Datenstrukturen und Algorithmen, SS 2024
Prof. Dr. P. Rossmanith
D. Cloerkes, M. Gehnen, D. Mock, N. Speetzen

Theoretische Informatik | RWTH AACHEN UNIVERSITY

Abgabe: 07.06.2024

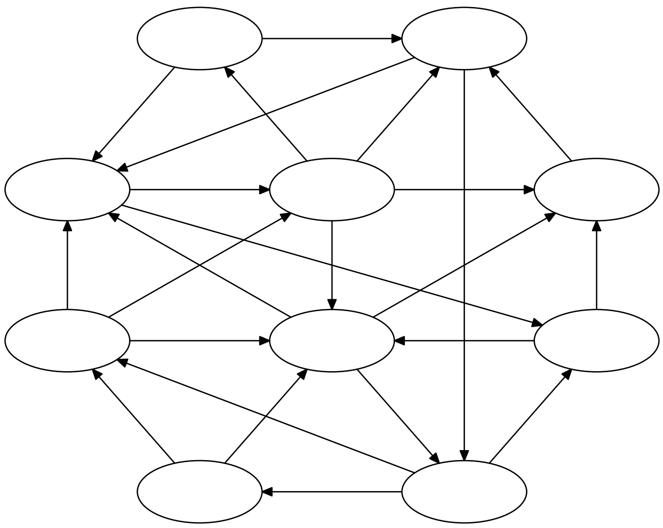
Übungsblatt 07

Aufgabe T23

Zeigen oder widerlegen Sie: Jeder endliche ungerichtete Graph mit mehr als einem Knoten hat mindestens zwei Knoten mit gleichem Grad.

Aufgabe T24

Führen Sie eine Tiefensuche auf dem folgenden Graphen durch. Geben Sie zu jedem Knoten *discovery* und *finish* Zeiten an, und geben Sie dann zu jeder Kante an, welchen Typ sie bezüglich des entstandenen Tiefensuchwaldes hat (Baumkante, Vorwärtskante, Rückwärtskante, Querkante).



Welche starken Zusammenhangskomponenten hat der Graph?

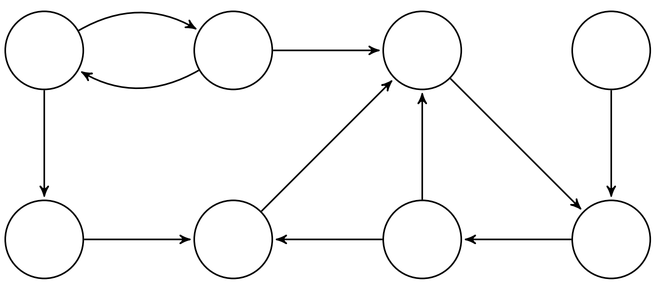
Aufgabe T25

Beweisen oder widerlegen Sie: Ergibt eine Tiefensuche in einem gerichteten Graphen genau eine Rückwärtskante, so liefert jede Tiefensuche in diesem Graphen genau eine Rückwärtskante.

Aufgabe T26

Verwenden Sie den Algorithmus von Kosaraju, um die starken Zusammenhangskomponenten dieses Graphen zu finden.

Markieren Sie zum Darstellen der Lösung im Graphen alle Knoten, welche zur selben starken Zusammenhangskomponente gehören, mit demselben Buchstaben.



Aufgabe H22 (8 Punkte)
Zeigen Sie, wie viele Zusammenhangskomponenten ein ungerichteter Graph höchstens haben kann, wenn alle n Knoten einen Grad von mindestens $\lfloor \frac{n}{2} \rfloor - 1$ haben.
Achten Sie auch auf Ausnahmen für kleine n. Geben Sie Ihre Lösung ggfs. in Abhängigkeit von n an.

Aufgabe H23 (3 + 6 + 6 Punkte)
In einem ungerichteten Graph ist jede Rückwärtskante auch eine Vorwärtskante sowie jede Vorwärtskante auch eine Rückwärtskante. Beweisen Sie die folgenden Aussagen:

a) Ein ungerichteter Baum mit $n \geq 1$ Knoten hat genau $n - 1$ Kanten.

b) Eine Tiefensuche in einem ungerichteten Graphen liefert keine Querkanten.

c) Ergibt eine Tiefensuche in einem ungerichteten Graphen genau eine Rückwärtskante, so liefert jede Tiefensuche in diesem Graphen genau eine Rückwärtskante.

Aufgabe H24 (7 Punkte)
Sie haben Zuweisungen der folgenden Art:

```
y := 3*a*a     d := y + 3     a := c - x     x := 7     c := 2*x - 3
```

Gehen Sie davon aus, dass arithmetische Operationen mit nicht initialisierten Variablen einen Fehler werfen. Wie können Sie eine gültige Folge der Zuweisungen bestimmen, die keinen Fehler wirft? Wann ist dies nicht möglich?
Falls relevant, dürfen Sie annehmen, dass jeder Variable nur einmal ein Wert zugewiesen wird.

Aufgabe H25 (10 Punkte)
Ein *deterministischer Büchi-Automat* (DBA) ist ein Automat, welcher auf einem endlosen Eingabewort läuft und dieses genau dann akzeptiert, wenn während dieses Laufs ein Endzustand des Automaten unendlich oft besucht wird.
Formal ist dieser definiert durch $\mathfrak{A} = (Q, \Sigma, q_0, \delta, F)$, mit der gleichen Bedeutung der Symbole wie bei einem DFA. Für jedes unendliche Wort $\alpha \in \Sigma^\omega$ gibt es einen eindeutigen Lauf ρ von $\mathfrak{A}$ auf α, definiert als $\rho(0) = q_0, \rho(i+1) = \delta(\rho(i), \alpha(i))$. Das Akzeptanzkriterium lautet dann: $\exists^\omega i : \rho(i) \in F$, wobei $\exists^\omega i$ zu lesen ist als *es existieren unendlich viele i*. Somit enthält die durch den Automaten erkannte Sprache genau alle unendlichen Worte, welche einen Lauf beschreiben, der mindestens einen Endzustand unendlich oft besucht.
Unter dieser Aufgabe finden Sie zwei Beispiele für solche Automaten.
Gegeben sei nun ein deterministischer Büchi-Automat $\mathfrak{A}$. Geben Sie einen Algorithmus an, mit dem Sie in Polynomialzeit feststellen können, ob die Sprache, die $\mathfrak{A}$ erkennt, mindestens ein Wort enthält.

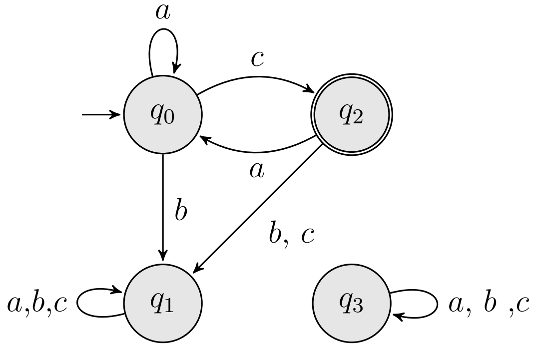


Abb. 1: Ein DBA für die Sprache $L = a^*(ca^+)^\omega$

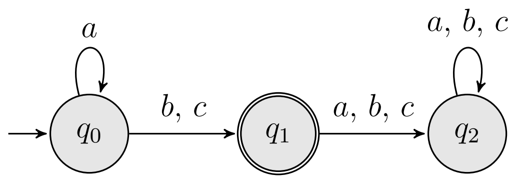


Abb. 2: Ein DBA für die Sprache $L = \emptyset$: