IWI Fakultät für Informatik und Wirtschaftsinformatik

Lehrgebiet für Grundlagen der Informatik
Prof. Dr. Heiko Körner

Hochschule Karlsruhe Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES
Näher dran.

6. Übung zur Vorlesung Algorithmen auf Graphen

Musterlösungen

Aufgabe 1: Anbei die Lösungen zu allen drei Teilaufgaben:

a) Die starken Zusammenhangskomponenten sind $\{a, e\}$, $\{b, c\}$, $\{d, g, h\}$ und $\{f\}$.

b) Die *Reach*–Mengen lauten

$$Reach[a] = \{a, d, e, f, g, h\} \ , \quad Reach[b] = V \ ,$$
$$Reach[c] := V \ , \quad Reach[d] := \{d, g, h\} \ , \quad Reach[e] := \{a, d, e, f, g, h\} \ ,$$
$$Reach[f] := \{d, f, g, h\} \ , \quad Reach[g] := \{d, g, h\} \ , \quad Reach[h] := \{d, g, h\} \ .$$

c) Wir geben die Knotenmenge V' an, die beim Erreichen des Kopfes der `while`–Schleife (in Zeile 2) jeweils noch vorhanden ist. Ein möglicher Programmablauf sieht dann wie folgt aus:

—

$\Downarrow$ Die erste Zeile setzt V' auf ganz V

$\boxed{V' = \{a, b, c, d, e, f, g, h\}}$

$\Downarrow$ Wähle $d \in V'$ in Zeile 3; in Zeile 10 wird $\{d, g, h\}$ ausgegeben

$\boxed{V' = \{a, b, c, e, f\}}$

$\Downarrow$ Wähle $b \in V'$ in Zeile 3; in Zeile 10 wird $\{b, c\}$ ausgegeben

$\boxed{V' = \{a, e, f\}}$

$\Downarrow$ Wähle $a \in V'$ in Zeile 3; in Zeile 10 wird $\{a, e\}$ ausgegeben

$\boxed{V' = \{f\}}$

$\Downarrow$ Wähle $f \in V'$ in Zeile 3; in Zeile 10 wird $\{f\}$ ausgegeben

$\boxed{V' = \emptyset}$

Möglicherweise wird aber auch zuerst der Knoten c ausgewählt. Ein möglicher Programmablauf wäre dann:

—

$\Downarrow$ Die erste Zeile setzt V' auf ganz V

$\boxed{V' = \{a, b, c, d, e, f, g, h\}}$

$\Downarrow$ Wähle $c \in V'$ in Zeile 3; in Zeile 10 wird $\{b, c\}$ ausgegeben

$\boxed{V' = \{a, d, e, f, g, h\}}$

$\Downarrow$ Wähle $f \in V'$ in Zeile 3; in Zeile 10 wird $\{f\}$ ausgegeben

$\boxed{V' = \{a, d, e, g, h\}}$

$\Downarrow$ Wähle $h \in V'$ in Zeile 3; in Zeile 10 wird $\{d, g, h\}$ ausgegeben

$\boxed{V' = \{a, e\}}$

$\Downarrow$ Wähle $a \in V'$ in Zeile 3; in Zeile 10 wird $\{a, e\}$ ausgegeben

$\boxed{V' = \emptyset}$

Falls dagegen der Algorithmus mit den Knoten e beginnt, so wäre dieser Ablauf möglich:

