Universität Karlsruhe (TH)
Lehrstuhl Informatik für Ingenieure und Naturwissenschaftler
Dr. Thomas Worsch
Matthias Schulz

26. Mai 2008

Einführung in die Informatik

Lösungen zu Übungsblatt 5

– Schwierige Probleme, Fibonacci-Zahlen –

Aufgabe 1:

a) Ein möglicher Greedy-Algorithmus, um für einen ungerichteten Graphen $U = (V, E)$ eine kleine Knotenüberdeckung zu finden, sieht wie folgt aus:

Man wählt einen Knoten v maximalen Grades, fügt ihn zur Knotenüberdeckung hinzu, und löscht v sowie alle Kanten, die an v angrenzen, aus U. Dies wiederholt man, bis alle Kanten gelöscht sind; dann grenzt jede Kante an einen Knoten in der Knotenüberdeckung an.

In Pseudocode:

1: $K \leftarrow \emptyset$.
2: **solange** $E \neq \emptyset$
2.1: Wähle Knoten $v \in V$ mit maximalem Grad.
2.2: $V \leftarrow V \setminus \{v\}$
2.3: $E \leftarrow V \times V \cap E$
2.4: $K \leftarrow K \cup \{v\}$

Eine andere Strategie: Man wähle einen Knoten v mit minimalem Grad, füge alle Knoten u mit $\{u, v\} \in E$ zur Knotenüberdeckung hinzu, und lösche alle Kanten, die von diesen Knoten ausgehen, aus U. Dies wiederholt man, bis alle Kanten gelöscht wurden.

Auch in diesem Fall liegt für jede Kante in U mindestens ein Endpunkt in der Knotenüberdeckung.

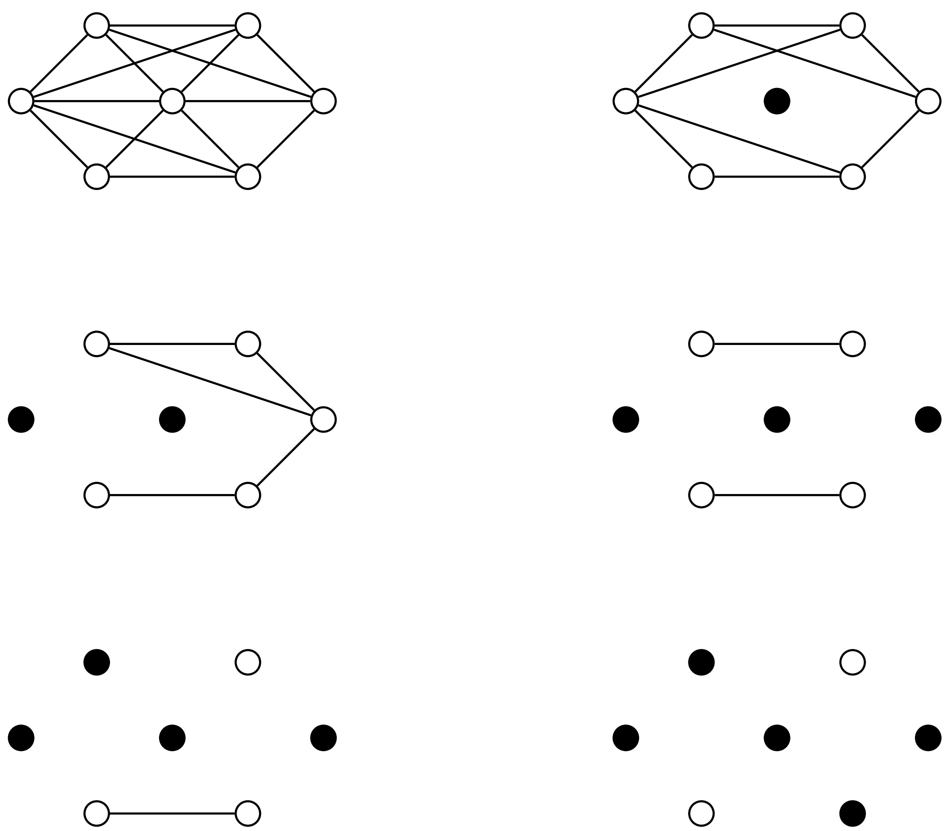
Falls der Minimalgrad der jeweils betrachteten Knoten v immer 1 ist, findet dieser Algorithmus eine minimale Knotenüberdeckung:

