Über beschränkte Interaktion in der Kommunikationskomplexität

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt
beim Fachbereich Biologie und Informatik
der
Johann Wolfgang Goethe–Universität
in Frankfurt am Main

von
Hartmut Klauck
aus Detmold

Frankfurt a.M. 2000

DF1

Vom Fachbereich Biologie und Informatik der Johann Wolfgang Goethe–Universität als Dissertation angenommen.

Dekan: Prof.Dr. Karl-Dieter Entian .....

Gutachter: Prof.Dr. Georg Schnitger, Prof.Dr. Juraj Hromkovič .....

Datum der Disputation: 5.12.2000 .....

Zusammenfassung

Diese Arbeit beschäftigt sich mit dem Einfluß beschränkter Interaktion auf die Effizienz von Kommunikationsprotokollen. Es werden Resultate über die Eigenschaften der Kommunikationskomplexität wie auch über deren Anwendungen auf andere Berechnungsmodelle vorgestellt.

In der Theorie der Kommunikationskomplexität wird folgendes Modell untersucht: Zwei Spieler erhalten jeweils eine Eingabe. Sie müssen eine Funktion auf der Konkatenation dieser Eingaben berechnen. Das betrachtete Komplexitätsmaß ist die benötigte Kommunikationslänge. Weiterhin kann es sein, daß die Fähigkeiten der Spieler zur Interaktion eingeschränkt sind, das heißt, daß nur eine beschränkte Anzahl von Kommunikationsrunden erlaubt ist, im Extremfall nur ein Monolog eines Spielers.

Diese Arbeit beschreibt Resultate, welche zeigen, daß einige Funktionen im Kommunikationsmodell wesentlich effizienter berechnet werden können, wenn hinreichend viele Kommunikationsrunden zur Verfügung stehen. Solche Rundenhierarchien waren zuvor für deterministische und probabilistische Kommunikation bekannt. Die Arbeit verbessert das Ergebnis für probabilistische Kommunikation und zeigt eine Rundenhierarchie für nichtdeterministische Kommunikation mit beschränktem Nichtdeterminismus.

Desweiteren wird quantenmechanische Kommunikation betrachtet. In diesem Modell tauschen die Spieler Partikel aus, welche den Regeln der Quantenmechanik gehorchen. Es wird gezeigt, daß quantenmechanische Einwegkommunikation der Kommunikation mit unbeschränkten Runden unterlegen ist. Quantenmechanische Einweg Protokolle mit beschränktem Fehler können für totale Funktionen exponentiell schlechter als deterministische Zweiweg Protokolle sein.

Weiterhin wird quantenmechanische Las Vegas Kommunikation untersucht. Es wird gezeigt, daß quantenmechanische Las Vegas Protokolle mit vielen Runden für eine bestimmte totale Funktionen polynomiell effizienter sein können als klassische probabilistische Protokolle. Dann wird gezeigt, daß quantenmechanische Las Vegas Einweg Protokolle für totale Funktionen nie wesentlich effizienter sein können als deterministische Einweg Protokolle. Auch hier ist Interaktion also wesentlich.

Eine Analyse ihrer internen Kommunikationseigenschaften erlaubt oft, das Verhalten von Berechnungsmodellen zu analysieren. Dieser Ansatz erlaubt dann, untere Schranken für die Kommunikation auf andere Modelle zu übertragen, und dort untere Schranken für bestimmte Resourcen zu beweisen. Oft gilt, daß die zu untersuchenden Kommunikationsprobleme mit beschränkter Interaktion gelöst werden.

Es werden Anwendungen der Kommunikationsergebnisse für endliche Automaten mit beschränkter Reversalkomplexität und Einweg Automaten gezeigt. Die Nečiporuk Methode für untere Schranken für die Länge von Formeln wird mit Hilfe von Einweg Kommunikation neu hergeleitet und verallgemeinert, so daß untere Schranken für probabilistische, für beschränkt nichtdeterministische und für Quanten Formeln folgen. Außerdem werden untere Schranken für die Tiefe von monotonen Schaltkreisen mit beschränktem Nichtdeterminismus bewiesen.

Die meisten der Resultate nutzen aus, daß Kommunikationsprotokolle mit beschränkter Interaktion häufig ineffizient sind.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Fragestellungen der Arbeit

Kommunikation ist ein wichtiger Bestandteil der Lösung von Berechnungsaufgaben aller Art. Es gibt viele Aspekte von Kommunikation, welche in verschiedenen Anwendungen und Situationen relevant sind, wie Sicherheit, Privatheit von Information, Fehlertoleranz. Einer der grundlegenden Aspekte ist Effizienz. In der auf Shannon zurückgehenden Informationstheorie (siehe z.B. [G90]) wird ein bestimmtes Einwegkommunikationsmodell (mit oder ohne Fehler bei der Datenübertragung) betrachtet. Die Grundfrage ist dabei die Effizienz von Kodierungen. Aber viele Kommunikationsprozesse sind komplizierter und enthalten Interaktion zwischen den kommunizierenden Parteien.
Im folgenden wird ein formales Kommunikationsmodell [Y79] betrachtet, in welchem zwei Spieler das Ziel verfolgen, eine Funktion auf der Konkatenation ihrer privaten Eingaben zu berechnen. Die Spieler, gewöhnlich Alice und Bob genannt, könnten beispielsweise während eines Telefongesprächs versuchen, herauszufinden, ob es eine CD gibt, welche beiden gefällt. Wenn beide einen ungewöhnlichen Geschmack sowie relativ große Sammlungen besitzen, kann eine solche Kommunikation sehr aufwendig werden. Allgemeiner betrachtet erhält jeder Spieler eine Eingabe aus $\{0,1\}^n$, und das Ziel ist es, eine Funktion auf der Konkatenation der Eingaben zu berechnen (im Beispiel also die Frage zu beantworten, ob die Mengen der CDs, welche die beiden mögen, disjunkt sind). Sicherlich kann ein solches Problem nur durch Kommunikation gelöst werden. Dabei nehmen wir an, daß diese Kommunikation nach einem vorher vereinbarten Schema abläuft, einem sogenannten Protokoll, welches beschreibt, wie ein Spieler aus seiner Eingabe und den bisher erhaltenen Nachrichten seine nächste Nachricht ermittelt.
Die Theorie der Kommunikationskomplexität befaßt sich mit der Frage, wieviel Kommunikation Alice und Bob benötigen, und welche Struktur ihre Kommunikation haben sollte. Dabei wird angenommen, daß Alice und Bob