$\Downarrow$ Die erste Zeile setzt V' auf ganz V

$\boxed{V' = \{a, b, c, d, e, f, g, h\}}$

$\Downarrow$ Wähle $e \in V'$ in Zeile 3; in Zeile 10 wird $\{a, e\}$ ausgegeben

$\boxed{V' = \{b, c, d, f, g, h\}}$

$\Downarrow$ Wähle $b \in V'$ in Zeile 3; in Zeile 10 wird $\{b, c\}$ ausgegeben

$\boxed{V' = \{d, f, g, h\}}$

$\Downarrow$ Wähle $f \in V'$ in Zeile 3; in Zeile 10 wird $\{f\}$ ausgegeben

$\boxed{V' = \{d, g, h\}}$

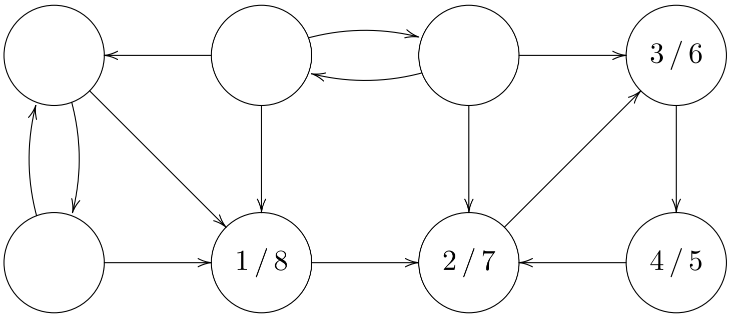
$\Downarrow$ Wähle $g \in V'$ in Zeile 3; in Zeile 10 wird $\{d, g, h\}$ ausgegeben

$\boxed{V' = \emptyset}$

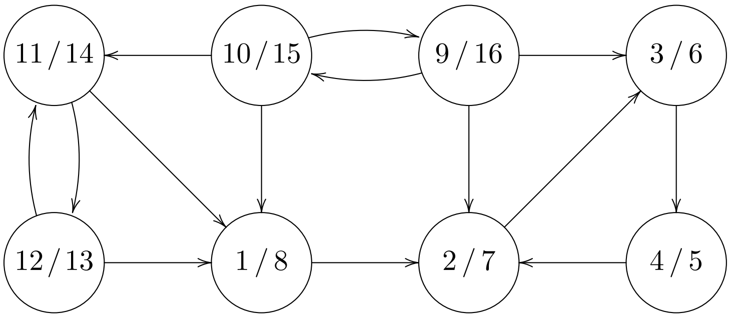
In allen drei dieser exemplarischen Ausführungen werden die starken Zusammenhangskomponenten von G korrekt bestimmt.

Aufgabe 2: Unter den angegebenen Bedingungen führt der Algorithmus Starke Zusammenhangskomponenten zu den nachfolgenden Ergebnissen.

a) Wenn der Algorithmus in der ersten Phase mit dem Knoten f beginnt, so ergibt der zugehörige DFS–Aufruf mit f als Startknoten die folgenden *Push*– und *Pop*–Zeiten:

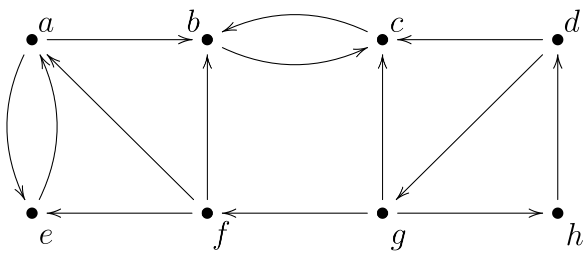


Der Inhalt des Kellers L ist zu diesem Zeitpunkt $[h, d, g, f]$. Wenn nun als nächstes der Knoten c überprüft wird, so führt der zugehörige DFS–Aufruf mit c als Startknoten zu einer Markierung aller restlichen Knoten:



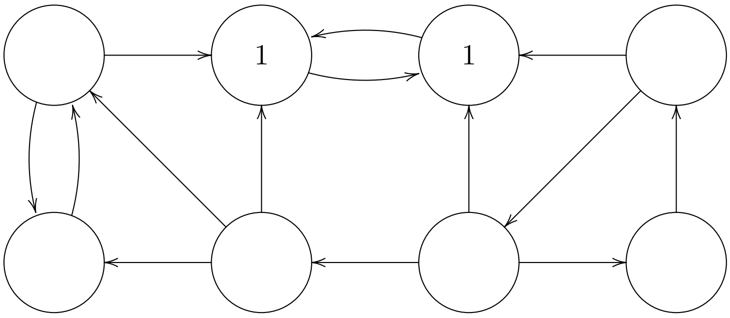
Der Inhalt des Kellers L ist jetzt $[h, d, g, f, e, a, b, c]$. Da nun bereits alle Knoten markiert sind, führt die Verarbeitung der restlichen Knoten a, b, d, e, g, h zu keinen Änderungen mehr.