Gibt es eine Knotenüberdeckung V_1, die v enthält, und ist u der Knoten, mit dem v verbunden ist, so ist auch $V_1 \setminus \{v\} \cup \{u\}$ ebenfalls eine Knotenüberdeckung. Es gibt also immer eine minimale Knotenüberdeckung, die einen gegebenen Knoten mit Grad 1 nicht enthalten.

In Pseudocode:

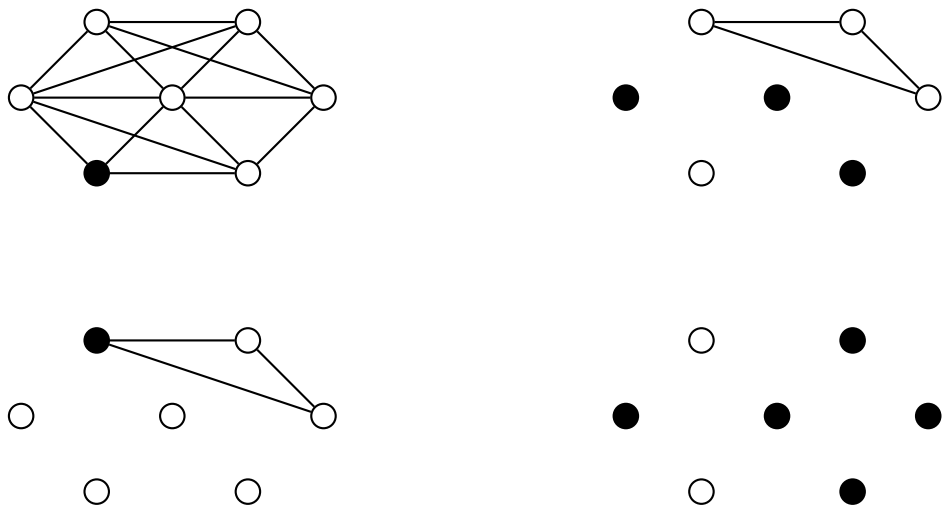
1: $K \leftarrow \emptyset$.
2: **solange** $E \neq \emptyset$
2.1: Wähle Knoten $v \in V$ mit minimalem Grad.
2.2: **for** $u \in V : \{u, v\} \in E$
2.2.1: $V \leftarrow V \setminus \{u\}$
2.2.3: $K \leftarrow K \cup \{u\}$ 2.3: $E \leftarrow V \times V \cap E$

b) Wir führen den Greedy-Algorithmus für den ersten Graphen durch. Die Knoten der Knotenüberdeckung werden dabei schwarz gefärbt.



Für die Wahl der letzten beiden Knoten gibt es vier verschiedene Möglichkeiten, und wir erhalten eine Knotenüberdeckung mit fünf Knoten.

Führt man die zweite Strategie aus, so ergibt sich ebenfalls eine Knotenüberdeckung mit fünf Knoten (abwechselnd wird ein Knoten minimalen Grades und die Knoten in der Überdeckung schwarz eingefärbt):