unbeschränkte Berechnungskraft besitzen und all ihre internen Berechnungen kostenfrei sind. Die Spieler kennen allerdings die Eingabe des jeweils anderen Spielers nicht. Dies erlaubt, sich auf den reinen Kommunikationsbedarf zu konzentrieren.
Diese Arbeit beschäftigt sich mit der Erforschung beschränkter Interaktion im formalen Kommunikationsmodell. Die Alltagsintuition, welche den mathematischen Resultaten zugrundeliegt, ist, daß Interaktion notwendig ist, um bestimmte Kommunikationsaufgaben zu lösen. Selten ist ein Monolog so nützlich wie eine (fruchtbare) Diskussion. Und Interaktion ist eine der wesentlichen Eigenschaften, welche Yaos Kommunikationsmodell von Shannons Kanalmodell unterscheiden.
Man betrachte zum Beispiel die folgende Situation: Alice besitzt eine sehr gute Kenntnis der Lokalitäten in einer Stadt (vielleicht besitzt sie eine photographische Erinnerung an einen Stadtplan). Bob kennt sich überhaupt nicht aus, weiß aber, wo die Party stattfindet, zu der beide gehen wollen (eine Adresse). Während nun weder Alice noch Bob das Orientierungsproblem alleine lösen können, geht es, wenn beide kommunizieren. Sollte Alice allerdings Bob nicht zu Wort kommen lassen, so bleibt ihr nichts anderes übrig, als den ganzen Inhalt des Stadtplans vorzutragen, bis Bob den Weg weiß. Das ist sicherlich ineffizient im Vergleich zu der Lösung, in der Bob das Ziel benennt, und Alice den besten Weg sucht. Es kommt auf die richtige Interaktion an. Sollte diese spezielle Interaktion nicht erlaubt sein, wird das Kommunikationsproblem äußerst schwierig, wenn nicht praktisch unlösbar.
Wir schließen also, daß es wichtig ist zu wissen, in welchen Situationen welche Beschränkungen der Interaktion relevant sind. Aber erstens haben wir hiermit noch keine mathematische Analyse, und zweitens kann die Situation komplizierter sein. Gibt es Funktionen, die in k Runden effizienter zu berechnen sind als in $k-1$ Runden? Was, wenn Alice und Bob probabilistisch arbeiten, oder gar ein Kommunikationsmedium nutzen, in dem sie verschieden polarisierte Photonen austauschen, welche den Regeln der Quantenmechanik folgen statt der klassischen Informationstheorie? Es ist notwendig, die Intuition in solchen Situationen durch formale Beweise zu erhärten und Abhängigkeiten zu quantifizieren.
Eine weitere wichtige Frage ist, ob es Berechnungsmodelle gibt, welche implizit interaktionsbeschränkte Kommunikationsprozesse auszuführen haben. Kommunikation scheint ein wesentlicher Bestandteil aller Berechnungen zu sein, denn viele Berechnungsmodelle lassen sich so auffassen, daß einfache Bestandteile, die zu elementaren Operationen fähig sind, durch Kommunikationsmedien verbunden sind.
Im Schaltkreismodell sind beispielsweise die Gatter durch Kanten verbunden, welche Ergebnisse der Berechnungen der Gatter kommunizieren. Im Modell der endlichen Automaten kommuniziert der Zustand Information von einem Ende der Eingabe zum anderen. Im Turingmaschinenmodell kommunizieren die Arbeitsbänder Ergebnisse usw.

Yao [Y79] hat das zwei Spieler Kommunikationsmodell als Verallgemeinerung von Betrachtungen über den Informationsfluß in einem formalen Modell für VLSI Chips definiert. Das Modell ist seitdem auf viele Probleme erfolgreich angewendet worden, siehe die Monographien [Hr97], [KN97].
Es gibt zwei wesentliche Aspekte der Forschung im Kommunikationsmodell. Erstens wurde eine erstaunliche Anzahl von Anwendungen gefunden, welche es erlauben, untere Schranken für die Kommunikationskomplexität auf bestimmte Resourcen in diesen Modelle zu übertragen (wobei sich z.B. untere Schranken für die Tiefe von Schaltkreisen, die Fläche von VLSI Chips, die Größe endlicher Automaten usw. ergeben). Diese Forschung konzentriert sich auf die Anwendungen von Kommunikationsergebnissen, und auf den Beweis unterer Schranken für konkrete Probleme. Der Vorteil des Ansatzes besteht darin, daß untere Schranken für die Kommunikationskomplexität im allgemeinen einfacher zu beweisen sind, da im Kommunikationsmodell alle überflüssige Struktur abgestreift ist.
Der zweite Forschungsansatz besteht in der Untersuchung der mathematischen Eigenschaften des Kommunikationsmodells. Es ergibt sich hier ein Kontext, in dem Fragen über die Natur von Berechnungsmodi wie Nichtdeterminismus, Probabilismus usw. untersucht werden können. Oft sind solche Untersuchungen auch relevant für Anwendungen. Als Beispiel soll eine untere Schranke für die Tiefe monotoner Schaltkreis für das Matchingproblem dienen. In der Arbeit [RW92] wird im Beweis eine untere Schranke für die probabilistische Kommunikationskomplexität eines bestimmten Problems eingesetzt, obwohl deterministische Schaltkreise betrachtet werden. Das eher von der Theorie motivierte Ergebnis über probabilistische Kommunikation fand also später eine Anwendung.
Abgesehen von den eher klassischen Berechnungsmodi, wie Probabilismus und Nichtdeterminismus, untersuchen wir auch quantenmechanische Kommunikation. In den letzten Jahren ist das Interesse am Thema quantenmechanischer Berechnungsmodelle sprunghaft angestiegen. Die grundlegenden Kommunikationsprobleme, die beim Entwurf klassischer Rechner auftreten, werden vermutlich auch bei der Konstruktion von Quantenrechnern auftreten. Es gibt allerdings noch keine so weit entwickelte Theorie der quantenmechanischen Kommunikation wie die der klassischen Kommunikation. Ein wichtiges Theorem von Holevo [H73] besagt, daß in n quantenmechanischen Bits, sogennanten Qubits, nicht mehr als n klassische Bits Information enthalten sein können. In einer einfachen informationstheoretischen Hinsicht ist quantenmechanische Kommunikation also nicht außergewöhnlich mächtig. Aber eine solche Aussage ersetzt ebensowenig eine Theorie der Quantenkommunikation, wie die Shannonsche Informationstheorie die klassische Kommunikationstheorie enthält.
Diese Arbeit besteht aus drei Kapiteln, die sich mit probabilistischer, (beschränkt) nichtdeterministischer und quantenmechanischer Kommunikation befassen. In jedem Kapitel gibt es einen Teil, der strukturelle Aussagen zur

Kommunikationskomplexität enthält, meistens zur Thematik beschränkter Interaktion. In einem weiteren Teil werden verschiedene Anwendungen oder verwandte Resultate für andere Berechnungsmodelle beschrieben.