b) Der umorientierte Graph G' sieht wie folgt aus:

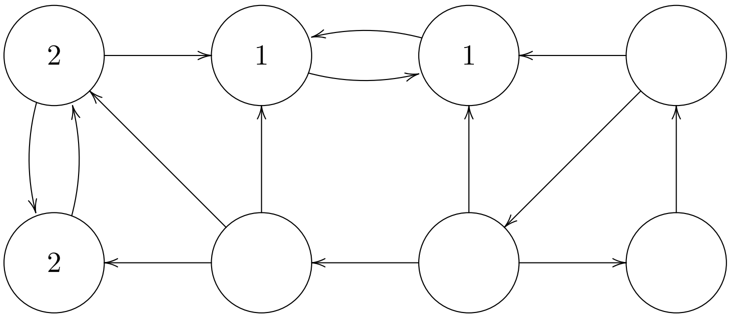


Die starken Zusammenhangskomponenten von G' sind $\{a, e\}$, $\{b, c\}$, $\{d, g, h\}$ und $\{f\}$. Der Graph G' besitzt also die gleichen starken Zusammenhangskomponenten wie G.

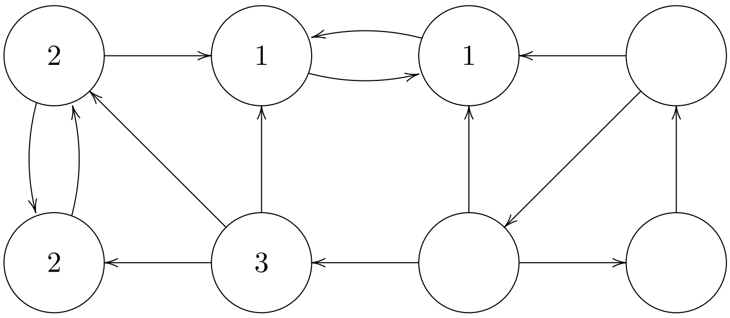
c) Der Inhalt des Kellers L nach der ersten Phase war $[h, d, g, f, e, a, b, c]$. Also beginnt der Algorithmus in der zweiten Phase nun mit einem DFS–Aufruf mit c als Startknoten. Von c abgesehen ist nur b von c aus erreichbar. Diese beiden Knoten stellen die erste ermittelte Komponente dar. Wir markieren sie mit einer Eins:



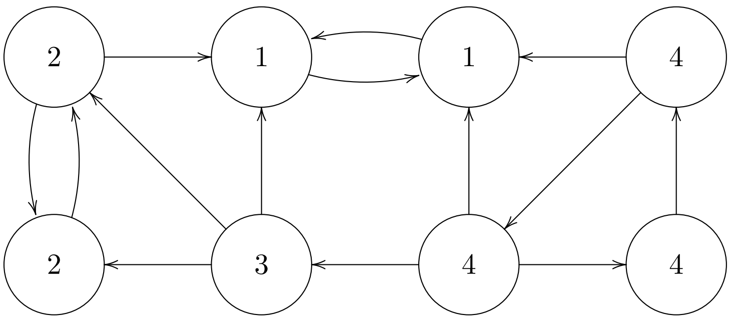
In dem Keller L stehen jetzt noch die Knoten $[h, d, g, f, e, a, b]$. Der nächste Knoten b hat keine Auswirkungen, weil er bereits markiert ist. Dagegen löst der Knoten a ein neues DFS–Verfahren aus. Es ergibt sich die zweite Komponente $\{a, e\}$:



Der nächste Knoten e ist wieder bereits markiert und wird ignoriert. Der folgende Knoten f führt dagegen zur dritten Komponente $\{f\}$:



Jetzt sind in dem Keller L nur noch die Knoten $[h, d, g]$ übrig. Das DFS–Verfahren mit g als Startknoten markiert die letzte Komponente $\{d, g, h\}$:



Alle Knoten sind damit markiert, insbesondere die beiden abschließenden Knoten d und

h, die deshalb ohne weitere Auswirkungen verarbeitet werden. Die gefundenen Knotenmengen $\{b, c\}$, $\{a, e\}$, $\{f\}$ und $\{d, g, h\}$ sind wie erwartet die korrekten starken Zusammenhangskomponenten von G.