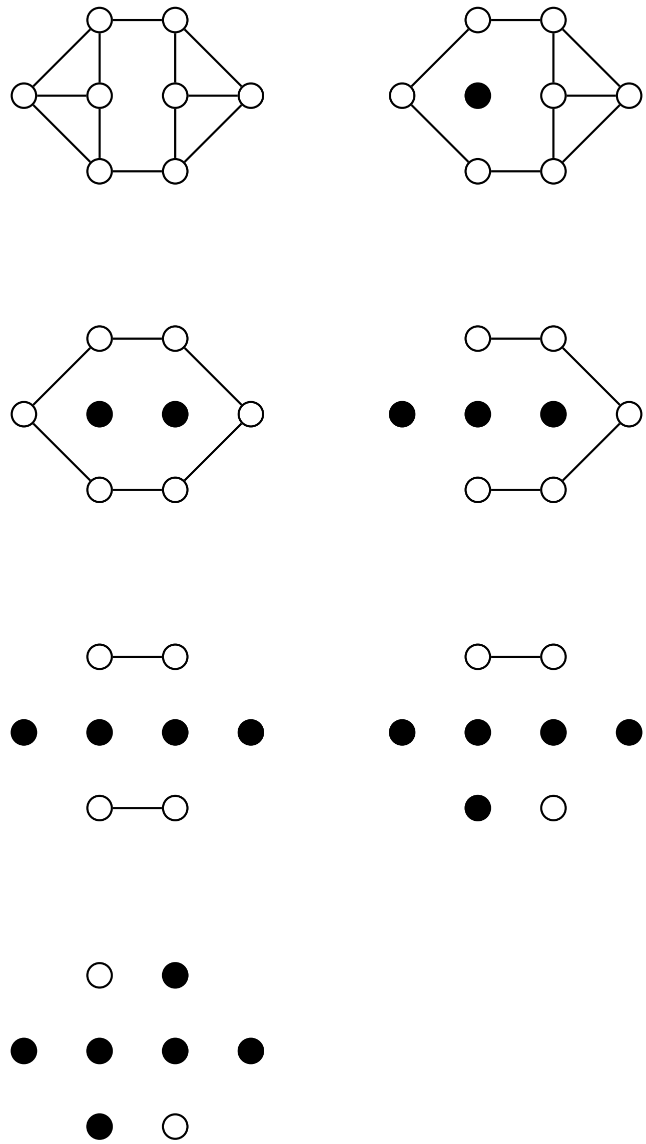


Das Ergebnis ist eine minimale Knotenüberdeckung: Angenommen, es gibt eine Knotenüberdeckung, die mit höchstens vier Knoten auskommt. Dann muss es drei Knoten geben, zwischen denen es keine Verbindung gibt. Vom Knoten in der Mitte gehen Kanten zu allen anderen Knoten aus, von daher kann dieser Knoten nicht zu den drei gesuchten Knoten gehören. Der Knoten ganz links ist nur mit einem Knoten nicht verbunden, kann daher auch nicht zu den drei Knoten gehören. Die einzigen Knoten, mit denen die beiden oberen Knoten nicht verbunden sind, sind die beiden unteren Knoten; wäre einer der oberen Knoten einer der drei gesuchten Knoten, so müssten die beiden anderen Knoten die unteren Knoten sein, die durch eine Kante mit einander verbunden sind, was einen Widerspruch zur Voraussetzung bedeutet.

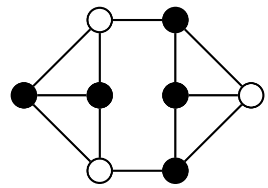
Damit sind vier Knoten ausgeschlossen, und die restlichen drei Knoten müssten die gesuchten Knoten sein. Da zwischen diesen jedoch Kanten verlaufen, gibt es keine drei Knoten, zwischen denen keine Kante verläuft, und somit ist 5 eine untere Grenze für eine Knotenüberdeckung.

Wenn wir den Greedy-Algorithmus oder den zweiten Algorithmus für den zweiten Graphen durchführen wollen, stellen wir fest, dass der erste Knoten mit maximalem/minimalem Grad nicht eindeutig festgelegt ist, da alle Knoten den Grad 3 haben. Es muss also noch festgelegt werden, wie ein Knoten ausgewählt wird, wenn verschiedene Knoten den gleichen Grad haben.