1.2 Resultate und Aufbau der Arbeit

Abschnitt 2.1 gibt die wesentlichen Definitionen der betrachteten Berechnungsmodelle, sowie grundlegende Resultate, welche im Verlauf der Arbeit wichtig sind. Abschnitt 2.2 beschreibt den notwendigen Hintergrund aus der Wahrscheinlichkeitstheorie, 2.3 einige Resultate aus der Informationstheorie. Abschnitt 2.4 definiert die Vapnik Chervonenkis Dimension (VC-Dimension, siehe [VC71]), eine kombinatorische Größe, welche sehr nützlich im Zusammenhang mit Einweg Kommunikation ist.
Abschnitt 3.1 gibt einen Überblick über Kapitel 3. Abschnitt 3.2 erklärt einige Resultate über probabilistische Einweg Kommunikation und untersucht die probabilistische Komplexität der „Pointer Jumping“ Funktion. Gegenüber vorigen Arbeiten verbesserte untere und obere Schranken werden angegeben. Auch wird eine asymptotisch schlechtere untere Schranke angegeben, die aber für kleine Rundenzahlen wegen besserer Konstanten interessant ist. Die Abschnitte 3.3.1 und 3.3.2 studieren die Größenkomplexität probabilistischer Automaten, Abschnitt 3.3.3 erforscht die Größe probabilistischer Boolescher Formeln.
Abschnitt 4.1 gibt einen Überblick über Kapitel 4. Abschnitt 4.2 untersucht nichtdeterministische Kommunikation mit beschränktem Nichtdeterminismus und beschränkter Interaktion. 4.3 beschreibt Anwendungen und Resultate über nichtdeterministische Automaten, Formeln und die Tiefe nichtdeterministischer monotoner Schaltkreise.
Abschnitt 5.1 gibt einen Überblick über Kapitel 5 und 5.2 gibt eine kurze Einführung in die benötigten Begriffe aus der Quantenmechanik, sowie einige Resultate der Quanteninformationstheorie. 5.3 führt die betrachteten Modelle von Quantenrechnern ein, 5.4 ist der Quantenkommunikation gewidmet, 5.5 den Anwendungen auf Quanten Automaten und Formeln.
Die Hauptresultate der Arbeit sind die folgenden:

1. Eine verbesserte Analyse der Komplexität des Pointer Jumping Problems (Theoreme 3.1, 3.2, 3.3).

2. Eine Methode zum Beweis unterer Schranken für probabilistische und quantenmechanische Formeln (Korollar 3.3 und Theorem 5.13).

3. Beweis eines polynomiellen Unterschiedes zwischen den Längen von (sogar quantenmechanischen) Las Vegas Formeln und von (klassischen) Monte Carlo Formeln (Korollare 3.6 und 5.9).

4. Ein polynomieller Unterschied zwischen den Längen von quantenmechanischen Formeln mit und ohne mehrfach lesbare Zufallseingaben (Korollar 5.6).

5. Eine Rundenhierarchie für beschränkt nichtdeterministische Kommunikationskomplexität (Theorem 4.2).

6. Reversalhierarchien für nichtdeterministische Zweiweg Automaten (Theoreme 4.4, 4.5).

7. Eine polynomielle untere Schranke für die Länge beschränkt nichtdeterministischer Boolescher Formeln (Theorem 4.6).

8. Tiefenhierarchien für beschränkt nichtdeterministische monotone Schaltkreise (Theoreme 4.8, 4.9, 4.10).

9. Entwurf eines programmierbaren Quanten Gatters mit beliebig hoher Erfolgswahrscheinlichkeit (Theorem 5.2).

10. Ein polynomieller Unterschied zwischen quantenmechanischer Las Vegas Kommunikation und klassischer probabilistischer Kommunikation für eine totale Funktion (Theorem 5.5).

11. Untere Schranken für quantenmechanische Einweg Kommunikation (Theoreme 5.7, 5.8, 5.9, 5.10).

12. Untere Schranken für quantenmechanische Automaten (Theorem 5.11).

Ein Großteil der Resultate wurde bereits in den folgenden Arbeiten veröffentlicht: [Kl97], [Kl98], [Kl00a], [Kl00b], [HKKSS00].
Resultate anderer Autoren sind generell als „Fakt“ angegeben, manchmal werden solche Resultate der Vollständigkeit halber auch bewiesen.

Kapitel 2

Definitionen und Grundlagen

In diesem Kapitel wird der zum Verständnis der Kapitel über klassische Kommunikation notwendige Hintergrund beschrieben. Grundlagen zum Thema Quantenrechner werden in Kapitel 5 gegeben.
Zu Beginn einige Notationen: $\mathcal{P}(S)$ sei die Potenzmenge einer Menge S. $\mathcal{P}(m,n)$ sei die Menge der Teilmengen der Größe n aus dem Universum $\{1,\ldots,m\}$. Logarithmen sind stets zur Basis 2 gemeint, außer eine andere Basis ist explizit angegeben. Weiterhin bezeichne $\log^{(1)} n = \log n$ und $\log^{(k)} n = \log(\log^{(k-1)} n)$, sowie $\log^* n = \min\{k | \log^{(k)} n \leq 1\}$.

2.1 Die Berechnungsmodelle

2.1.1 Das Kommunikationsmodell

Das wichtigste Berechnungsmodell für diese Arbeit ist das zwei Spieler Kommunikationsmodell von Yao [Y79]. Unreferenzierte Fakten in diesem Abschnitt sind in [Hr97] oder [KN97] zu finden. Alice und Bob heißen von jetzt ab A und B.

Definition 2.1 *Es sei $f : X_A \times X_B \to Y$ eine Funktion über abzählbaren Mengen X_A, X_B, Y. Zwei Spieler A und B mit unbeschränkter Berechnungskraft erhalten Eingaben x_A, x_B aus X_A und X_B. Ihr Ziel ist es, mit Hilfe eines Kommunikationsprotokolls den Wert $f(x_A, x_B)$ zu berechnen.*
Hierzu tauschen die Spieler binär kodierte Nachrichten aus. Ein Spieler beginnt, indem er die erste Nachricht sendet, der andere Spieler antwortet usw. Wir nehmen an, daß keine zwei zu einem Zeitpunkt möglichen Nachrichten die Eigenschaft haben, daß eine ein echter Präfix der anderen ist, und somit auf ein spezielles Ende-Symbol verzichtet werden kann. A beginnt, berechnet ihre erste Nachricht aus der Eingabe, sendet die Nachricht zu B. B berechnet seine nächste Nachricht aus der erhaltenen Nachricht und seiner Eingabe, sendet seine neue Nachricht usw. Dies geht so lange, bis ein

Spieler die Ausgabe kennt. Dieser Spieler sendet dann die Ausgabe zu dem anderen Spieler. Alle Berechnungen sind deterministisch.
Die Kommunikationskomplexität eines solchen Protokolls ist die maximale Anzahl ausgetauschter Bits über alle Eingaben, wobei die letzte Nachricht mit der Ausgabe nicht mitgerechnet wird. Ist die Kommunikationskomplexität eines Protokolls nicht endlich, so wird sie als unendlich (∞) definiert. Die deterministische Kommunikationskomplexität $D(f)$ einer Funktion f ist die Komplexität eines optimalen Protokolls für f.
Die Kommunikationsmatrix einer Funktion f ist die Matrix M mit der Eigenschaft $M(x_A, x_B) = f(x_A, x_B)$ für alle $x_A \in X_A, x_B \in X_B$.

Deterministische Protokolle induzieren auf natürliche Art und Weise einen sogenannten Protokollbaum. Die Wurzel ist indiziert mit der Kommunikationsmatrix. Knoten entsprechen Situationen im Protokoll. Jeder Knoten hat so viele Kinder, wie es Möglichkeiten für die nächste Nachricht im Protokoll gibt. Für jedes Kind gibt es eine reduzierte Kommunikationsmatrix, welche die mit den Nachrichten auf dem Weg von der Wurzel konsistenten Eingaben enthält. Da ein Spieler jeweils nur weiß, welche Zeile oder welche Spalte der Gesamteingabe entspricht, wird jeweils die Zeilen- oder Spaltenmenge eingeschränkt. Die Blätter des Baums sind mit monochromatischen Matrizen assoziiert, da nun beide Spieler die Ausgabe kennen, und somit der Funktionswert feststeht.
Das Hauptthema der Arbeit ist die Abhängigkeit der Kommunikationskomplexität von der Anzahl der Kommunikationsrunden.

