Technische Universität München
Fakultät für Informatik
Lehrstuhl für Informatik 15
Computergraphik & Visualisierung
Prof. Dr. Rüdiger Westermann
Dr. Werner Meixner

Wintersemester 2007/08
Lösungsblatt 10
30. Januar 2008

Diskrete Strukturen

Hausaufgabe 1 (5 Punkte)

Beweisen oder widerlegen Sie:

1. In jedem planaren Graphen gibt es einen Knoten der höchstens Grad 5 hat.
2. Es gibt einen 5-regulären planaren Graphen.

Lösungsvorschlag

1. Sei $G = (V, E)$ ein planarer Graph. Dann haben wir zu zeigen, dass ein Knoten x mit $deg(x) \leq 5$ existiert.

   Wir führen einen Widerspruchsbeweis, indem wir annehmen, dass alle Knoten aus V mindestens den Grad 6 besitzen, d. h. $deg(x) \geq 6$ gilt, und diese Annahme zum Widerspruch führen.

   Sei also $deg(x) \geq 6$ für alle $x \in V$.

   Es folgt zunächst $|V| \geq 3$.

   Die Summe aller Gradzahlen von Knoten in G ist gleich dem Doppelten der Anzahl der Kanten, d. h. es gilt

   $$\sum_{x \in V} deg(x) = 2 \cdot |E| \,.$$

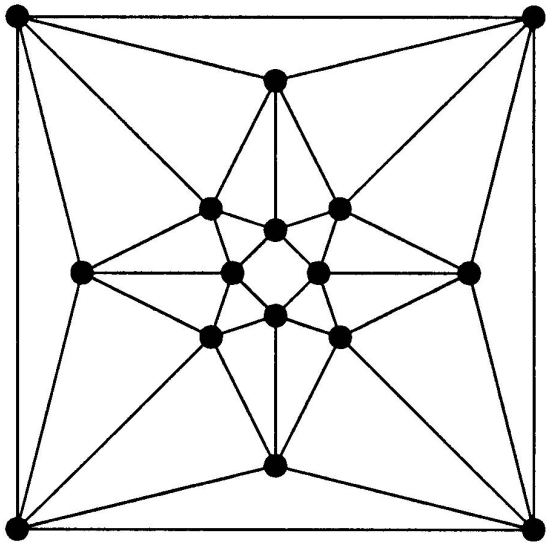
   Mithin gilt

   $$2 \cdot |E| = \sum_{x \in V} deg(x) \geq |V| \cdot 6 \,, \quad \text{d. h.} \quad |E| \geq 3|V| \,.$$

   Andererseits gilt für planare Graphen mit mindestens 3 Knoten nach einem Satz der Vorlesung

   $$|E| \leq 3 \cdot |V| - 6 < 3 \cdot |V| \,.$$

   Widerspruch!



2.

Hausaufgabe 2 (5 Punkte)

Beweisen oder widerlegen Sie:

1. Jeder 3-reguläre Graph mit Hamiltonkreis hat chromatischen Index 3.
2. Die chromatische Zahl eines k-regulären Graphen (V, E) ist mindestens $\left\lceil \frac{|V|}{|V|-k} \right\rceil$.

Lösungsvorschlag

1. Wegen
$$\sum_{x \in V} deg(x) = 3|V| = 2 \cdot |E|$$
ist die Anzahl $|V|$ der Knoten gerade. Die Kanten des existierenden Hamiltonkreises können also mit genau 2 Farben gefärbt werden. Da die verbleibenden Kanten ein nichtleeres Matching bilden, können also die restlichen Kanten mit genau einer dritten Farbe gefärbt werden.

2. Eine Knotenfärbung mit n Farben bestimmt insbesondere die Anzahl f_i der Knoten mit Farbe i. Es gilt dann
$$|V| = \sum_{i=1}^{n} f_i \,.$$
Da jeder Knoten k Nachbarn besitzt, können mit einer Farbe i höchstens $|V| - k$ Knoten gefärbt werden, d. h. $f_i \leq |V| - k$. Damit folgt
$$|V| = \sum_{i=1}^{n} f_i \leq n \cdot (|V| - k) \,.$$
Daraus folgt für $n = \chi(G)$ die behauptete Ungleichung
$$\frac{|V|}{|V| - k} \leq \chi(G) \,.$$

Hausaufgabe 3 (5 Punkte)

Wir betrachten [4] als linear geordnete Knotenmenge eines Suchbaumes B.