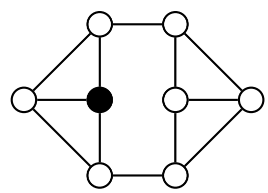
Eine mögliche Berechnung des Greedy-Algorithmus liefert folgende Knotenüberdeckung:

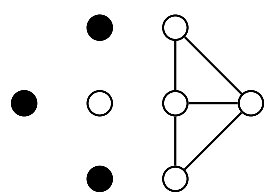


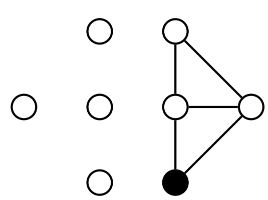
Diese Knotenüberdeckung enthält 6 Knoten. Es gibt jedoch eine Knotenüberdeckung, die nur 5 Knoten enthält:

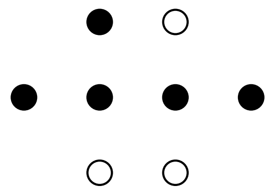


Führt man die zweite Heuristik aus, so gibt es (bis auf Symmetrien) zwei verschiedene Fälle (Markierungen wie oben):

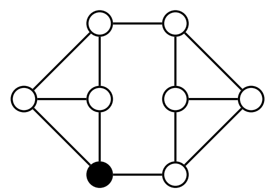


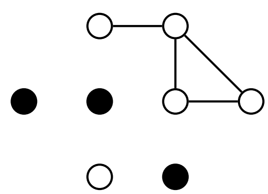


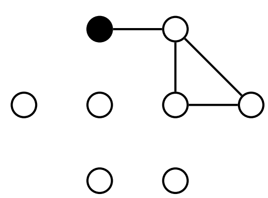


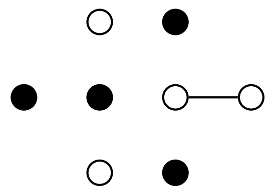


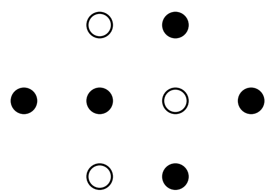
Zweite Möglichkeit:











Der Greedy-Algorithmus hat also in diesem Fall nicht die kleinste Knotenüberdeckung geliefert, der zweite Algorithmus jedoch in beiden Fällen.

c) Wenn eine Heuristik in polynomiell vielen Schritten deterministisch für jeden Graphen die kleinste Knotenüberdeckung bestimmt, lässt sich daraus ein deterministischer Algorithmus konstruieren, dessen Laufzeit polynomiell in n ist und der entscheidet, ob ein gegebener Graph U eine Knotenüberdeckung mit höchstens k Knoten besitzt.

   Dazu wird eine minimale Knotenüberdeckung C von G berechnet (in polynomieller Zeit) und die Größe m von C wird bestimmt. Wenn $k \geq m$ gilt, gibt es eine Knotenüberdeckung mit höchstens k Knoten.

   Da obiges Problem **NP**-vollständig ist, folgt nach Definition, dass jedes andere Problem in **NP** ebenfalls mit einem deterministischen Algorithmus in polynomieller Zeit gelöst werden kann, und es folgt **P**=**NP**.

Aufgabe 2:

a) Nach dem Greedy-Algorithmus wird zuerst die Zeile ersetzt, die am meisten fehlerhafte Elemente enthält.

				X	X
~~X~~	~~X~~	~~X~~	~~X~~	–	–
X					
	X				
		X			
			X		

Da nun alle Ersatzzeilen aufgebraucht sind, werden nach einander die vier Spalten mit den meisten fehlerhaften Elementen ersetzt:

–				X	X
–					
–					
~~X~~	~~X~~	~~X~~	~~X~~	–	–
–					
~~X~~					
–	X				
–		X			
–			X		

–		–		X	X
–		–			
–		–			
~~X~~	~~X~~	~~X~~	~~X~~	–	–
–		–			
~~X~~		–			
–	X	–			
–		~~X~~			
–		–	X		

–		–		~~X~~	X
–		–		–	
–		–		–	
~~X~~	~~X~~	~~X~~	~~X~~	–	–
–		–		–	
~~X~~		–		–	
–	X	–		–	
–		~~X~~		–	
–		–	X	–	

–		–		~~X~~	~~X~~
–		–		–	–
–		–		–	–
~~X~~	~~X~~	~~X~~	~~X~~	–	–
–		–		–	–
~~X~~		–		–	–
–	X	–		–	–
–		~~X~~		–	–
–		–	X	–	–

