

Theoretische Informatik I

4. Übung

Abgabe: Lösen Sie Aufgabe 4 handschriftlich. Ihre Lösungen geben Sie bitte entweder

- bis zum 12.11.2021 um 13:00 Uhr per Mail
an `julian.pape-lange@informatik.tu-chemnitz.de`
mit *Betreff:* TI1 Hausaufgaben oder
- bis zum 12.11.2021 um 13:00 Uhr im Briefkasten der Professur Theoretische Informatik (vor Raum A10.266.4)

ab.

1. Aufgabe:

Aus jedem *ungerichteten* Graph $G' = (V, E)$ kann ein *gerichteter* Graph G konstruiert werden, indem für jede Kante eine Richtung festgelegt wird. Das heißt, dass die ungerichtete Kante $\{u, v\}$ entweder durch die Kante (u, v) oder (v, u) (aber nicht beide) ersetzt wird.

Welche Bedingungen muss der ungerichtete Graph G' erfüllen, dass es *möglich* ist, seine Kanten so zu richten, dass

- (a) G einen Kreis enthält,
- (b) G keinen Kreis enthält,
- (c) G stark zusammenhängend ist,
- (d) G nicht stark zusammenhängend ist?

Geben Sie für die Fälle (a)-(d) auch an, wie sich ein entsprechender Graph G aus dem ungerichteten Graph G' konstruieren läßt.

2. Aufgabe: Wir betrachten noch einmal die topologische Sortierung bei gerichteten Graphen.

- (a) Formulieren Sie einen Algorithmus, der die topologische Sortierung eines Graphen mit Hilfe der Tiefensuche findet.
- (b) Begründen Sie, warum Ihr Algorithmus eine gültige topologische Sortierung liefert.
- (c) Welche Laufzeit hat Ihr Algorithmus?

Hinweis: Benutzen Sie die Zeitpunkte, die die Tiefensuche für die einzelnen Knoten liefert.

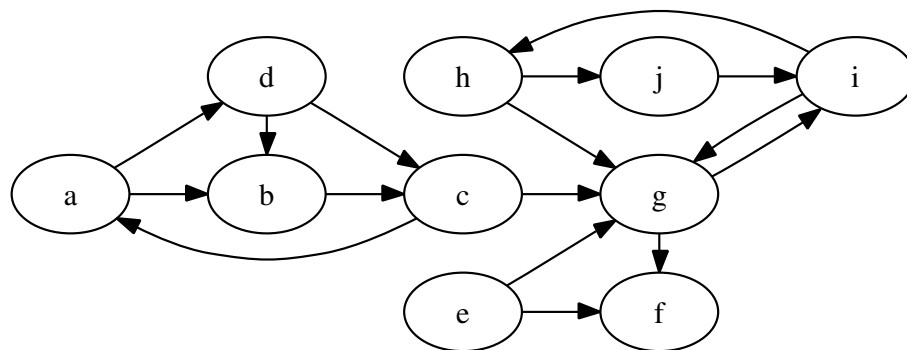
3. Aufgabe:

Wir betrachten die Wege in stark zusammenhängenden Graphen.

- (a) Gibt es einen Graphen $G(V, E)$ mit verschiedenen Knoten $u, t, v \in V$, sodass jeder Weg von u nach v über t geht und jeder Weg von v nach u über t geht?
- (b) Gibt es einen Graphen $G(V, E)$ mit Knoten $u, v \in V$ und Kante $(s, t) \in E$, sodass jeder Weg von u nach v die Kante (s, t) enthält und jeder Weg von v nach u die Kante (s, t) enthält?

4. Aufgabe: ((7+3)P)

- (a) Bestimmen Sie die *starken Zusammenhangskomponenten* des folgenden Graphen.



Demonstrieren Sie dazu den Algorithmus aus der Vorlesung. Gehen Sie davon aus, dass alle Adjazenzlisten alphabetisch geordnet sind und beginnen Sie die erste Tiefensuche bei Knoten b .

- (b) Geben Sie die topologische Sortierung der starken Zusammenhangskomponenten an.