

Praktikum

Algorithmen und Datenstrukturen

Wagner/Willhalm

Blatt 1

Wir betrachten einen gerichteten Graphen $G = (V, E)$. Der Graph heißt (stark) **zusammenhängend**, wenn es zwischen je zwei Knoten $u, v \in V$ in beide Richtungen einen (gerichteten) Weg gibt. Eine (starke) **Zusammenhangskomponente** in einem Graphen ist ein maximaler Subgraph, der stark zusammenhängend ist.

Zwei Knoten liegen also genau dann in derselben Zusammenhangskomponente, wenn sie auf einem (nicht notwendig einfachen) gerichteten Kreis liegen. Die Relation „liegen in derselben Zusammenhangskomponente“ auf den Knoten ist eine Äquivalenzrelation. Identifiziert man die Knoten einer jeden starken Zusammenhangskomponente, so erhält man einen azyklischen Graphen.

Ziel des Praktikums ist es, einen Algorithmus zu implementieren, der in der Lage ist, die Zusammenhangskomponenten, die Artikulationspunkte und den Baum auszugeben bzw. weiterzuverarbeiten. Dabei soll mit einer modifizierten Tiefensuche vorgegangen werden:

Prozedur DFS(v)

1. markiere den Knoten v
2. für alle Kanten $\{v, w\}$
3. falls w nicht markiert ist
4. make_set(w)
5. lege w auf den Repräsentanten-Stack
6. DFS(w)
- 6a. falls Top(Repräsentanten-Stack)= w dann entferne w
7. sonst
8. vereinige v mit allen Knoten auf dem Repräsentanten-Stack
9. bis Top(Repräsentanten-Stack) oberhalb oder gleich w

Die Vereinigungen sollen mit einer Union-Find-Struktur mit Pfadkompression und mit linearer Laufzeit realisiert werden.

Um sich die Korrektheit des Algorithmus' klar zu machen, betrachte man den Graphen der durch die bereits besuchten Kanten induziert wird, wobei die Knoten wie im Algorithmus identifiziert werden. Dieser Graph ist azyklisch...