß wir eine Konstruktion finden, die uns in polynomieller Zeit zu einer gegebenen $3 - \text{SAT}$-Formel F einen ungerichteten Graphen $G = (V, E)$ sowie eine Zahl k liefert, so daß F genau dann erfüllbar ist, wenn G eine unabhängige Menge der Größe k hat.

Die Menge der Variablen sei $\{x_1, \ldots, x_n\}$, es gibt also n Variablen bzw. $2n$ Literale. Die Formel F bestehe aus m Klauseln. Die Formel F kann nun wie folgt aufgeschrieben werden:

$$F = \bigwedge_{i=1}^{m} (v_{i,1} \vee v_{i,2} \vee v_{i,3}) \ .$$

Das Literal $v_{i,k}$ entspricht also dem k-ten Literal der i-ten Klausel. Natürlich gilt $v_{i,k} \in \{x_1, \ldots, x_n\} \cup \{\overline{x_1}, \ldots, \overline{x_n}\}$.

Diese umbenannten Literale bilden die Knoten unseres Graphen, es ist also $V = \{v_{1,1}, v_{1,2}, \ldots, v_{i,k}, \ldots, v_{m,3}\}$. Die weitere Konstruktion zielt nun darauf ab, daß der Graph eine unabhängige Menge der Größe $k = m$ hat. Diese soll so aussehen, daß jeweils ein Knoten dieser unabhängigen Menge ebenfalls als wahres Literal in einer Klausel vorkommt. Im weiteren werden wir nun beschreiben, welche Kanten der Graph enthalten muß.

Pro Klausel soll genau ein Knoten in einer gefundenen unabhängigen Menge sein. D. h. die drei zur Klausel i gehörenden Knoten müssen paarweise mit einer Kante verbunden werden: Für jede Klausel i werden die drei Kanten $\{v_{i,1}, v_{i,2}\}$, $\{v_{i,1}, v_{i,3}\}$ und $\{v_{i,2}, v_{i,3}\}$ in den Graph aufgenommen.

Die Knoten der unabhängigen Menge sollen eine erfüllende Belegung von F liefern, indem diese Literale auf 1 (wahr) gesetzt werden. D. h. also, daß zwei Knoten genau dann duch eine Kante verbunden sein müssen, wenn die entsprechen Literale nicht gleichzeitig auf 1 gesetzt werden können: $\{x_{i,k}, x_{j,l}\}$ ist Kante in G, wenn

- $i \neq j$ (verschiedene Klauseln)
- $x_{i,k} = \overline{x_{j,l}}$ (nicht gleichzeitig wahr).

Man überlegt sich nun leicht, daß die Formel F genau dann erfüllbar ist, wenn der Graph G eine unabhängige Menge der Größe k hat:

Ist die Formel F erfüllbar, so existiert eine Belegung der Variablen mit 0 oder 1, die jede Klausel erfüllt. D. h. in jeder Klausel existiert mindestens ein Literal, welches auf 1 gesetzt ist. Nimmt man aus jeder Klausel ein wahres Literal und betrachtet die entsprechenden Knoten in G, so sind diese Knoten auf Grund der Konstruktion paarweise nicht mit einer Kante verbunden und bilden dadurch eine unabhängige Menge der Größe m.

Existiert in G eine unabhängige Menge der Größe m so indizieren je zwei Knoten dieser Menge auf Grund der Konstruktion Literale in verschiedenen Klauseln. Auf Grund der Konstruktion existiert eine Belegung der Variablen, die diese indizierten Literale auf 1 setzt und damit alle Klauseln und somit die Formel F erfüllt.

Aufgabe 3

Die Menge aller Zahlen a_i setzt sich Zusammen aus den Zahlen v_i, v_i', c_i und d_i. Die Konstruktion der Vorlesung ergibt für diese Aufgabe:

$$\begin{aligned}
v_1 &= 10\ 100\\
v_2 &= 11\ 010\\
v_3 &= 00\ 001\\
v_1' &= 01\ 100\\
v_2' &= 00\ 010\\
v_3' &= 11\ 001\\
c_1 &= 10\ 000\\
c_2 &= 01\ 000\\
d_1 &= 20\ 000\\
d_2 &= 02\ 000\\
b &= 44\ 111
\end{aligned}$$

Die Formel F ist hier erfüllbar, da

$$v_1 + v_2' + v_3' + c_1 + c_2 + d_1 + d_2 = b$$

gilt. Die erfüllende Belegung ergibt sich somit aus $x_1 = 1$, $x_2 = 0$ und $x_3 = 0$.

Aufgabe 4

Es sei wieder $\{x_1, \ldots, x_n\}$ die Menge der n Variablen. Die $3-$SAT-Formel F mit ihren m Klauseln läßt sich nun als:

$$F = \bigwedge_{i=1}^{m} (x_{i,1} \vee x_{i,2} \vee x_{i,3})$$

darstellen. Dabei sei $x_{i,k} \in \{x_1, \ldots, x_n\} \cup \{\overline{x_1}, \ldots, \overline{x_n}\}$ wieder das k-te Literal in der i-ten Klausel.

