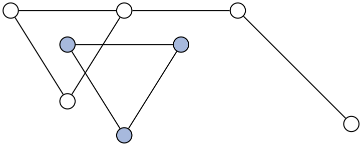
Graphentheorie
06 – Erreichbarkeit, Zusammenhang

Dr. Sven Köhler
Rechnernetze und Telematik
Technische Fakultät
Albert-Ludwigs-Universität Freiburg

Erreichbarkeit



Definition 3.10

Ein Knoten w heißt *von* v *erreichbar*,
wenn es Weg P mit $\alpha(P) = v$ und $\omega(P) = w$ gibt.

Notation: $E_G(v) :=$ Menge aller von v erreichbaren Knoten in G

Erreichbarkeit

Algorithmus 3.2 ERREICHBAR(G, s, p)

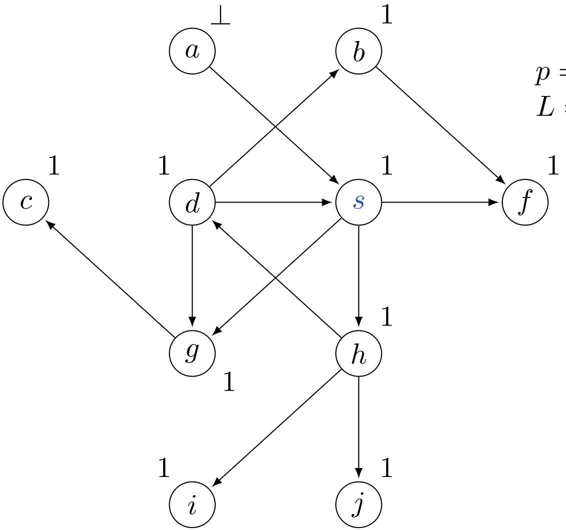
Eingabe: Graph $G = (V, R)$ oder (V, E), Knoten $s \in V$ und Wert $p \in \mathbb{N}$

for each $v \in V$ **do**
  $marke[v] := \perp$

$L := (s)$ ▷ L ist eine Liste
$marke[s] := p$
while $L \neq ()$ **do** ▷ $()$ ist die leere Liste
  $v :=$ entferne erstes Element aus L
  for each $w \in ADJ[v]$ **do** ▷ Adjazenzliste von v
    if $marke[w] = \perp$ **then**
      Füge w am Ende von L ein
      $marke[w] := p$

return $\{v \in V \mid marke[v] = p\}$

Erreichbarkeit – Beispiel (gerichtet)



$p = 1$

$L = s\ f\ h\ g\ d\ i\ j\ c\ b$

Erreichbarkeit

Beobachtung

Algorithmus 3.2 implementiert eine Breitensuche (BFS = breadth-first search). Diese findet die kürzesten Wege von s zu allen Knoten aus $E_G(s)$.

Der Liste L werden der Reihe nach hinzugefügt:

- alle Knoten mit Distanz 0,
- alle Knoten mit Distanz 1,
- alle Knoten mit Distanz 2,
- usw.

Distanz $=$ Länge des kürzesten Weges von s zum Knoten

Erreichbarkeit

Satz 3.11
Algorithmus 3.2 berechnet $E_G(s)$ in Laufzeit $\mathcal{O}(n+m)$.

Beweis, Teil 1.
Initialisierung:

- Laufzeit $\mathcal{O}(n)$

Innere for-each-Schleife:

- Laufzeit $\mathcal{O}(g^+(v))$ bzw. $\mathcal{O}(g(v))$
- Annahme: G liegt in Adjazenzlisten-Repräsentation vor

Äußere for-Schleife:

- Besucht jeden Knoten höchstens einmal
- Benutzt jede Kante höchstens einmal bzw. zweimal
- Laufzeit $\mathcal{O}(n+m)$

$\square$

Erreichbarkeit

Satz 3.11

Algorithmus 3.2 berechnet $E_G(s)$ in Laufzeit $\mathcal{O}(n+m)$.

Beweis, Teil 2.

Zu zeigen: $v \in E_G(s) \;\Rightarrow\; v$ wird markiert

Sei $P = (v_0, r_1, v_1, \ldots, r_k, v_k)$ ein kürzester Weg von s nach v.

Induktionsanfang: $i = 0$

- $s = v_0$ wird markiert und L hinzugefügt

Induktionsschluss:

- v_i wurde L hinzugefügt, hat Distanz i
- v_{i+1} hat Distanz $i+1$
- v_{i+1} wird ebenfalls markiert und L hinzugefügt

$\Rightarrow v = v_k$ wird markiert $\square$

Erreichbarkeit

Satz 3.11

Algorithmus 3.2 berechnet $E_G(s)$ in Laufzeit $\mathcal{O}(n+m)$.

Beweis, Teil 3.

Zu zeigen: $v \notin E_G(s) \;\Rightarrow\; v$ wird *nicht* markiert

Es gibt keinen Weg von s zu v.
Es werden immer nur Nachfolger markiert.

Daher kann v nicht markiert werden. $\square$

Zusammenhang

Definition 3.12

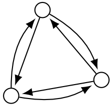
Sei G ein gerichteter oder ungerichteter Graph.

Knoten $v, w \in V(G)$ heißen *stark zusammenhängend* (Notation: $v \leftrightarrow w$) wenn $v \in E_G(w)$ und $w \in E_G(v)$.