1. Listen Sie alle möglichen Suchbäume mit [4] als Knotenmenge auf.
2. Wie viele Knoten muß der kleinste vollständige Suchbaum enthalten, der alle Knoten von B enthält?

Lösungsvorschlag

1. Falls 1 in die Baumwurzel eingetragen wird, dann gibt es 5 mögliche Suchbäume.
   Falls 4 in die Baumwurzel eingetragen wird, dann gibt es analog zum ersten Fall ebenfalls 5 mögliche Suchbäume.
   Falls 2 in die Baumwurzel eingetragen wird, dann gibt es 2 mögliche Suchbäume.
   Falls 3 in die Baumwurzel eingetragen wird, dann gibt es ebenfalls 2 mögliche Suchbäume.

2. 7.

Hausaufgabe 4 (5 Punkte)

Konstruieren Sie jeweils einen Algorithmus, der bestimmt, ob ein beliebig gegebener Graph $G = (V, E)$ ein Baum ist, basierend auf

1. Tiefensuche,
2. Breitensuche.

Lösungsvorschlag

1. Wir modifizieren den Algorithmus der Tiefensuche aus der Vorlesung so, dass wir erkennen, ob wir Knoten zum wiederholten Mal besuchen und testen, ob alle Knoten besucht werden. Aus dem ersten Test können wir auf die Kreisfreiheit schließen, mit dem zweiten auf Zusammenhang.

   Eingabe: Graph $G = (V, E)$, Startknoten $s \in V$
   Ausgabe: Boolean $isTree$
   Feld $pred[v]$ mit $v \in V$

   for all $v \in V$ **do** $pred[v] = nil$
   S $\leftarrow$ new STACK;
   $v \leftarrow s$;
   $isTree \leftarrow true$;
   repeat
   if $\exists u \in \Gamma(v) \backslash \{pred[v]\}$ mit $pred[u] \neq nil$ und $pred[u] \neq v$ **then**
   $isTree \leftarrow false$; // Circle detected!
   if $\exists u \in \Gamma(v) \setminus \{s\}$ mit $pred[u] = nil$ **then**

```
            S.PUSH(v);
            pred[u] ← v;
            v ← u;
        else if not S.ISEMPTY() then
            v ← S.POP();
        else
            v = nil;
    until v = nil;
    for all v ∈ V \ {s} do
        if pred[v] = nil then
            isTree ← false;    // not connected!
```

2. Wir modifizieren den Algorithmus der Breitensuche aus der Vorlesung so, dass wir erkennen, ob wir Knoten zum wiederholten Mal besuchen und testen, ob alle Knoten besucht werden. Aus dem ersten Test können wir auf die Kreisfreiheit schließen, mit dem zweiten auf Zusammenhang.

```
Eingabe: Graph G = (V, E), Startknoten s ∈ V
Ausgabe: Boolean isTree
Felder d[v], pred[v] mit v ∈ V

for all v ∈ V do begin
    if v = s then d[v] ← 0 else d[v] ← ∞;
    pred[v] ← nil;
end
Q ← new QUEUE;
Q.INSERT(s);
isTree ← true;
while not Q.ISEMPTY() do begin
    v ← Q.DEQUEUE();
    for all u ∈ Γ(v)\{pred[v]} do
        if pred[u] ≠ nil then
            isTree ← false;    // Circle detected!
        if d[u] = ∞ then begin
            d[u] ← d[v] + 1;
            pred[u] ← v;
            Q.INSERT(u);
        end
end
for all v ∈ V \ {s} do
    if pred[v] = nil then
        isTree ← false;    // not connected!
```

Hausaufgabe 5 (0 Punkte)

Wahr oder falsch?

