ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Departement Informatik

Open Class – Sieben Wunder der Informatik

Prof. Dr. Juraj Hromkovič

Übungsaufgaben – Blatt 5

Zürich, 30. November 2005

Zusammenfassung

Eine vollständige Darstellung des heutigen Vortrages finden Sie im Buch

> Juraj Hromkovič: *Randomisierte Algorithmen.*
> Teubner-Verlag (Juli 2004), ISBN 3-519-00470-4,
> Seiten 11–22, 105–112.

Eine ausführliche Zusammenfassung finden Sie auch in dem Buch

> Juraj Hromkovič: *Theoretische Informatik.*
> Teubner-Verlag (Februar 2004), ISBN 3-519-10332-X,
> Seiten 274–284.

Wir bieten daher an dieser Stelle lediglich eine kurze Zusammenfassung.

Gemäss Wörterbüchern ist ein Objekt *zufällig,* wenn es nach keinem Plan gebaut wurde. Eine Erscheinung ist zufällig, wenn sie unvorhersehbar ist.
Wir wissen nicht, ob echter Zufall existiert. Dank der Quantenmechanik, der Evolutionsbiologie und der Entwicklung effizienter zufallsgesteuerter Verfahren in der Informatik glaubt man heutzutage an die Existenz echten Zufalls. Es besteht aber keine Möglichkeit, diese Existenz zu beweisen.
Wir werden im folgenden den Zufall als eine Quelle von Effizienz aufzeigen.
Angenommen, man gibt uns die folgende Aufgabe. Auf zwei Rechnern R_I und R_{II} ist jeweils eine Datei, d. h. eine 0-1-Folge, der Länge n Bits gespeichert. Die Aufgabe besteht darin, durch Kommunikation zwischen den Rechnern zu überprüfen, ob die beiden Dateien (völlig) übereinstimmen oder nicht. Die Komplexität dieses Vergleichs misst man in der Anzahl der kommunizierten Bits.
Man kann beweisen, dass das beste erdenkliche Verfahren gerade in der folgenden naiven Art und Weise besteht, nämlich darin, dass R_I seine gesamte Information, also alle n Bits, an R_{II} schickt und R_{II} nun den eigentlichen Vergleich durchführt. Die Komplexität der Kommunikation ist also nachweislich so gross wie n. Ist etwa $n = 10^{16}$ (das entspricht

einem Datenbestand von ca. 250'000 DVDs), so können wir uns soviel Kommunikation nicht leisten.
Nun werden wir ein randomisiertes Protokoll für diese Aufgabe aufbauen. Dazu interpretieren wir die n Bits beider Rechner als (jeweils) eine Zahl aus dem Bereich von 0 bis $2^n - 1$. Das Protokoll arbeitet wie folgt:

Schritt 1 R_I wählt zufällig eine Primzahl p aus den Primzahlen im Bereich von 2 bis n^2. Für jede Primzahl ist dabei die Wahrscheinlichkeit, dass gerade sie gezogen wird, genauso gross wie für jede andere.

Schritt 2 R_I ermittelt den Rest bei der Division der dort gespeicherten (n Bits langen) Zahl durch p und kommuniziert diese Zahl s sowie p (etwa in Binärdarstellung) an R_{II}.

Schritt 3 R_{II} ermittelt den Rest bei der Division der *dort* gespeicherten (n Bits langen) Zahl durch p und vergleicht nun diese Zahl t mit s.
Falls $s \neq t$, dann gibt R_{II} „ungleich“ aus.
Falls $s = t$, dann gibt R_{II} „gleich“ aus.

Aufgaben

Aufgabe 14

Es sei PRIM(m) die (endliche) Menge aller Primzahlen, die nicht grösser als m sind. Betrachten wir für $n = 7$ die Eingaben

$x = 1001001$	für R_I und
$y = 0101011$	für R_{II}.

a.) Listen Sie die Elemente der Menge PRIM(n^2) auf, d. h. listen Sie alle Primzahlen auf, aus denen R_I eine auswählt.

b.) Wählen Sie („zufällig“) drei dieser Primzahlen aus, und führen Sie die Arbeit des Protokolls für jede der von Ihnen gewählten Primzahlen durch.

10 Punkte

Aufgabe 15

Berechnen Sie die genaue Fehlerwahrscheinlichkeit des Protokolls für die Eingaben aus Aufgabe 14. Wie kann man die Fehlerwahrscheinlichkeit des Protokolls für *beliebige* Werte für x und y einschränken? **10 Punkte**

Aufgabe 16

Ändern Sie in Aufgabe 14 das Protokoll, indem Sie statt aus PRIM(n^2) eine Primzahl aus PRIM(n^3) wählen lassen. Wächst dabei die Kommunikationskomplexität? Wie gross ist jetzt die Fehlerwahrscheinlichkeit für $x = 1001001$ und $y = 0101011$? **10 Punkte**

Bonus-Aufgabe 5

Der Primzahlsatz besagt, dass es unter den natürlichen Zahlen bis zur Zahl m ungefähr $\frac{m}{\ln m}$ viele Primzahlen gibt. Also gibt es in PRIM(n^2) ungefähr $\frac{n^2}{2 \ln n}$ viele Primzahlen. Wie wir gezeigt haben, gibt es für alle möglichen Eingabepaare x und y, die sich unterscheiden, höchstens $n - 1$ schlechte Primzahlen, also solche, die den Unterschied zwischen x und y nicht bezeugen.
Modifizieren Sie das Kommunikationsprotokoll so, dass R_I nun eine Primzahl aus der Menge PRIM(n^4) statt aus der Menge PRIM(n^2) zufällig wählt.

a.) Wie wächst dabei die Kommunikationskomplexität?

b.) Wie ändert sich in diesem Fall die Fehlerwahrscheinlichkeit?

c.) Was ist besser? Zweimal hintereinander die Arbeit des ursprünglichen Protokolls zu simulieren und „gleich“ auszugeben, wenn bei *beiden* Simulationen „gleich“ ausgegeben wurde (und ansonsten „ungleich“), oder das vorgeschlagene modifizierte Protokoll zu verwenden?

10 Bonus-Punkte

Ihre Lösungen zu den Aufgaben können Sie entweder persönlich bei der Open-Class-Veranstaltung am 7. Dezember 2005 abgeben oder bis zum 7. Dezember per E-Mail (möglichst als PDF-Datei) an `hjb@inf.ethz.ch` oder per Post an folgende Adresse schicken:

Dr. Hans-Joachim Böckenhauer
Informationstechnologie und Ausbildung
ETH Zentrum CAB F 11.1
Universitätsstrasse 6
8092 Zürich

Bitte vergessen Sie nicht, Ihre Lösung mit Ihrem Namen und Ihrer E-Mail-Adresse zu versehen.
Falls Ihre Lösung uns bis zum 5. Dezember 2005 erreicht, können Sie die korrigierte Lösung bereits in der Veranstaltung am 7. Dezember abholen (sonst eine Woche später).