$ZK_G(v) := \{w \in V(G) \mid v \leftrightarrow w\}$ heißt *starke Zusammenhangskomponente* von v.

G heißt *stark zusammenhängend* falls $\exists v : ZK_G(v) = V(G)$.

Falls G gerichtet:
Knoten $v, w \in V(G)$ heißen schwach zusammenhängend wenn $v \leftrightarrow w$ in G^{sym}.

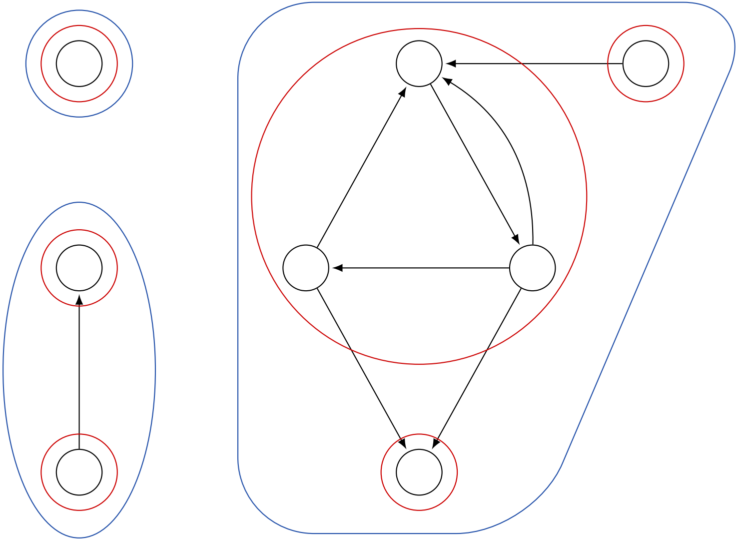


Beobachtung

$ZK_G(v) = ZK_G(w)$ für alle $w \in ZK_G(v)$.

Wenn $ZK_G(v) = V(G)$ für ein $v \in V(G)$ gilt, dann gilt es für alle $v \in V(G)$.

Schwache vs. Starke Zusammenhangskomponenten

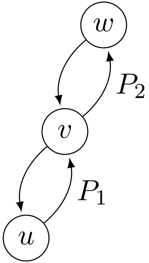


Zusammenhang

Lemma 3.14

Die Relation $\leftrightarrow$ ist eine Äquivalenzrelation.

Beweis.



Reflexivität: $v \leftrightarrow v$

- Folgt aus Definition von $E_G(v)$

Symmetrie: $v \leftrightarrow w \;\Leftrightarrow\; w \leftrightarrow v$

- Folgt aus Definition von $\leftrightarrow$.

Transitivität: $u \leftrightarrow v \wedge v \leftrightarrow w \;\Rightarrow\; u \leftrightarrow w$

- Sei P_1 Weg von u nach v
- Sei P_2 Weg von v nach w.
- Dann $w \in E_G(u)$, denn $P_1 \circ P_2$ ist Weg von u nach w
- Zeige $u \in E_G(w)$ analog

□

Zusammenhangskomponenten

Fakt

Falls $R \subseteq V \times V$ eine Äquivalenzrelation ist, dann liefern die Äquivalenzklassen eine Partition von V.

Daraus folgt:
Es gibt eine Partition $V = ZK_1 \,\dot{\cup}\, ZK_2 \,\dot{\cup} \ldots \dot{\cup}\, ZK_k$,
so dass jedes ZK_i eine starken Zusammenhangskomponente ist.

Zusammenhangskomponenten

Algorithmus 3.3 Zusammenhangskomponenten berechnen

Eingabe: ungerichteter Graph $G = (V, E)$

for each $v \in V$ **do**
  $marke[v] := \perp$

$p := 0$
for each $v \in V$ **do**
  if $marke[v] = \perp$ **then**
    $p := p + 1$
    ERREICHBAR(G, v, p) ▷ setzt $marke[w] := p$ für alle $w \in E_G(v)$

Algorithmus 3.3 hat Laufzeit $O(n + m)$, denn:
ERREICHBAR(G, v, p) hat eigentlich Laufzeit $O(n' + m')$,
wobei n' und m' die Anzahl der Knoten bzw. Kanten in $ZK(v)$ ist.

Zusammenhangskomponenten

Lemma 3.22

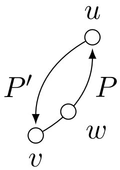
Sei G gerichtet oder ungerichtet, $v \in V(G)$ und $u \in ZK(v) \setminus \{v\}$.

- Jeder Weg P von v nach u (oder u nach v) berührt nur Knoten aus $ZK(v)$.
- Es existiert ein Kreis C, der genau alle Knoten in $ZK(v)$ berührt.

Beweis.

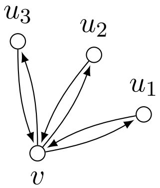
Sei w ein Knoten der von P berührt wird und P' ein Weg von u nach v.

- Dann existiert Weg von v nach w (Prefix von P)
- und ein Weg von w nach v (Suffix von $P \circ P'$).
- Damit ist $w \in ZK(v)$.



Konstruktion von C:

- Sei C_i ein Kreis der v und $u_i \in ZK(v)$ berührt
- Wähle $C = C_1 \circ C_2 \circ C_3 \circ \ldots$



$\square$