– – Es gibt keinen Baum ohne Blätter! – –

Lösungsvorschlag

Falsch!

Hinweis: *Die im Folgenden als Vorbereitung bezeichneten Aufgaben werden nicht bewertet und dienen der häuslichen Vorbereitung der Tutoraufgaben, die ebenfalls nicht bewertet werden. Die Abgabe einer Bearbeitung der Vorbereitungsaufgaben zusammen mit der Bearbeitung der Hausaufgaben wird empfohlen. Tutoraufgaben werden in den Übungsgruppen bearbeitet.*

Vorbereitung 1

Die Komposition $f \circ g$ von Abbildungen f und g ist durch $(f \circ g)(x) = f(g(x))$ definiert. Dies gilt insbesondere für Permutationen einer Menge $[n]$. Die Menge der Permutationen von $[n]$ zusammen mit der Komposition $\circ$ bildet eine Algebra, die wir bekanntlich die Symmetrische Gruppe S_n nennen.

Wir betrachten die folgenden Permutationen q, r, s aus S_5

$$q = (15)\,, \qquad r = (321)\,, \qquad s = (3452)\,.$$

q, r, s sind hier in der Zyklenschreibweise angegeben, wobei Zyklen der Länge 1 weggelassen wurden. Beispielweise gilt $q(1) = 5$ und $q(3) = 3$.
Sei

$$p = q \circ r \circ s\,.$$

1. Geben Sie p als Liste von Paaren $(x, p(x))$ an.
2. Geben Sie p als Komposition von disjunkten zyklischen Permutationen an.

   Hinweis: Permutationen $s \in S_n$ und $t \in S_n$ heißen disjunkt, falls für alle $x \in [n]$ gilt $s(x) = x$ oder $t(x) = x$.

Lösungsvorschlag

1. $p = \{(1,3), (2,2), (3,4), (4,1), (5,5)\}$.
2. $p = (1\ 3\ 4)(2)(5)$.

Vorbereitung 2

Berechnen Sie

$$\text{(i)} \quad x = 17 \bmod 3\,, \qquad \text{(ii)} \quad y = 3^{20} \bmod 4\,, \qquad \text{(iii)} \quad z = (-30) \bmod 7\,.$$

Lösungsvorschlag

(i) $x = 2$.

(ii) $y = 3^{20} \bmod 4 = 9^{10} \bmod 4 = (9 \bmod 4)^{10} \bmod 4 = 1^{10} \bmod 4 = 1$.

(iii) $z = (-30) \bmod 7 = (-30 + 35) \bmod 7 = 5 \bmod 7 = 5$.

Man beachte, dass für alle $k \in \mathbb{Z}$ stets $x \bmod n = (x + k \cdot n) \bmod n$ gilt.

Tutoraufgabe 1

Gegeben ist die folgende Entfernungstabelle. Ein Strich (" -") bedeutet, dass keine direkte Verbindung zwischen den Städten angenommen wird. Andernfalls ist die Entfernung in Kilometern angegeben.

	Dortmund	Frankfurt	Karlsruhe	Kassel	Köln	Nürnberg	Mannheim	München	Stuttgart	Ulm
Dortmund	-	-	-	165	83	-	-	-	-	-
Frankfurt	-	-	-	-	189	228	88	-	-	294
Karlsruhe	-	-	-	-	-	-	68	-	82	-
Kassel	165	-	-	-	-	-	-	-	-	465
Köln	83	189	-	-	-	-	-	-	-	-
Nürnberg	-	228	-	-	-	-	-	167	-	-
Mannheim	-	88	68	-	-	-	-	-	-	-
München	-	-	-	-	-	167	-	-	-	139
Stuttgart	-	-	82	-	-	-	-	-	-	92
Ulm	-	294	-	465	-	-	-	139	92	-