Definition 2.2 *Ein Protokoll benutzt k Runden auf einer Eingabe, wenn die Spieler auf dieser Eingabe k Nachrichten austauschen, wobei der Sender der Nachricht jeweils wechselt, und in einer $k+1$ten Nachricht die Ausgabe bekanntgegeben wird. Ein Protokoll hat k Runden, wenn für alle Eingaben höchstens k Runden benutzt werden. Die Komplexität von Funktionen, wenn das Kommunikationsmodell auf Protokolle mit maximal k Runden eingeschränkt ist, wird durch Superskripte der Form $D^{(k)}$ notiert. Hierbei startet Spieler A. Wenn Spieler B startet, wird die Notation $D^{(B,k)}$ benutzt. Im Fall $k = 1$ wird auch der Ausdruck Einweg Kommunikationskomplexität verwendet.*

Eine weitere mögliche Einschränkung der Interaktion besteht darin, verschiedene Schranken der erlaubten Kommunikation für Spieler A und B festzulegen. Dieser Ansatz wird in [MNSW95] verfolgt und führt zu Anwendungen auf Datenstrukturprobleme. In dieser Arbeit wird der Ansatz jedoch nicht weiter betrachtet.
Verschiedene weitere Akzeptanzmodi (außer Determinismus) werden für Kommunikationsprotokolle betrachtet. Zunächst Nichtdeterminismus.

Definition 2.3 *In einem nichtdeterministischen Protokoll für eine Boolesche Funktion $f : X_A \times X_B \to \{0,1\}$ dürfen beide Spieler zunächst eine beliebig lange binäre Zeichenkette nichtdeterministisch raten. Abhängig von dieser Zeichenkette darf jeder Spieler eine beliebige deterministische Strategie wählen, nach der seine Nachrichten berechnet werden. Die Ratworte sind allerdings privat, d.h. nur dem ratenden Spieler zugänglich. Eine Eingabe wird akzeptiert, wenn es ein Ratewort gibt, bei dem das Protokoll akzeptiert. Ansonsten wird eine Eingabe verworfen.*
Im Modell des beschränkten Nichtdeterminismus ist die Länge der binären Rateworte durch einen Wert s begrenzt.
Die Komplexität eines nichtdeterministischen Protokolls wird als das Maximum der Kommunikation über alle akzeptierten Eingaben und alle verwendeten Rateworte gemessen.
Die nichtdeterministische Kommunikationskomplexität $N(f)$ ist die Komplexität eines optimalen nichtdeterministischen Protokolls für f mit privatem, unbeschränktem Nichtdeterminismus.
$N_s(f)$ bezeichnet die Komplexität eines optimalen nichtdeterministischen Protokolls für f, welches für alle Eingaben höchstens s private Ratebits verwendet.

Wir stellen zunächst fest, daß unbeschränkt nichtdeterministische Protokolle keine Interaktion benötigen.

Fakt 2.1 *$N^{(1)}(f) = N(f)$ für alle Funktionen $f : X_A \times X_B \to \{0,1\}$.*

Beweis: Es sei ein beliebiges optimales nichtdeterministisches Protokoll für f gegeben. Spieler A rät die gesamte auf der Eingabe x_A, x_B stattfindende Kommunikation und prüft, ob diese Kommunikation konsistent mit einer Berechnung ist, die A auf einem Ratewort ausführen würde. Wenn ja, so schickt sie das Ratewort zu B, der ebenfalls Konsistenz prüft. B akzeptiert, wenn auch seine Kommunikation im geratenen Dialog konsistent mit seiner Eingabe ist und das Protokoll mit der geratenen Kommunikation akzeptieren würde. Ansonsten verwirft B. Hat A keine aus ihrer Sicht konsistente Nachrichtenfolge geraten, so rät sie erneut, bis dies der Fall ist. Die Kommunikationskomplexität des Protokolls bleibt offensichtlich unverändert, und es werden dieselben Eingaben akzeptiert. So erhält man ein Protokoll mit nur einer Runde. □
Die obige Tatsache erlaubt eine einfache kombinatorische Charakterisierung der nichtdeterministischen Kommunikationskomplexität. Eine Teilmatrix der Kommunikationsmatrix heiße 1-chromatisch, wenn sie nur Einsen enthält. Eine Überdeckung der Einsen der Kommunikationsmatrix ist eine Menge von 1-chromatischen Teilmatrizen, so daß jede Eins in mindestens einer derselben enthalten ist.

Fakt 2.2 *Ein nichtdeterministisches (Einweg) Protokoll für f, welches genau c verschiedene Nachrichten verwendet, existiert genau dann, wenn es eine Überdeckung der Kommunikationsmatrix mit c 1-chromatischen Teilmatrizen gibt (d.h. mit Teilmatrizen, welche nur Einsen enthalten).*

Während nichtdeterministische Kommunikation ein theoretisch motiviertes Modell ist, beschreibt probabilistische Kommunikation das abgesehen von quantenmechanischen Modellen mächtigste realistische Modell der Kommunikation.

Definition 2.4 *In einem probabilistischen Protokoll mit privaten Zufallsbits besitzen A und B je eine Quelle von unabhängigen Zufallsbits mit der Verteilung 1/2 für 1 und 1/2 für 0. Die Spieler dürfen jeweils auf ihre Zufallsquelle zugreifen und ihre Kommunikation von den gelesenen Bits abhängig machen. Wir unterscheiden folgende Akzeptanzmodi:*

1. *In einem Las Vegas Protokoll müssen die Spieler immer das korrekte Ergebnis ausgeben. Die Kosten eines Protokoll werden bestimmt als das Maximum (über alle Eingaben) der erwarteten Kommunikation des Protokolls, wobei der Erwartungswert über alle Zufallswahlen geht. Die Las Vegas Komplexität von f ist definiert als die Komplexität eines optimalen Las Vegas Protokolls für f und wird mit $R_0(f)$ bezeichnet.*

   Eine zweite Variante der Las Vegas Protokolle ist wie folgt definiert. Die Kosten eines Protokolls sind das Maximum der Kommunikation über alle Eingaben und alle Münzwürfe. Allerdings darf nun das Protokoll für jede Eingabe mit einer bestimmten Wahrscheinlichkeit ϵ aufgeben, ohne das Ergebnis zu liefern. Ansonsten wird wieder verlangt, daß alle Ausgaben korrekt sind. Die Komplexität von f als die Komplexität eines optimalen Protokolls dieses Typs wird mit $R_{0,\epsilon}(f)$ bezeichnet.

