

Vorlesungsankündigung

- Wintersemester 2005/2006 -

Parameterisierte Algorithmen

Dr. Henning Fernau

FB IV - Informatik

Wohl jeder angehende Informatiker hat schon von der P/NP-Problematik gehört. Vereinfachend gesprochen folgt aus der Annahme, dass sich Probleme, die mit nichtdeterministischen Maschinen in Polynomzeit lösbar sind, eben nicht mit deterministischen Maschinen in Polynomzeit lösen lassen: Zur exakten Lösung NP-harter Probleme wird superpolynomielle Zeit benötigt. Allerdings tauchen NP-harte Probleme in großer Zahl in unterschiedlichsten Anwendungen auf. Hier hilft es dem Praktiker nichts zu hören, dass sein Problem NP-hart ist und daher "praktisch" nicht mit Rechnerhilfe lösbar ist. Im Gegenteil lehrt die Erfahrung, dass durch ingenieurmäßige Herangehensweise an solche Probleme oft sehr schnell exakte Lösungen für NP-harte Probleminstanzen gefunden werden können. Lässt sich dieser scheinbare Widerspruch auflösen? Diese Fragestellung ist einer der möglichen Zugänge zur parameterisierten Algorithmik. Konkret würde hier nun versucht, typische Teile der Eingabe zu beschreiben, die in den vorkommenden Instanzen "klein" sind und daher eine exponentielle Verknüpfung mit der Laufzeit des Algorithmus "vertragen".

Formaler ausgedrückt, versuchen wir eine zweidimensionale Sicht auf Probleminstanzen zu entwickeln: neben der Gesamtgröße n der Eingabe fließt ein Teil der Eingabe als so genannter *Parameter* k ein. Ein klassisches Problem mag dies verdeutlichen: Beim *Knotenüberdeckungsproblem* wird gefragt, ob ein gegebener Graph $G = (V, E)$ eine Knotenüberdeckung der Größe höchstens k besitzt, wobei unter einer Knotenüberdeckung eine Knotenmenge C zu verstehen ist mit der Eigenschaft, dass jede Kante $e \in E$ mindestens einen ihrer Eckpunkte in C zu liegen hat. In der klassischen Sicht auf NP-harte Probleme würde gefolgert, dass es vermutlich keinen polynomiellen Algorithmus für beliebige Knotenüberdeckungsinstanzen (G, k) gibt. In der parameterisierten Sicht wäre man froh, einen (deterministischen) Algorithmus zu finden, der eine Laufzeit der Form $p(|G|)f(k)$ hätte, wobei p ein Polynom ist und f eine beliebige Funktion (unabhängig von $|G|$). Dies führt uns auf die Problemklasse FPT (fixed parameter tractable). Im konkreten Beispiel ist dies recht einfach einzusehen: Man kann einen einfachen Suchbaum der Tiefe k aufbauen durch die Vorschrift, sich irgendeine bislang unbedeckte Kante herauszugreifen und dann gemäß den beiden Überdeckungsmöglichkeiten zu verzweigen. Die Laufzeit solch einer Vorschrift lässt sich mit $O(|G|2^k)$ abschätzen: sollte nämlich keiner der Suchäste zu einer gültigen Knotenüberdeckung führen, so ist die Ausgangsfrage des Knotenüberdeckungsproblems für die vorliegende Instanz zu verneinen.

Wir werden an zahlreichen Beispielen Techniken einüben, um möglichst gute FPT-Algorithmen aufzufinden: neben Suchbäumen sind hier sogenannte parameterisierte Kerne, Umparameterisierungen und Graphparameter zu nennen. Die Analyse der entsprechenden Algorithmen stellt auch ein besonderes Teilgebiet dar. Daneben werden wir auch die "harten" Seiten dieser Theorie kennenlernen.

Die Vorlesung kann in der Informatik-Diplomprüfung in Theoretischer Informatik geprüft werden.

Voraussetzung: Kenntnisse, die einem Vordiplom in Informatik (oder Mathematik) entsprechen

Termine:	Vorlesung:	Montag	16-18 Uhr	HZ 204	Beginn: 31.10.
	Übung (14tg):	Dienstag	10-12 Uhr	H 6	Beginn: 8.11.