Es wurden mit Hilfe des Greedy-Algorithmus nicht alle fehlerhaften Elemente ersetzt.

b) Ersetzt man die ersten vier Spalten und die oberste Zeile, sind alle fehlerhaften Elemente ersetzt.

–	–	–	–	~~X~~	~~X~~
–	–	–	–		
–	–	–	–		
~~X~~	~~X~~	~~X~~	~~X~~		
–	–	–	–		
~~X~~	–	–	–		
–	~~X~~	–	–		
–	–	~~X~~	–		
–	–	–	~~X~~		

Aufgabe 3:

a) Ein möglicher Greedy-Algorithmus wäre: Man sucht das Literal l, das in F am häufigsten vorkommt, setzt l auf "wahr" und streicht alle Klauseln, in denen l vorkommt. Dies wiederholt man, bis alle Variablen gesetzt sind oder alle Klauseln gelöscht sind. Wurden alle Klauseln gelöscht, so hat man eine Lösung gefunden.

b) Das Literal a kommt 5 mal vor, das Literal b 4 mal, alle anderen Literale höchstens 3 mal. Somit wird a auf "wahr" gesetzt, und wir erhalten die Formel

$$F_2 = (b \lor d \lor e) \land (b \lor f \lor g) \land (b \lor h \lor i) \land (b \lor j \lor k)$$
$$\land (c \lor d \lor e) \land (c \lor f \lor g) \land (c \lor h \lor i) \land (\overline{a} \lor \overline{b} \lor \overline{c}).$$

Hier kommt das Literal b 4 mal vor, das Literal c 3 mal und jedes andere Literal höchstens 2 mal. Das Literal b wird somit auf "wahr" gesetzt, und wir erhalten

$$F_3 = (c \lor d \lor e) \land (c \lor f \lor g) \land (c \lor h \lor i) \land (\overline{a} \lor \overline{b} \lor \overline{c}).$$

Das Literal c kommt 3 mal vor, alle anderen Literale seltener, und somit setzen wir c auf "wahr":

$$F_4 = (\overline{a} \lor \overline{b} \lor \overline{c}).$$

Da jedoch $a, b, c =$"wahr" gilt, wird diese Klausel nicht erfüllt.

Es gibt allerdings Lösungen für die Formel F_1: Setzt man alle Variablen von d bis m auf "wahr" und die Variablen a, b, c auf "falsch", so nimmt F_1 den Wert "wahr" an: Bis auf die letzte Klausel enthalten alle Klauseln mindestens eines der Literale d bis m, die alle "wahr" sind, und sind somit erfüllt; die letzte Klausel enthält die Negationen von a, b, c, die alle "wahr" sind, da a, b, c auf "falsch" gesetzt wurden.

Hinweis: Man kann den Greedy-Algorithmus noch verfeinern, indem man das Literal $\overline{l}$ aus der Formel streicht, wenn man das Literal l auf "wahr" setzt, und Literale, die alleine in einer Klausel stehen, auf "wahr" setzt (anstelle des am häufigsten vertretenen Literals).

Aufgabe 4:

a) Es gilt: $2F_{n+2} = F_{n+2} + F_{n+2} = (F_n + F_{n+1}) + F_{n+2} = F_n + (F_{n+1} + F_{n+2}) = F_n + F_{n+3}$.

b) Es gilt $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} F_2 \\ F_1 \end{pmatrix}$.

Damit gilt die Behauptung für $n = 1$.

$n \to n+1$: $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n+1} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
$= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} =$
$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix} = \begin{pmatrix} F_{n+1} + F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix}$.