2. *In einem probabilistischen Protokoll mit beschränktem Fehler ϵ darf das Protokoll für jede Eingabe mit der Wahrscheinlichkeit höchstens ϵ eine falsche Ausgabe machen. Die Komplexität eines Protokolls ist als das Maximum der Kommunikation über alle Eingaben und alle Zufallswahlen definiert. Die probabilistische Komplexität von f bei beschränktem Fehler ist die Komplexität eines optimalen solchen Protokolls und wird mit $R_\epsilon(f)$ bezeichnet. Für $\epsilon = 1/3$ wird auch $R(f)$ geschrieben.*

3. *Ein Protokoll mit beschränktem Fehler heißt Monte Carlo Protokoll, wenn die Spieler eine Eingabe mit $f(x_A, x_B) = 0$ mit Sicherheit verwerfen, und ansonsten mit beschränktem Fehler arbeiten.*

Weiterhin untersuchen wir probabilistische Kommunikation mit öffentlichem Zufall. Hier haben beide Spieler Zugriff auf eine (beliebig lange) Liste von

Zufallsbits, ohne daß hierfür Kosten entstehen. Die Komplexität in diesem Modell wird mit R^{pub} bezeichnet, die verschiedenen Akzeptanzmodi sind wie oben definiert.

Untere Schranken für probabilistische Kommunikationskomplexität werden üblicherweise bewiesen, indem man untere Schranken für deterministische Protokolle zeigt, die auf fast allen Eingaben bezüglich einer Wahrscheinlichkeitsverteilung korrekt sind.

Definition 2.5 *Ein deterministisches Protokoll berechnet f mit Fehler ϵ unter der Verteilung μ, wenn die Wahrscheinlichkeit eines Fehlers des Protokolls unter dieser Verteilung durch ϵ beschränkt ist. Die Komplexität eines optimalen deterministischen Protokolls für f mit Fehler ϵ bezüglich μ wird mit $D^{\mu}_{\epsilon}(f)$ bezeichnet.*

Die folgende Konsequenz einer fundamentalen Beobachtung von Yao stellt eine Verbindung zwischen den oben definierten Maßen her und folgt aus dem Minimax Prinzip der Spieltheorie (siehe [Y77] und [KN97]).

Fakt 2.3 *Für alle $\epsilon > 0$, Verteilungen μ und alle Funktionen f:*
$R^{pub}_{\epsilon}(f) = \max_{\mu} D^{\mu}_{\epsilon}(f)$.
Für alle k: $R^{(k,pub)}_{\epsilon}(f) = \max_{\mu} D^{(k,\mu)}_{\epsilon}(f)$.

Vergleicht man optimale Protokolle mit öffentlichem Zufall, und solche mit privatem Zufall, so ergibt sich nur ein kleiner Unterschied [Ne91]:

Fakt 2.4 *Für alle $1 > \epsilon, \delta > 0$, alle $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, alle k:*
$R_{(1+\delta)\epsilon}(f) \leq R^{pub}_{\epsilon} + \log n + O(\log(1/(\epsilon\delta))$.
$R^{(k)}_{(1+\delta)\epsilon}(f) \leq R^{(k,pub)}_{\epsilon} + \log n + O(\log(1/(\epsilon\delta))$.
$R_{0,(1+\delta)\epsilon}(f) \leq R^{pub}_{0,\epsilon} + \log n + O(\log(1/(\epsilon\delta))$.
$R^{(k)}_{0,(1+\delta)\epsilon}(f) \leq R^{(k,pub)}_{0,\epsilon} + \log n + O(\log(1/(\epsilon\delta))$.

Durch (parallele) Wiederholung mit unabhängigen Zufallsexperimenten kann die Fehlerwahrscheinlichkeit probabilistischer Protokolle verringert werden, indem man die Mehrheitsausgabe bestimmt.

Fakt 2.5 *Für alle $1 > \epsilon > \delta > 0$ und für alle f, für alle k:*
$R_{\delta}(f) \leq R_{\epsilon}(f) \cdot O(\log_{\epsilon} \delta)$.
$R^{(k)}_{\delta}(f) \leq R^{(k)}_{\epsilon}(f) \cdot O(\log_{\epsilon} \delta)$.

Seit der Einführung des Modells der Kommunikationskomplexität sind viele Resultate zum Thema bewiesen worden, siehe etwa die Monographien [Hr97],[KN97]. Im folgenden stellen wir einige Resultate zusammen, welche für die Arbeit wichtig sind. Eine erste Frage ist, um wieviel die Kommunikation durch probabilistische oder nichtdeterministische Protokolle im Vergleich zu deterministischen Protokollen verringert werden kann. Standardargumente zeigen (siehe etwa [KN97] oder [Hr97]):

Fakt 2.6 *Für alle f:*
$D^{(1)}(f) \leq 2^{O(R(f))}$.
$D^{(1)}(f) \leq 2^{N(f)}$.

Fakt 2.7 *Für alle $f : X_A \times X_B \to \{0,1\}$:*
$D(f) \geq R_0(f) \geq R(f), N(f)$.
$D(f) = D(\neg f)$.
$R_\epsilon(f) = R_\epsilon(\neg f)$.
$R_0(f) = R_0(\neg f)$.

An dieser Stelle führen wir einige wichtige Kommunikationsprobleme ein, die in dieser Arbeit benötigt werden.

Definition 2.6 *Das Disjunktheitsproblem $DISJ_n(x_1 \dots x_n, y_1 \dots y_n) = 1 \iff \forall i : \neg x_i \vee \neg y_i$. Die Funktion akzeptiert eine Eingabe, wenn die von der Eingabe beschriebenen Mengen disjunkt sind.*
Das Gleichheitsproblem $EQ_n(x_1 \dots x_n, y_1 \dots y_n) = 1 \iff \forall i : x_i = y_i$. Die Funktion akzeptiert eine Eingabe, wenn $x = y$.
Das innere Produkt $IP_n(x_1 \dots x_n, y_1 \dots y_n) = 1 \iff \sum x_i y_i \equiv 0 \ mod\ 2$.
Die Index Funktion $IX_{2^n}(x_1 \dots x_{2^n}, y_1 \dots y_n) = 1 \iff x_y = 1$.
Die Größer-Gleich Funktion $GT_n(x_1 \dots x_n, y_1 \dots y_n) = 1 \iff$ die von x kodierte Zahl ist nicht kleiner als die von y kodierte Zahl.

Als ein Beispiel bestimmen wir die Kommunikationskomplexität des Gleichheitsproblems. Wir geben den Beweis an, weil er die wichtige Idee des Fingerabdrucks eines Objekts einführt. Um zwei Objekte zu vergleichen, bestimmt man jeweils eine kurze von ihnen abhängige Information, welche man miteinander vergleicht. Diese soll mit hoher Wahrscheinlichkeit eine Unterscheidung ermöglichen, siehe [MR95] für eine Diskussion des Ansatzes. Weiterhin werden wir sehen, daß für dieses fundamentale Problem keine Interaktion notwendig ist.

Fakt 2.8 $R_\epsilon^{(1,pub)}(EQ_n) = O(\log(1/\epsilon))$.
$R_\epsilon^{(1)}(EQ_n) \leq \log n + O(\log(1/\epsilon))$.
$R(EQ_n) = \Theta(\log n)$.
$N(EQ_n) = n$.
$N(\neg EQ_n) = \Theta(\log n)$.

