Rolf Wanka Erlangen, 14. November 2018

Übungen zur Vorlesung
Effiziente kombinatorische Algorithmen
WS 2018/19
Blatt 4

AUFGABE 9:

Gegeben sei ein zusammenhängender ungerichteter Graph $G = (V,E)$. Eine Knotenüberdeckung $C \subseteq V$ ist eine Knotenmenge, für die gilt: $\forall\{u,v\} \in E : \{u,v\} \cap C \neq \emptyset$. Mit anderen Worten: Jede Kante des Graphen ist zu mindestens einem Knoten aus C inzident.

Das MINIMUM VERTEX COVER-Problem MVC ist die Aufgabe, eine möglichst kleine Knotenüberdeckung zu bestimmen. Dieses Problem ist (in seiner Entscheidungsvariante) NP-vollständig. Deswegen geben wir uns erst einmal mit der „Holzhammer-Methode“ zufrieden, die alle Teilmengen von V der Größe nach daraufhin überprüft, ob sie Knotenüberdeckungen sind. Dieses Verfahren hat eine Laufzeit von $O(|E| \cdot 2^{|V|})$.

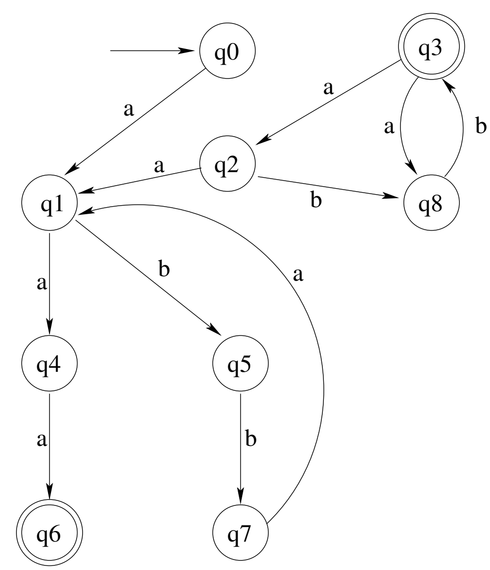
Nehmen Sie nun an, die *größte* zweifache Zusammenhangskomponente besteht aus ℓ Knoten. Enwickeln Sie einen Algorithmus, der eine minimale Knotenüberdeckung in Zeit $O(|V| \cdot |E| \cdot 2^{\ell})$ berechnet.

Benutzen Sie, daß der Superstrukturgraph, den der Algorithmus DFS-2ZK berechnet, ein Baum ist. Betrachten Sie einen Blattknoten G_b des Superstrukturgraphen und den Artikulationsknoten u, über den G_b mit dem Restgraphen verbunden ist.

AUFGABE 10:

(a) Entwerfen Sie einen Algorithmus, der auf starken Zusammenhangskomponenten basiert und entscheidet, ob eine durch einen deterministischen endlichen Automaten gegebene Sprache endlich ist. Welche Laufzeit hat Ihr Algorithmus?

(b) Führen Sie Ihren Algorithmus an folgendem Automaten durch:



AUFGABE 11:
Gegeben sei folgende kontextfreie Grammatik G, die die Sprache $L(G)$ erzeugt (dabei ist es für uns völlig ohne Belang, was $L(G)$ denn nun tatsächlich ist):

$$\begin{array}{lll} S \to A \mid aB \mid aC & B \to S \mid Ba & D \to d \mid dDD \\ A \to B \mid C \mid cAd & C \to D \mid c & \end{array}$$

Kettenregeln sind Regeln, deren linke und rechte Seite jeweils genau ein Nichtterminal-Symbol enthalten, also z.B. $A \to B$. Konstruieren Sie eine Grammatik G' mit $L(G') = L(G)$ ohne Kettenregeln. Die Konstruktion soll in zwei Schritten erfolgen:

1. Finden Sie zyklische Abhängigkeiten, indem Sie einen gerichteten Graphen H aufstellen, dessen Knoten die Variablen und dessen Kanten die Kettenregeln repräsentieren. Wie können nun die zyklischen Abhängigkeiten identifiziert werden? Wie können sie entfernt werden?

2. Entfernen Sie die übrigen Kettenregeln. In welcher Reihenfolge der Variablen muss dies geschehen?