Für eine Klausel in der $3-$SAT-Formel F werden wir drei Klauseln in einer neuen $3-$SAT-Formel F' konstruieren. Diese Formel F' ist genau dann not-all-equal-erfüllbar (NAE-erfüllbar), wenn F erfüllbar ist. Bevor wir damit beginnen überlegt man sich leicht die Gültigkeit folgender Sätze:

Ist eine $3-$SAT-Formel F erfüllbar, so existiert eine Belegung der Variablen, die jede Klausel erfüllt, d. h. in jeder Klausel ist mindestens ein Literal auf 1 gesetzt.

Ist eine $3-$SAT-Formel F nicht erfüllbar, so existiert keine Belegung, die alle Klauseln nicht erfüllt. Würde sie existieren, so würde eine weitere Belegung, in der alle Werte der Variablen negiert sind, alle Klauseln und damit die Formel erfüllen.

Ist eine $3-$SAT-Formel F' NAE-erfüllbar, so generiert jede NAE-erfüllende Belegung der Variablen ein Literal mit dem Wert 1 sowie eines mit dem Wert 0 pro Klausel.

Hat man eine NAE-erfüllende Belegung der Variablen für eine $3-\text{SAT}$-Formel F', so findet man eine weitere Belegung, indem alle Werte der Variablen negiert werden.

Nun zur Konstruktion: Pro Klausel i der Formel F werden zwei neue Variablen y_i und z_i eingeführt. Weiter führt man die Variable a ein. Nun generiert man für eine Klausel $(x_{i,1} \vee x_{i,2} \vee x_{i,3})$ aus F folgende drei Klauseln in F':

$$(x_{i,1} \vee x_{i,2} \vee y_i) \wedge (\overline{y_i} \vee x_{i,3} \vee z_i) \wedge (y_i \vee z_i \vee a) \ .$$

Jede erfüllende Belegung von F erfüllt auch die Klausel Nr. i. Man überzeugt sich leicht davon, daß für eine erfüllende Belegung von $x_{i,1}$, $x_{i,2}$ und $x_{i,3}$ zusätzlich y_i und z_i so gesetzt werden können, so daß die drei neuen Klauseln NAE-erfüllt sind und $a = 1$ gilt.

Ist F' NAE-erfüllbar, so existiert eine erfüllende Belegung mit $a = 1$. Man sieht leicht, daß diese Belegung eines der drei Literale $x_{i,1}$, $x_{i,2}$ oder $x_{i,3}$ für jedes i auf 1 setzen muß und dadurch auch die Klausel Nr. i in F erfüllt wird. Es werden somit alle Klauseln wahr und F ist erfüllbar.

Bemerkung: Setzt eine Belegung für eine nicht erfüllbare $3-\text{SAT}$-Formel F in einer Klausel $x_{i,1} = x_{i,2} = x_{i,3} = 0$ (es muß mindestens eine solche Klausel pro Belegung geben), so wird $a = 0$ in F' erzwungen. Es können aber nicht alle Klauseln von F so belegt werden. Es wird also in F für jede Belegung der Variablen mindestens eine Klausel geben, die $a = 1$ in F' erzwingt. Dadurch wird F' nicht NAE-erfüllbar.

Aufgabe 5

Die Reduktion ist wirklich sehr einfach. Entfernt man die k Knoten der Knotenüberdeckung, so ist der Graph kantenfrei. Bei der Reduktion wird für jede Kante ein neuer Kreis im neuen gerichteten Graphen G' konstruiert: Es ist $G' = (V, E')$ und $E' = \{(u, v) : \{u, v\} \in E\}$. Man überlegt sich leicht, daß diese Reduktion korrekt ist.

Aufgabe 6

Bei dieser Aufgabe ist die Konstruktion einer 3-SAT-Formel F gesucht, die genau dann erfüllbar sein soll, wenn ein gegebener ungerichteter Graph $G = (V, E)$ 3-färbbar ist. Die Farben haben die Nummern 1 bis 3. Die Menge der Knoten sei $V = \{v_1, \ldots, v_n\}$. Für jeden Knoten v_i führen wir in F die drei Variablen $x_{i,1}$, $x_{i,2}$ und $x_{i,3}$ ein. Dabei sollen diese Variablen die Färbung des Knoten v_i kodieren. Setzen wir $x_{i,k} = 1$, so können wir dies so interpretieren, daß der Knoten v_i mit der Farbe k gefärbt werden darf, um eine korrekte Färbung zu erhalten.

In einer korrekten Färbung sind zwei benachbarte Knoten verschieden gefärbt. Dies kann für eine Kante $\{v_i, v_l\}$ durch das Hinzufügen der drei Klauseln $(\overline{x_{i,1}} \vee \overline{x_{l,1}}) \wedge (\overline{x_{i,2}} \vee \overline{x_{l,2}}) \wedge (\overline{x_{i,3}} \vee \overline{x_{l,3}})$ erreicht werden. Jeder Knoten v_i soll aber mit einer Farbe gefärbt werden, es ist also für jeden Knoten v_i die Klausel $(x_{i,1} \vee x_{i,2} \vee x_{i,3})$ hinzuzufügen. Auch hier überlegt man sich wieder leicht, daß F genau dann erfüllbar ist, wenn G 3-färbbar ist.