Beweis: Zum Beweis der ersten Behauptung betrachte man das folgende Protokoll. Die Spieler raten $\log(1/\epsilon)$ öffentliche binäre Zufallsworte der Länge n. Für jedes Wort $r^{(j)}$ sendet Spieler A das Bit $\bigoplus_{i=1}^n x_i \wedge r_i^{(j)}$. B berechnet $\bigoplus_{i=1}^n y_i \wedge r_i^{(j)}$, und akzeptiert, wenn beide Werte gleich sind. Die behauptete Kommunikationsschranke gilt also. Man sieht leicht ein, daß jeder

Test ungleiche Worte x, y mit Wahrscheinlichkeit $1/2$ verwirft. Die Wahrscheinlichkeit, fehlerhaft zu akzeptieren, ist also höchstens ϵ. Gleiche Worte werden offensichtlich immer mit Wahrscheinlichkeit 1 akzeptiert.
Die obere Schranke im Modell mit privatem Zufall folgt mit Fakt 2.4. Die untere Schranke für die probabilistische Kommunikationskomplexität ist eine direkte Konsequenz von Fakt 2.6 und der folgenden unteren Schranke für nichtdeterministische Kommunikation (und damit für deterministische Kommunikation).
Diese wird wie folgt bewiesen: Jedes nichtdeterministische Protokoll mit Kommunikation c induziert eine Überdeckung der Einsen der Kommunikationsmatrix mit höchstens 2^c 1-chromatischen Teilmatrizen. Da die Einsen der Kommunikationsmatrix auf der Diagonalen liegen, kann eine solche Matrix nur einen Eintrag bedecken und es sind 2^n Teilmatrizen notwendig, also $n \leq c$. Andererseits reicht es, wenn A ihre gesamte Eingabe sendet, daher ist die nichtdeterministische Kommunikationskomplexität von EQ_n genau n.
Die untere Schranke für die nichtdeterministische Komplexität des Komplements von EQ_n folgt aus den Fakten 2.6 und 2.7. Für die obere Schranke betrachte man folgendes Protokoll: A rät eine Position i, an der x und y sich möglicherweise unterscheiden und sendet i sowie x_i. B akzeptiert, wenn $x_i \neq y_i$. Daher sind $\log n + O(1)$ Bits Kommunikation ausreichend. □
Das Komplement des Disjunktheitsproblems besitzt die nichtdeterministische Kommunikationskomplexität $O(\log n)$, d.h., $N(\neg DISJ_n) = O(\log n)$, aber seine probabilistische Komplexität ist hoch, wie in [KS92] und [R92] bewiesen wird.

Fakt 2.9 *$R(DISJ_n) = \Omega(n)$.*

Daher ergibt sich folgendes Korollar der Fakten 2.8 und 2.9, welches zeigt, daß Probabilismus und Nichtdeterminismus in der Kommunikationskomplexität nicht vergleichbar sind.

Korollar 2.1 *Es gibt eine totale Funktion mit $R(f) = \Omega(n)$, aber $N(f) = O(\log n)$. Es gibt eine totale Funktion mit $N(f) = n$, aber $R(f) = O(\log n)$. Diese Unterschiede sind wegen Fakt 2.6 maximal.*

Während Nichtdeterminismus verglichen mit Determinismus eine exponentielle Einsparung in der Kommunikation ermöglicht, zeigt das folgende fundamentale Resultat von [AUY83] (siehe auch [HR93]), daß eine solch drastische Einsparung nicht zugleich für eine Funktion f und ihr Komplement möglich ist, wenn f eine totale Funktion ist.

Fakt 2.10 *Für alle totalen Booleschen Funktionen f:*
$D(f) \leq (1 + o(1))(N(f) \cdot N(\neg f))$.
$D(f) \leq (1 + o(1))R_0^2(f)$.

Es ist wichtig zu bemerken, daß die obige Simulation im allgemeinen deterministische Protokolle mit einer großen Anzahl von Runden ergibt. Fürer [Fü87] hat gezeigt, daß der obige maximale Unterschied auch erreicht wird.

Fakt 2.11 *Es gibt ein totale Boolesche Funktion so daß* $R_0(f) = \Theta(\sqrt{n})$, *aber* $D(f) = \Theta(n)$.

Eine verallgemeinerte Fassung von Fakt 2.10 wird in [L90] bewiesen. Es sei $trank(M(f))$ die maximale Anzahl von Zeilen in einer Teilmatrix der Kommunikationsmatrix von f, in der sich oberhalb einer Diagonale mit Einsen nur Nullen befinden.

Fakt 2.12 *Für alle totalen Booleschen Funktionen* f*:*
$D(f) \leq (1 + o(1))(N(\neg f) \cdot trank(f))$.

Las Vegas Probabilismus kann also eine quadratische Beschleunigung ergeben, Probabilismus mit beschränktem Fehler eine exponentielle Beschleunigung im Vergleich zu Determinismus. An dieser Stelle ist ein weiterer Vergleich möglich: gibt es eine totale Funktion, so daß probabilistische Protokolle quadratische teurer sind als nichtdeterministische Protokolle für die Funktion und für ihr Komplement (und daher so teuer wir deterministische Protokolle schlimmstenfalls)? Eine Antwort wurde von Beame und Lawry [BL92] gegeben:

Fakt 2.13 *Es gibt eine totale Funktion* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *mit* $R_0(f) = \Omega(\log^2 n)$ *und* $N(f), N(\neg f) = O(\log n)$.

Diese Aussage kann noch in gewisser Hinsicht verbessert werden. Zuerst wenden wir uns jedoch der nichtdeterministischen Kommunikation mit beschränktem Nichtdeterminismus zu. Man betrachte die folgende iterierte Version des (komplementierten) Disjunktheitsproblems.

Definition 2.7 *Sei* $D_{n,s}$ *die folgende Boolesche Funktion:*

$$D_{n,s}(x_1 x_2 \dots x_s, y_1 y_2 \dots y_s) = 1 \iff \forall i : x_i, y_i \in \mathcal{P}(n^{32}, n) \text{ und } x_i \cap y_i \neq \emptyset$$

Man beachte, daß die Eingabelänge $\Theta(ns \log n)$ ist und daß offensichtlich nichtdeterministische Protokolle $D_{n,s}$ bzw. das Komplement von $D_{n,s}$ mit Kommunikation $O(s \log n)$ bzw. $O(n \log n)$ berechnen können. Hromkovič und Schnitger [HrS96] haben die Frage betrachtet, wie effizient ein nichtdeterministisches Protokoll für $D_{n,s}$ sein kann, wenn es nur s nichtdeterministische Bits verwendet.

