Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

Datum: 30.5.2019

Übungsblatt 08

Aufgabe T27

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 T28

Gegeben sei ein gerichteter Graph G und zwei seiner Knoten s und t.

Es soll festgestellt werden, ob es zwischen s und t mindestens zwei kantendisjunkte Pfade gibt. Überlegen Sie, was das genau bedeutet.

Nun entwirft ein genialer Erfinder folgenden Algorithmus, um dieses Problem zu lösen:

> *Nun, zuerst suchen wir nach irgendeinem Pfad zwischen s und t. Wenn es keinen gibt, dann gibt es natürlich auch keine zwei kantendisjunkte Pfade. Das kann ich mit Tiefensuche in linearer Zeit bewerkstelligen.*
>
> *Falls ich einen solchen Pfad finde, dann lösche ich einfach alle Kanten aus G, die auf dem Pfad liegen. Dadurch kann ich keinesfalls eine Kante aus Versehen zweimal verwenden.*
>
> *Jetzt suche ich einfach wieder nach einem Pfad von s nach t. Finde ich einen, dann gibt es zwei kantendisjunkte Pfade zwischen diesen Knoten. Wenn nicht, dann eben nicht. Die Gesamtlaufzeit ist jedenfalls linear.*

Was sagen Sie dazu?

Aufgabe T29

Der Algorithmus von Bellman und Ford kann die Abstände eines Knotens s zu allen anderen Knoten finden, falls es keinen Kreis mit negativem Gewicht gibt.

Entwerfen Sie einen Algorithmus, der nicht nur feststellen kann, ob es einen Kreis mit negativem Gewicht in einem gerichteten Graphen mit Kantengewichten gibt, sondern einen solchen auch finden und ausgeben kann.

Aufgabe T30

Professor Rossmanith hat in seiner Spezialvorlesung *Kreative Fragestellungen für wehrlose Informatiker* fünf Prüflinge $\{A, B, C, D, E\}$, welche eine mündliche Prüfung bei ihm ablegen müssen. Da Professor Rossmanith keine Lust hat, sich für jeden Prüfling neue Fragen auszudenken, schickt er seine Assistenten, um mehr über die Prüflinge zu erfahren.

Nach monatelanger Recherche berichten die Assistenten ihm, dass es Zwist zwischen einzelnen Prüflingen gibt und diese sich nicht gegenseitig helfen wollen. Der Sachverhalt ist der folgende, dabei bedeutet die Relation $X < Y$, dass Y das eigene Wissen an X weitergeben würde: $A < C, B < A, D < A, B < E, D < B, E < C$. Beachten Sie, dass diese Relation nicht symmetrisch ist. Gibt es eine Prüfungsreihenfolge, in der keine Absprachen zwischen den Prüflingen geschehen? Wenn ja, wie lautet diese?

Aufgabe H22 (3+3+4 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 H23 (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*. 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, leer ist.

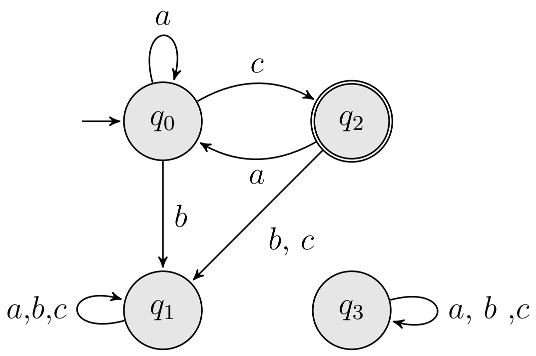

Abb. 1: Ein DBA für die Sprache $L = a^*(ca^+)^\omega$

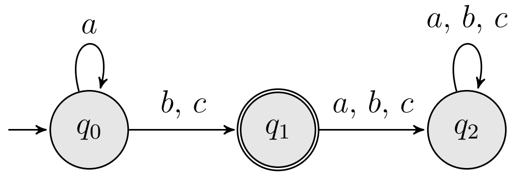

Abb. 2: Ein DBA für die Sprache $L = \emptyset$:

Aufgabe H24 (10 Punkte)

Stewie möchte aus dem Jahr 2019 mit seiner Zeitmaschine in das Jahr 1889 zurück reisen um die Geschichte zu ändern. Leider funktioniert seine Zeitmaschine momentan nur eingeschränkt. Sie hat drei Optionen: Falls man momentan im Jahr x ist, kann man in das Jahr $x + 7$, $2x$ oder (falls x durch drei teilbar ist) $x/3$ reisen. Geben Sie eine kürzestmögliche Folge von Zeitsprüngen an, um in das Jahr 1889 zu reisen.

Hinweis: Verwenden sie Breitensuche und lassen Sie die meiste Arbeit von einem Computer in einer vernünftigen Programmiersprache erledigen.

Reichen Sie Ihren ausführbaren Quellcode als Beleg für ihr Ergebnis mit ein.