Aufgabe 3: Aufgrund der vorgegebenen Knotenreihenfolge für die erste Phase führt der Algorithmus Starke Zusammenhangskomponenten zu den nachfolgenden Ergebnissen.

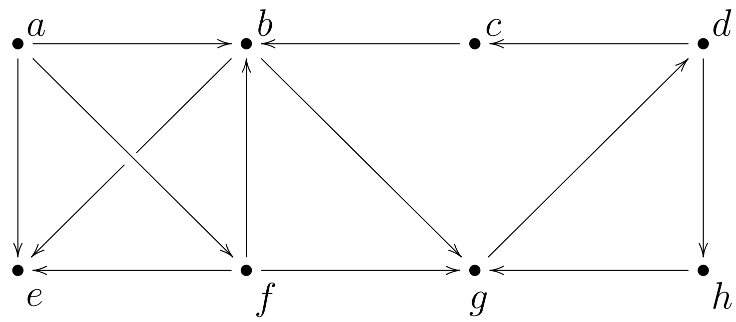
a) Die starken Zusammenhangskomponenten sind $\{a\}$, $\{b, c, d, g, h\}$, $\{e\}$ und $\{f\}$.

b) Die *Push*– und *Pop*–Zeiten sind bei der vorgegebenen Knotenreihenfolge eindeutig und können der folgenden Tabelle entnommen werden:

Knoten	a	b	c	d	e	f	g	h
Push–Zeit $d[v]$	2	8	9	6	15	1	7	5
Pop–Zeit $f[v]$	3	11	10	13	16	4	12	14

Der Keller L enthält nach der ersten Phase die Knotensequenz $[a, f, c, b, g, d, h, e]$. Dabei ist e der oberste Knoten und wird in der nächsten Phase als erstes wieder entfernt.

c) Der umorientierte Graph G' sieht wie folgt aus:



d) Die Verarbeitung der Knoten aus L hat die folgenden Auswirkungen:

Knoten	Reaktion
e	DFS wird mit e gestartet, Komponente $\{e\}$ wird ausgegeben
h	DFS wird mit h gestartet, Komponente $\{b, c, d, g, h\}$ wird ausgegeben
d	Knoten ist bereits markiert, keine Reaktion
g	Knoten ist bereits markiert, keine Reaktion
b	Knoten ist bereits markiert, keine Reaktion
c	Knoten ist bereits markiert, keine Reaktion
f	DFS wird mit f gestartet, Komponente $\{f\}$ wird ausgegeben
a	DFS wird mit a gestartet, Komponente $\{a\}$ wird ausgegeben

Aufgabe 4: Sei x ein beliebiger Knoten auf einem Pfad, der v mit w verbindet. Das erste Teilstück des Pfades (bis zum Knoten x) verbindet v mit x. Das zweite Teilstück des Pfades verbindet analog x mit w, und wir können diesen zweiten Pfad weiter bis nach v verlängern, denn es gibt auch einen Pfad von w nach v (denn w und v liegen in der gleichen starken Zusammenhangskomponente C). Also sind v und x wechselseitig mit Pfaden verbunden, d.h. x liegt ebenso wie v in der gleichen starken Zusammenhangskomponente C, und in dieser liegt auch w.

Mit v und w liegt also jeder beliebige Knoten x eines verbindenden Pfades in der gleichen starken Zusammenhangskomponente C, d.h. der gesamte Pfad muss in C enthalten sein. Dies war zu beweisen. $\square$

Aufgabe 5: In der zweiten Phase des Algorithmus STARKE ZUSAMMENHANGSKOMPONENTEN kommt es nur darauf an, alle von einem bestimmten Startknoten aus erreichbaren Knoten zu markieren. Dies kann mit jedem beliebigen Durchmusterungsalgorithmus geschehen, insbesondere also auch mit dem BFS–Verfahren. Man bevorzugt das DFS–Verfahren allein deshalb, weil man es für die erste Phase des Algorithmus STARKE ZUSAMMENHANGSKOMPONENTEN sowieso benötigt. Bei der Wahl eines anderen Verfahrens für die zweite Phase würde ein erhöhter Implementierungsaufwand anfallen.