Daniela Kacso
Christian Brandl

Bochum, den 21. 04. 2008

Übungen zur Vorlesung

Effiziente Algorithmen

SS 08

Blatt 2

Hinweis: In der Vorlesung am Dienstag, den 22. April, werden wichtige Informationen zum Prüfungsablauf mitgeteilt.

Aufgabe 2.1

Beweise die folgenden Aussagen über lineare Programme mit beschränkten Differenzen (s. Skript S. 8f.):

a)
1. Wenn der Digraph $G_{\mathcal{P}}$ einen Pfad von i nach j der Länge c enthält, dann kann man aus den linearen Randbedingungen in $\mathcal{P}$ die Ungleichung $x_j \leq x_i + c$ herleiten (Beweis durch Induktion über die Knotenlänge des Pfades).
2. Wenn der Digraph $G_{\mathcal{P}}$ einen negativen Kreis enthält, dann besitzen die linearen Randbedingungen in $\mathcal{P}$ keine Lösung.
3. Sei i_0 ein Wurzelknoten von $G_{\mathcal{P}}$. Wenn der Digraph $G_{\mathcal{P}}$ keinen negativen Kreis enthält, dann erhalten wir eine Lösung von $\mathcal{P}$, indem wir für $i = 0, \ldots, n-1$ die Variable x_i mit der Länge des kürzesten Pfades von i_0 zum Knoten i belegen.

b) Sei $G''_{\mathcal{P}}$ der Graph mit neuem Startknoten s, neuer Kante (s, i_0) mit Kosten p, sowie neuer Kante (j, s) mit Kosten $-q$.
1. Wenn $G''_{\mathcal{P}}$ einen negativen Kreis enthält, so hat $\mathcal{P}$ erweitert um die Bedingungen $x_{i_0} = p$ und $x_j \geq q$ keine Lösung.
2. Wenn $G''_{\mathcal{P}}$ keinen negativen Kreis enthält, dann liefern die von s ausgehenden kürzesten Pfade eine Lösung für $\mathcal{P}$ erweitert um die Randbedingungen $x_{i_0} = p$ und $x_j \geq q$.

Aufgabe 2.2

a) Zeichne den zugehörigen Digraphen für das lineare Programm mit beschränkten Differenzen in Beispiel 2.1.4 des Skripts ohne die Randbedingungen $x_{23} = p$ und $x_0 \geq 0$. Begründe, warum jeder Knoten ein Wurzelknoten ist. Modifiziere dann den Graphen so, dass er auch die beiden obigen Randbedingungen berücksichtigt.

b) Aus der Vorlesung ist bekannt, dass das lineare Programm genau dann eine Lösung besitzt, wenn der Digraph keinen negativen Kreis enthält. Zeige, dass die Bedingung

$$p \geq b_{i+16} + b_{i+8} + b_i \quad \text{für } i = 0, \ldots, 7$$

für die Existenz einer Lösung notwendig aber nicht hinreichend ist.

Aufgabe 2.3

Konstruiere einen Algorithmus, der einen DAG mit Wurzelknoten als Eingabe erhält und eine topologisch sortierte Liste seiner Knoten ausgibt. Der Algorithmus soll Laufzeit $O(m)$ haben, wobei m die Anzahl der Kanten ist.

Aufgabe 2.4

Führe Dijkstras Algorithmus an folgendem Beispiel durch. Gib für jede Iteration der while-Schleife die Menge T und die Distanzfunktion d an.