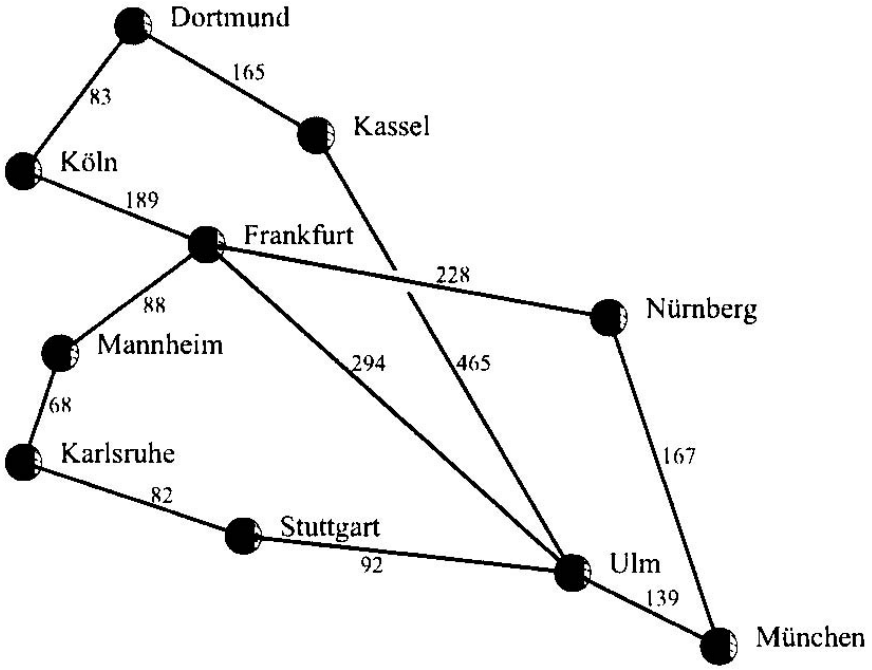
1. Zeichnen Sie den zu der Entfernungstabelle gehörigen (ungerichteten) gewichteten Graphen.

2. Bestimmen Sie nach dem Algorithmus von Dijkstra die Entfernung von München nach Köln.

3. Wie müssen Sie den Algorithmus von Dijkstra modifizieren, damit Sie den kürzesten Weg von München nach Köln als Resultat erhalten?

Lösungsvorschlag

1.



2.,3. Der Dijkstra-Algorithmus liefert zunächst nur die Entfernung. Um den Pfad zu bekommen, muss für jeden Knoten ein Zeiger auf einen Vorgängerknoten gehalten werden, genau wie in der Breitensuche. Dieser Zeiger erhält dann ein Update, wenn auch $d[v]$ geändert wird. Dann kann man den durch diese Zeiger gegebenen Pfad rückwärts traversieren, um die Orte "einzusammeln". Wir geben also zusätzlich zu den Distanzen d den Zeiger p an.

1) $F = \{$Ulm,Stuttgart,Karlsruhe,Mannheim,
Frankfurt,Nürnberg,Kassel,Dortmund,Köln$\}$.
d[Ulm] = 139, d[Stuttgart] = ∞, d[Karlsruhe] = ∞, d[Mannheim] = ∞, d[Frankfurt] = ∞, d[Nürnberg] = 167, d[Kassel] = ∞, d[Dortmund] = ∞, d[Köln] = ∞, d[München] = 0.
p[Ulm] = München, p[Nürnberg] = München.

2) u = Ulm,
$F = \{$Stuttgart,Karlsruhe,Mannheim,
Frankfurt,Nürnberg,Kassel,Dortmund,Köln$\}$.
Updates: d[Kassel] = 604, d[Stuttgart] = 231, d[Frankfurt] = 433.
p[Kassel] = Ulm, d[Stuttgart] = Ulm, d[Frankfurt] = Ulm.

3) u = Nürnberg,
$F = \{$Stuttgart,Karlsruhe,Mannheim,Frankfurt,Kassel,Dortmund,Köln$\}$.
Updates: d[Frankfurt] = 395, p[Frankfurt] = Nürnberg.

4) $u =$ Stuttgart, $F =$ {Karlsruhe,Mannheim,Frankfurt,Kassel,Dortmund,Köln}.
Updates: d[Karlsruhe] = 313, p[Karlsruhe] = Stuttgart.

