

Symbolische Berechnung kürzester Wege

Daniel Sawitzki

13.7.2004

Graphprobleme spielen eine wichtige Rolle in vielen Anwendungsgebieten. Oft müssen sehr große Graphen verarbeitet werden, die jedoch nicht zufällig strukturiert sind, sondern starke Regelmäßigkeiten besitzen. Dann kann eine *symbolische* Darstellung durch Datenstrukturen für boolesche Funktionen wesentlich kleiner sein als eine explizite Aufzählung aller Knoten und Kanten durch Adjazenzlisten.

Wir verwenden *Ordered Binary Decision Diagrams (OBDDs)* zur Darstellung der *charakteristischen Funktion* $E(u, v) = 1 :\Leftrightarrow (u, v) \in E$ eines gerichteten Graphen $G = (V, E)$. Entsprechende symbolische Graphalgorithmen müssen sich im Wesentlichen auf die von OBDDs angebotenen funktionalen Operationen beschränken (z. B. binäre Synthese und Quantifizierung) um Probleme in G zu lösen. Dabei muss oft ganz anders vorgegangen werden als in klassischen Algorithmen, da Iterationen über einzelne Knoten und Kanten vermieden werden sollen.

In diesem Vortrag wird ein symbolischer Algorithmus für das All-Pairs Shortest-Paths (APSP) Problem vorgestellt. Er arbeitet auf der OBDD-Darstellung eines gewichteten Graphen und benötigt $\mathcal{O}(\log^2(NB))$ OBDD-Operationen für die Knotenanzahl N und das maximale Kantengewicht B .

Bei der Analyse der Gesamtlaufzeit möchte man idealerweise zeigen, dass brauchbare symbolische Algorithmen effizient sind, wenn Eingabe- und Ausgabe-OBDD klein sind. Wir zeigen, dass eine konstante Breite der (vollständigen) Eingabe- und Ausgabe-OBDDs eine Gesamtlaufzeit von $\mathcal{O}(\log^3(NB))$ impliziert. Entscheidend für den Beweis ist, dass alle während des Algorithmus erzeugten OBDDs ebenfalls konstante Breite besitzen. Diese Eigenschaft ist unter Graphkompositionsoperationen wie Vereinigungen und Kreuzprodukten abgeschlossen. Sie gilt jedoch nicht für einen alternativen Ansatz, der auch nicht-positive Kantengewichte zulässt.

Schließlich werden Anpassungen des symbolischen APSP-Algorithmus für verwandte Probleme angesprochen.