28. Zufall in der Informatik

(bzgl. fehlender Aufzeichnungen orginal aus Bodirksy-Skript übernommen)

Eine der wichtigsten Klassen von Problemen in der theoretischen Informatik ist die Klasse der Probleme, die ein Computer in polynomieller Zeit lo?sen kann. Polynomiell bedeutet hier: polynomiell in der Größe der Eingabe. Wenn n die Eingabegröße bezeichnet, dann ist also ein Algorithmus, der stets mit n5 + 1000 Rechenschritten auskommt, polynomiell, aber ein Algorithmus, der manchmal 2n Rechenschritte benötigt, nicht. Die Klasse von Problemen mit einem polynomiellen Algorithmus wird mit P bezeichnet. Eine formale Definition dieser Klasse werden Sie in den einschlägigen Informatikvorlesungen kennenlernen.
In der Praxis ist man aber auch oft mit einem Algorithmus zufrieden, der Zufalls- bits verwenden darf, und dessen Laufzeit im Erwartungsfall polynomiell ist. Die Klasse aller Probleme, die von einem solchen Algorithmus gelöst werden können, nennt man ZPP (Zero-Error Probabilistic Polynomial Time). Interessanterweise kennt man kein Problem in ZPP, von dem man nicht auch wüsste, dass es in P liegt. Lange Zeit hatte das bereits in Abschnitt 3.3 erwähnte Primalitätsproblem diesen Status: man kennt einen randomisierten Algorithmus mit erwartet polynomieller Laufzeit, aber man wusste nicht, ob das Problem in P ist. Aber wie wir bereits verraten haben, weiss man mittlerweile (seit 2002), dass es auch einen polynomiellen deterministischen (d.h., nicht randomisierten) Algorithmus fu?r den Test auf Primalität gibt.
Eine andere interessante Art von randomisierten Algorithmen ist die folgende. Anstatt zu fordern, dass die Laufzeit des Algorithmus im Erwartungsfall polynomial ist22, fordert man, dass der Algorithmus immer polynomial ist, aber nur mit großer Wahrscheinlichkeit das richtige Ergebnis liefern muss23. Ein Problem, von dem man einen solchen Algorithmus kennt, von dem man aber nicht weiss, ob es in P liegt, wird im nächsten Abschnitt eingeführt.