Damit ist die Behauptung durch vollständige Induktion bewiesen.

c) Beweis durch vollständige Induktion:

$n = 1 : \frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2}) - (\frac{1-\sqrt{5}}{2})) = \frac{1}{\sqrt{5}} \cdot \sqrt{5} = 1$
$n = 2 : \frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^2 - (\frac{1-\sqrt{5}}{2})^2) = \frac{1}{\sqrt{5}}((\frac{1+2\sqrt{5}+5}{4}) - (\frac{1-2\sqrt{5}+5}{4})) = \frac{1}{\sqrt{5}}\frac{4\sqrt{5}}{4} = 1.$

$n+1 \to n+2$: Die Gleichung $x^2 - x - 1 = 0$ hat die beiden Lösungen $x_1 = \frac{1+\sqrt{5}}{2}$ und $x_2 = \frac{1-\sqrt{5}}{2}$. Somit gilt für $i = 1, 2 : x_i^2 = x_i + 1 \Rightarrow x_i^{n+2} = x_i^{n+1} + x_i^n$.

Es folgt: $\frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^{n+2} - (\frac{1-\sqrt{5}}{2})^{n+2}) = \frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^{n+1} + (\frac{1+\sqrt{5}}{2})^{n} - (\frac{1-\sqrt{5}}{2})^{n+1} - (\frac{1-\sqrt{5}}{2})^{n}) = \frac{1}{\sqrt{5}}(((\frac{1+\sqrt{5}}{2})^{n+1} - (\frac{1-\sqrt{5}}{2})^{n+1}) + ((\frac{1+\sqrt{5}}{2})^{n} - (\frac{1-\sqrt{5}}{2})^{n})) = F_{n+1} + F_n = F_{n+2}$.

d) Für $n = 1$ gilt $gcd(F_n, F_{n+1}) = gcd(1, 1) = 1$.
$n \to n+1$: Sei g der größte gemeinsame Teiler von F_{n+1} und F_{n+2}. Dann teilt g auch $F_{n+2} - F_{n+1} = F_n$, da g sowohl F_{n+2} als auch F_{n+1} teilt. Da der größte gemeinsame Teiler von F_n und F_{n+1} nach Induktionsvoraussetzung 1 ist und g ein gemeinsamer Teiler von F_n und F_{n+1} ist, folgt $gcd(F_{n+1}, F_{n+2}) = 1$.

e) Wir stellen wieder eine Gleichung wie im Beweis zu c) auf:

$x^2 - x - 2 = 0$ (folgt aus $G_{n+2} - G_{n+1} - 2G_n = 0$).

Für die Lösungen x_1 und x_2 dieser Gleichung gilt wieder $x_i^{n+2} = x_i^{n+1} + 2x_i^n$, und somit erfüllt die Folge der Potenzen von x_1 und x_2 die Rekursionsgleichung.

Ebenso erfüllen alle Linearkombinationen von x_1 und x_2 die Rekursionsgleichung; es gilt also, die Zahlen a und b zu finden, für die gilt:

$ax_1 + bx_2 = G_1$
$ax_1^2 + bx_2^2 = G_2$.

Löst man die obige Gleichung auf, so erhält man

$x_{1,2} = \frac{1 \pm \sqrt{1+4\cdot 2}}{2} = \frac{1 \pm 3}{2}$ und somit $x_1 = 2, x_2 = -1$.

Das Gleichungssystem ergibt somit

$2a - b = 1$
$4a + b = 1$.

Es ergibt sich durch Addition der beiden Gleichungen $6a = 2 \Rightarrow a = \frac{1}{3}$ und somit $b = \frac{2}{3} - 1 = -\frac{1}{3}$.

Es folgt: Für die Zahlen G_n mit $G_n = \frac{1}{3}(2^n - (-1)^n)$ gilt $G_1 = G_2 = 1, G_{n+2} = G_{n+1} + 2G_n$, und die gesuchte Formel ist gefunden.