3.2 Problemdefinitionen:

PROBLEM 3.1 (s-t-Weg)

Gegeben: Graph $G=(V,E)$, Startknoten s, Zielknoten t

Gesucht: Weg von s nach t, falls einer existiert

Allgemeiner:

PROBLEM 3.2 (Zusammenhangskomponente)

Gegeben: Graph $G=(V,E)$, Startknoten s

Gesucht: Menge aller von s erreichbaren Knoten (→ „Zusammenhangskomponente" von s)

Wege, die die Erreichbarkeit sichern

Beobachtung:

SATZ 3.3

Wenn ein Weg zwischen zwei Knoten s und t in einem Graphen existiert, dann existiert auch ein Pfad.

Beweis:

Sei: $W = s, e_1, v_1, \ldots, v_m, e_{m+1}, t$ ein Weg von s nach t.

Idee:

Eliminiere Kreise auf dem Weg!

Technische Umsetzung:

Betrachte unter allen Wegen einen W^* mit möglichst wenigen Kanten. Angenommen, W^* hat einen doppelt besuchten Knoten, sagen wir v_i:

$$W^* = s, e_1, v_1, \ldots, \underset{\substack{\uparrow \\ \text{erster} \\ \text{Besuch}}}{v_i}, e_{i+1}, \ldots, \underset{\substack{\uparrow \\ \text{letzter} \\ \text{Besuch}}}{v_i}, e_k, \ldots, t$$

Dann gäbe es einen noch kürzeren Weg:

$$W^{**} = s, e_1, v_1, \dots, v_i, e_k, \dots, t$$

nur einmal besucht!

Das widerspricht der Annahme, dass W^* kürzestmöglich ist.
Also hat W^* keinen doppelt besuchten Knoten, ist also ein Pfad! □

Daraus ergibt sich bereits eine Konsequenz:

KOROLLAR 3.4 („logische Zugabe")

Für Problem 3.2 gibt es als Erreichbarkeit sichernde Menge von Kanten immer eine Menge, die keinen Kreis enthält.

Und aus dem Beweis ergibt sich sogar eine Verschärfung:

KOROLLAR 3.5

Für Problem 3.2 gibt es immer eine kreisfreie Menge von Kanten, die die kürzestmögliche Erreichbarkeit sichert.

Das motiviert

DEFINITION 3.6 (Wald, Baum)

(1) Ein <u>Wald</u> ist ein kreisfreier Graph.

(2) Ein <u>Baum</u> ist eine Zusammenhangskomponente in einem Wald.
(Also: ein kreisfreier, zusammenhängender Graph)

(3) Ein <u>aufspannender Baum</u> ist ein Baum, der alle Knoten verbindet. (Manchmal auch: <u>Spannbaum</u>. Englisch: "spanning tree")

3.3 Zusammenhangskomponenten

Idee: Löse Problem durch systematisches Hinzunehmen von Nachbarn!

(Beispiel)

Rot: im Baum, noch neue Nachbarn möglich
Gelb: im Baum.

ALGORITHMUS 3.7 (Graphen-Scan-Algorithmus)

INPUT: Graph $G=(V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt
– d.h. einen die Zusammenhangskomponente von s aufspannenden Baum (Y,T)

1. Sei $R := \{s\}$, $Y := \{s\}$, $T := \emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
   - 2.1 Wähle $v \in R$
   - 2.2 IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1 $R := R \setminus \{v\}$ // v erledigt
   - 2.3 ELSE {
     - 2.3.1 Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$;
     - 2.3.2 Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

SATZ 3.8

(1) Das Verfahren 3.7 ist endlich.
(2) Das Verfahren 3.7 funktioniert korrekt.

Beweis:

(1) Bei jedem Durchlauf der Schleife 2. wird entweder in 2.2.1 ein Element aus R entfernt, oder in 2.3.2 ein Element zu R hinzugefügt und auch zu Y hinzugefügt. Also kann man 2.3.2 nur (n-1)-mal durchlaufen, also auch nur R (n-1)-mal erweitern, also R höchstens n-mal verkleinern. 2. kann also höchstens (2n-1)-mal durchlaufen werden.

(2) Zu jedem Zeitpunkt ist $(Y;T)$ ein s enthaltender Baum, denn
(a) alle Knoten in G sind von s aus erreichbar (und umgekehrt)
(b) neu eingefügte Kanten verbinden die bisherige Knotenmenge Y nur mit bislang nicht erreichbaren Knoten, können also keinen Kreis schließen.

Jetzt müssen wir noch zeigen, dass alle erreichbaren Knoten auch korrekt identifiziert werden.

Angenommen, am Ende gibt es einen Knoten $w \in V \setminus Y$, der in G von s aus erreichbar ist.

Sei P ein s-w-Pfad in G, und sei $\{x,y\}$ eine Kante von P mit $x \in Y$, $y \notin Y$.

Da x zu Y gehört, wurde x auch zu R hinzugefügt. Der Algorithmus stoppt aber nicht, bevor x aus R entfernt wurde. Das wird in 2.2.1 nur vorgenommen, wenn es in 2.2 keine Kante $\{x,y\}$ mit $y \notin Y$ gibt – im Widerspruch zur Annahme. $\square$