Fakt 2.14 *1. Es gibt ein* $\epsilon > 0$ *so daß jedes deterministische Protokoll, welches* $D_{n,s}$ *korrekt auf einem Bruchteil* $1/2^{\epsilon s}$ *der Einsen berechnet, ohne eine Null zu akzeptieren, mindestens Kommunikation* $\Omega(ns)$ *benötigt.*

2. *Es gibt ein $\epsilon > 0$ so daß jedes nichtdeterministische Protokoll für $D_{n,s}$, welches höchstens ϵs nichtdeterministische Bits verwendet, Kommunikation $\Omega(ns)$ benötigt.*

Jedes Monte Carlo Protokoll (und jedes Las Vegas Protokoll) für eine Boolesche Funktion hat die folgende Eigenschaft: für eine beliebige Verteilung auf den zu akzeptierenden Eingaben gibt es ein deterministisches Protokoll, welches die Funktion korrekt auf der Hälfte der zu akzeptierenden Eingaben (gemäß der Verteilung) berechnet, ohne eine einzige zu verwerfende Eingabe zu akzeptieren. Das folgt aus dem Minimax Prinzip (Fakt 2.3). Somit gilt Teil 1 des obigen Resultates auch für Monte Carlo Protokolle. Das Resultat kann aber sogar für Protokolle mit beidseitigem Fehler gezeigt werden.

Lemma 2.1 *Es gibt ein $\epsilon > 0$ so daß jedes probabilistische Protokoll, welches mit zweiseitigem beschränktem Fehler eine Teilmenge der Größe $1/2^{\epsilon s}$ der Einsen gegen die Menge aller Nullen von $D_{n,s}$ entscheidet, Kommunikation $\Omega(ns)$ benötigt.*

Skizze des Beweises: Der Beweis folgt mit einigen Modifikationen des Beweises für Fakt 2.14 in [HrS96]. Wir beschreiben nur die notwendigen Modifikationen, lassen aber alle Details wegen der Länge des Beweises aus. In der Herleitung ihres Theorems 2.1 argumentieren die Autoren, daß ein deterministisches Protokoll, welches einen großen Anteil der Einsen akzeptiert, aber keine Null, viele 1-chromatische Teilmatrizen in der Kommunikationsmatrix als Nachrichten verwenden muß. Wir verwenden stattdessen fast 1-chromatische Teilmatrizen, d.h. Teilmatrizen mit $1 - \delta$ Einsen gemäß der Gleichverteilung. Dann kann das Lemma 2.1 von [HrS96] approximativ bewiesen werden. Ein probabilistisches Protokoll mit kleinem zweiseitigem Fehler führt zu einem deterministischen Protokoll mit kleinem Fehler bezüglich der Gleichverteilung, und damit zu einer Zerlegung der akzeptierten Eingaben in eine Menge fast 1-chromatischer Matrizen (die wenigen von nicht fast 1-chromatischen Matrizen akzeptierten Eingaben können ohne wesentlichen Verlust weggelassen werden). Der Beweis von Lemma 2.4 von [HrS96] kann dann ohne wesentliche Änderungen übernommen werden und das obige Lemma folgt. □
Lemma 2.1 impliziert

Theorem 2.1 $N(D_{n,n}), N(\neg D_{n,n}) \leq O(\sqrt{m \log m})$, *aber*
$R(D_{n,n}) = \Omega(m/\log m)$ *für die Eingabelänge* $m = \Theta(n^2 \log n)$.

2.1.2 Endliche Automaten

In diesem Abschnitt definieren wir die in der Arbeit betrachteten Modelle endlicher Automaten.

Endliche Automaten sind ein Beispiel eines Berechnungsmodells, das in seiner Kommunikationsfähigkeit eingeschränkt ist: es muß Information von der linken Seite zur rechten Seite der Eingabe transportiert werden, und dazu steht nur der im Zustand vorhandene endliche „Speicherplatz“ zur Verfügung. Eine hohe Kommunikationskomplexität für eine Zerlegung der Sprache in eine zweistellige Funktion bedingt dann eine große Zustandszahl. Wir geben jetzt die üblichen Definitionen für endliche Einweg Automaten.

Definition 2.8 *Ein deterministischer (endlicher) Einweg Automat, kurz dfa, ist ein Tupel $(Q, \Sigma, \delta, q_0, F)$, wobei Q die endliche Menge der Zustände ist, Σ ein endliches Alphabet, $\delta : Q \times \Sigma \to Q$ die Transitionsfunktion, $q_0 \in Q$ der Startzustand und $F \subseteq Q$ die Menge der akzeptierenden Zustände.*
Ein dfa startet seine Berechnung auf einem Wort $x \in \Sigma^*$ im Zustand q_0. Dann wird die Transition $\widetilde{\delta} : Q \times \Sigma^* \to Q$ angewendet, die wie folgt definiert ist: $\widetilde{\delta}(q, \epsilon) = q$ für das leere Wort ϵ, und $\widetilde{\delta}(q, xa) = \delta(\widetilde{\delta}(q, x), a)$ für jeden Buchstaben $a \in \Sigma$, alle Worte x, und alle Zustände q. Der dfa akzeptiert ein Wort x, wenn $\widetilde{\delta}(q_0, x) \in F$. Die von einem dfa A erkannte Sprache ist die Menge der akzeptierten Worte. Die Größe von A ist die Größe von Q. Sprachen, die von einem dfa erkannt werden können, heißen regulär.
Ein nichtdeterministischer (endlicher) Einweg Automat, kurz nfa, ist ähnlich wie ein dfa durch ein Tupel $(Q, \Sigma, \delta, q_0, F)$ definiert, wobei Q die endliche Menge der Zustände ist, Σ ein endliches Alphabet, $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ die Transitionsfunktion, $q_0 \in Q$ der Startzustand und $F \subseteq Q$ die Menge der akzeptierenden Zustände ist. Man beachte, daß δ diesmal eine Abbildung auf die Potenzmenge von Q ist.
Wieder kann δ zu einer Abbildung auf Worten expandiert werden, so daß $\widetilde{\delta}(q, x)$ die Menge der Zustände bezeichnet, die beim Lesen von x ab Zustand q erreichbar sind.
Ein nfa A akzeptiert ein Wort x, wenn $\widetilde{\delta}(q_0, x) \cap F \neq \emptyset$. Die von A erkannte Sprache ist die Menge der von A akzeptierten Worte. Die Größe von A ist die Größe von Q.
Ein probabilistischer (endlicher) Einweg Automat, kurz pfa, ist ähnlich wie ein dfa als ein Tupel $(Q, \Sigma, \delta, q_0, F)$ definiert, wobei Q die endliche Menge der Zustände ist, Σ ein endliches Alphabet, $\delta : Q \times \Sigma \times Q \to [0, 1]$ die Transitionsfunktion. Diese hat die Eigenschaft, daß für alle $a \in \Sigma, q \in Q$ gilt:

$$\sum_{q' \in Q} \delta(q, a, q') = 1,$$

d.h. auf den Nachfolgezuständen ist eine Wahrscheinlichkeitsverteilung definiert. $q_0 \in Q$ ist der Startzustand und $F \subseteq Q$ ist die Menge der akzeptierenden Zustände. Die Größe eines pfa ist die Größe von Q.
Wieder kann δ zu einer Abbildung auf Worten expandiert werden, so daß $\widetilde{\delta}(q, x, q')$ die Wahrscheinlichkeit ist, daß der pfa von Zustand q nach q' gelangt, wenn x gelesen wird.