5) $u =$ Karlsruhe, $F =$ {Mannheim,Frankfurt,Kassel,Dortmund,Köln}.
Updates: d[Mannheim] = 381, p[Mannheim] = Stuttgart.

6) $u =$ Mannheim, $F =$ {Frankfurt,Kassel,Dortmund,Köln}.
Updates: keine neuen kürzeren Wege.

7) $u =$ Frankfurt, $F =$ {Kassel,Dortmund,Köln}.
Updates: d[Köln] = 584, p[Köln] = Frankfurt.

8) $u =$ Köln, $F =$ {Kassel,Dortmund}.
Updates: d[Dortmund] = 667, p[Dortmund] = Köln.

9) $u =$ Kassel, $F =$ {Dortmund}.
Updates: keine kürzeren Wege.

10) $u =$ Dortmund, $F = \oslash$.
Updates: keine kürzeren Wege.

Rücktraversierung von p[Köln] liefert (Köln, Frankfurt, Nürnberg, München) mit 584km.

Tutoraufgabe 2

Wir betrachten Algebren $A = \langle S, \circ \rangle$ mit einer 4-elementigen Trägermenge S und einer Operation $\circ$, die die folgende (2-seitige) Kürzungsregel erfüllt für alle $x, x', y \in S$

$$x \circ y = x' \circ y \;\Rightarrow\; x = x' \quad \wedge \quad y \circ x = y \circ x' \;\Rightarrow\; x = x' \,.$$

Wir fordern außerdem, dass alle "Quadrate" von Elementen aus A (d. h. aus S) rechtsneutral (rechtes Einselement) sind, d. h., dass für alle $x, y \in S$ gilt

$$y \circ (x \circ x) = y \,.$$

1. Zeigen Sie die Existenz eines eindeutigen rechten Einselements in A, i. Z. $1_r \in A$.

2. Wir nehmen an, dass 1_r auch linkes Einselement ist, und können in diesem Fall 1 schreiben für 1_r. Geben Sie für diesen Fall eine Verknüpfungstafel für $\circ$ an!

   Machen Sie sich zunächst klar, was die (2-seitige) Kürzungsregel für die Elemente der Spalten bzw. Zeilen der Verknüpfungstafel bedeutet.

3. Wir nehmen nun an, dass 1_r nicht auch linksneutral (linkes Einselement) ist.

   (a) Geben Sie für diesen Fall eine Verknüpfungstafel für $\circ$ an!

   (b) Zeigen Sie die Eindeutigkeit der Verknüpfungstafel bis auf Isomorphie!

   (c) Zeigen Sie, dass die Verknüpfung $\circ$ nicht assoziativ und die Algebra A damit keine Halbgruppe ist!

4. Besitzt A in jedem Fall eine echte Unteralgebra? Begründung!

Lösungsvorschlag

1. Sei $x \in S$. Dann ist x^2 wegen $y \circ x^2 = y$ rechtsneutral (rechtes Einselement). Zum Beweis seiner Eindeutigkeit sei $z \in A$ ein beliebiges rechtes Einselement bezgl. $\circ$. Dann gilt einerseits $z \circ z = z$ und andererseits $z \circ x^2 = z$. Mit der rechten Kürzungsregel folgt dann $z = x^2$.

2. Die rechte Kürzungsregel bedeutet, dass alle Elemente einer Spalte einer Verknüpfungstafel voneinander verschieden sind. Entprechend bedeutet die linke Kürzungsregel, dass alle Elemente einer Zeile einer Verknüpfungstafel voneinaner verschieden sind.

   Wir tragen jene Elemente sofort in die Tafel ein, die sich aus den Gleichungen mit dem Einselement ergeben. Die übrigen Positionen sind zunächst noch unbestimmt.

$\circ$	1	b	c	d
1	1	b	c	d
b	b	1		
c	c		1	
d	d			1

   Die Vervollständigung geschieht in eindeutiger Weise nach den Kürzungsregeln.

$\circ$	1	b	c	d
1	1	b	c	d
b	b	1	d	c
c	c	d	1	b
d	d	c	b	1

3. Sei 1_r nicht auch linkes Einselement.

   (a) Wir tragen jene Elemente sofort in die Tafel ein, die sich aus den Gleichungen mit dem rechten Einselement 1_r ergeben. Die übrigen Positionen sind zunächst noch unbestimmt.

$\circ$	1_r	b	c	d
1_r	1_r			
b	b	1_r		
c	c		1_r	
d	d			1_r

   Wenn 1_r nicht auch linkes Einselement ist, dann muss die Gleichung $1_r \circ x = x$ für irgendein x verletzt sein. Wir betrachten zunächst den Fall, dass diese Gleichung weder für $x = b$, noch für $x = c$ und auch nicht für $x = d$ gilt und zeigen sofort einen Widerspruch. Die folgenden Tafeln zeigen die Fälle $1_r \circ b = c$ bzw. $1_r \circ b = d$. Die Stelle, an der ein Widerspruch entsteht, ist mit einem Fragezeichen gekennzeichnet.

$\circ$	1_r	b	c	d
1_r	1_r	c	d	b
b	b	1_r		
c	c	d	1_r	?
d	d	b		1_r

$\circ$	1_r	b	c	d
1_r	1_r	d	b	c
b	b	1_r	c	d
c	c		1_r	
d	d		?	1_r

Da die Gleichung $1_r \circ x = x$ von mindestens 2, und damit genau 2 Elementen verletzt sein muss, nehmen wir an, dass b dasjenige Element ist, für das die Gleichung gilt. Wir erhalten zwingend

$\circ$	1_r	b	c	d
1_r	1_r	b	d	c
b	b	1_r	c	d
c	c	d	1_r	b
d	d	c	b	1_r

(b) Wir nehmen eine Algebra $A' = \langle S', \circ' \rangle$ an mit Elementen $S' = \{1'_r, b', c', d'\}$, wobei o. B. d. A. $1'_r \circ b' = b'$, $1'_r \circ c' = d'$ und $1'_r \circ d' = c'$ gelte. Dann ergibt sich zwingend

$\circ$	$1'_r$	b'	c'	d'
$1'_r$	$1'_r$	b'	d'	c'
b'	b'	$1'_r$	c'	d'
c'	c'	d'	$1'_r$	b'
d'	d'	c'	b'	$1'_r$

Offenbar ist die Abbildung $h : S \to S'$ mit

$$h(1_r) = h(1'_r), \quad h(b) = b', \quad h(c) = c', \quad h(d) = d'$$

eine Isomorphie.

(c) Wäre die Verknüpfung assoziativ, dann würde gelten

$$1_r \circ x = (x \circ x) \circ x = x \circ (x \circ x) = x \circ 1_r = x$$

im Widerspruch zu $1_r \circ c = d$.

4. Eine echte Unteralgebra ist jedenfalls eine Unteralgebra, deren Trägermenge S' eine echte Teilmenge von S ist, d.h. u.a. ungleich $S' \neq S$. Bei Algebren schließt man übrigens die leere Menge als Trägermenge generell aus, insbesondere also auch bei Unteralgebren.

   Im ersten Fall (Teilaufgabe 2) sieht man sofort, dass es echte Teilmengen von S gibt, aus denen die Anwendung der Operation $\circ$ nicht herausführt. Solche Teilmengen sind $\{1, b\}$, $\{1, c\}$ und $\{1, d\}$, ja sogar aus $\{1\}$ führt die Operation $\circ$ nicht heraus.

   Dies bedeutet, dass es im ersten Fall echte Unteralgebren gibt.

   Im zweiten Fall (Teilaufgabe 3) bildet offenbar $U = \langle \{1_r\}, \circ \rangle$ eine Unteralgebra. Allerdings ist beispielsweise die Menge $\{1_r, c\}$ nicht abgeschlossen unter der Operation $\circ$ und kann deshalb nicht Trägermenge einer Unteralgebra sein.

   Wir folgern, dass es auch im zweiten Fall echte Unteralgebren gibt.