Wir betrachten zwei verschiedene Akzeptanzmodi.
Bei einem pfa mit beschränktem Fehler verlangen wir, daß für alle Worte x entweder $\sum_{q\in F}\widetilde{\delta}(q_0,x,q)\geq 2/3$, oder $\sum_{q\in Q-F}\widetilde{\delta}(q_0,x,q)\geq 2/3$ gilt. Im ersten Fall wird x akzeptiert, im zweiten Fall verworfen. Die von einem pfa mit beschränktem Fehler erkannte Sprache ist die Menge der akzeptierten Worte.
Ein Las Vegas pfa ist ein Tupel $(Q,\Sigma,\delta,q_0,F_a,F_r)$, wobei die ersten fünf Komponenten wie für normale pfa definiert sind, aber es eine weitere Teilmenge $F_r\subseteq Q$ mit $F_a\cap F_r=\emptyset$ gibt. Es sei $F_?=Q-F_a-F_r$.
Alle Worte, welche mit Wahrscheinlichkeit größer als 0 einen Zustand in F_a erreichen, werden akzeptiert. Alle Worte, welche mit Wahrscheinlichkeit größer als 0 einen Zustand in F_r erreichen, werden verworfen. Der Automat muß die Eigenschaft haben, daß jedes Wort entweder akzeptiert oder verworfen wird, aber niemals beides. Außerdem muß für jedes Wort die Wahrscheinlichkeit, einen Zustand in $F_a\cup F_r$ zu erreichen, mindestens $1-\epsilon$ betragen, die sogenannte Erfolgswahrscheinlichkeit. Wenn nichts anderes vereinbart ist, gilt $\epsilon=1/2$.

Ein Las Vegas pfa klassifiziert also eine Eingabe entweder korrekt als zur Sprache gehörig oder nicht zur Sprache gehörig, oder gibt ohne Ergebnis auf, das letztere allerdings mit beschränkter Wahrscheinlichkeit.
Als nächstes fassen wir einige grundlegende Relationen zwischen den verschiedenen Akzeptanzmodi zusammen, mit dem Hauptaugenmerk auf den Komplexitätsparameter Größe, vergleiche [MF71]. Zunächst zum Vergleich der Sprachen, die überhaupt erkannt werden können, siehe [HU79], [R63].

Fakt 2.15 *Für alle regulären Sprachen gibt es einen dfa mit minimaler Zustandsanzahl, der aus einem gegebenen dfa in polynomieller Zeit konstruiert werden kann.*
Alle Sprachen, die von einem pfa mit beschränktem Fehler oder von einem nfa erkannt werden, sind regulär.
Die Größe eines minimalen dfa für eine Sprache, die von einem nfa mit s Zuständen erkannt werden kann, ist höchstens 2^s.
Die Größe eines minimalen dfa für eine Sprache, die von einem pfa mit s Zuständen und beschränktem Fehler erkannt werden kann, ist höchstens $s^{O(s)}$.

Die minimale Größe ist gleich dem Nerode Index der Sprache, siehe [HU79].
Das folgende Resultat wurde (im wesentlichen) in Arbeiten von Meyer und Fischer [MF71] und Ambainis [A96] bewiesen.

Fakt 2.16 *Es gibt eine Sprache L so daß ein minimaler pfa für L Größe s hat, und ein nfa mit Größe $O(\log s)$ für L existiert.*
Es gibt eine Sprache L so daß ein minimaler nfa für L die Größe s hat, und ein pfa mit Größe $O(\log s\cdot\frac{\log\log s}{\log\log\log s})$ und beschränktem Fehler für L existiert.

In [DHRS97] werden Einweg Las Vegas Automaten mit Erfolgswahrscheinlichkeit 1/2 untersucht.

Fakt 2.17 *Für jede Sprache, welche einen Las Vegas pfa mit s Zuständen besitzt, gibt es einen dfa mit $O(s^2)$ Zuständen.*
Es gibt eine Sprache L, so daß jeder dfa für L mindestens s Zustände hat, aber ein Las Vegas pfa für L mit $O(\sqrt{s})$ Zuständen existiert.

Für Arbeiten zum Thema des beschränkten Nichtdeterminismus bei Automaten sei auf [KW80], [GKW90], [HKKSS00] verwiesen. Zunächst definieren wir, wie der Verbrauch von Nichtdeterminismus quantifiziert wird.

Definition 2.9 *Sei A ein nfa und $x_1 \dots x_n$ ein Eingabewort. Ein Schritt der Berechnung von A auf x ist ein Tripel q, a, q', wobei $q' \in \delta(q, a)$. Die (nichtdeterministische) Ratekomplexität eines Schrittes (q, a, q') ist $\lceil \log |\delta(q, a)| \rceil$.*
Die Ratekomplexität einer Berechnung $q_{i_0}, q_{i_1}, \dots, q_{i_{n-1}}$, wobei $q_{i_0} = q_0$ und alle $(q_{i_j}, x_{j+1}, q_{i_{j+1}})$ Schritte sind, ist die Summe der Ratekomplexitäten der einzelnen Schritte.
Der Verbrauch an Nichtdeterminismus von A auf x ist das Maximum über alle Berechnungen von A auf x der Ratekomplexitäten dieser Berechnungen.
Der Verbrauch an Nichtdeterminismus von A auf Eingabelänge n ist das Maximum über alle Worte der Länge n des Verbrauchs von Nichtdeterminismus auf diesen Worten.
Der Verbrauch an Nichtdeterminismus von A ist dann eine Funktion von n.

Die obige Definition unterscheidet sich von der Definition [GKW90] insoweit, als hier über alle Berechnungen auf einem Wort maximiert wird, anstatt von minimiert. Der Grund für diese Abweichung liegt darin, daß sich so die Beziehung zu Kommunikationsprotokollen einfacher darstellen läßt. Tatsächlich ist diese Unterscheidung bei Kommunikationsprotokollen jedoch irrelevant, und alle unteren Schranken für Automaten in unserer Definition, welche über Kommunikation bewiesen werden, gelten auch für Automaten gemäß der anderen Definition. Die Definition mit Maximierung wurde auch in z.B. [SV81] verwendet.
Unsere Definition unterscheidet sich auch von der Definition in [HKKSS00], wo nicht die exakte Anzahl von Nachfolgern betrachtet wird, sondern nur, ob ein Schritt nichtdeterministisch ist oder nicht. Unsere Definition erlaubt also eine genauere Messung des Nichtdeterminismus'.

Lemma 2.2 *Der Verbrauch an Nichtdeterminismus eines nfa mit s Zuständen ist entweder höchstens $s \log s$ oder mindestens $(n - s)/s$ für unendlich viele n.*

Beweis: Wenn ein erreichbarer Zustand q eines nfa A zu einem Zyklus in A gehört, und q zwei ausgehende Kanten mit demselben Buchstaben hat, dann

gibt es ein Wort der Länge n auf dem mindestens $(n-s)/s \geq n/s - 1$ Bits geraten werden. Der Zustand q kann von Startzustand über ein Wort der Länge höchstens $s-1$ erreicht werden. Dann kann die Berechnung auf den restlichen $n-s+1$ Buchstaben immer in dem Kreis laufen, wobei mindestens alle s Zeichen ein nichtdeterministisches Bit geraten wird.
Wenn andererseits für alle Worte alle Zustände mit einer nichtdeterministischen Entscheidung nur einmal durchlaufen werden, können höchstens $s \log s$ Bits geraten werden. □
Die Aussage von Lemma 2.2 gilt nicht in der Definition von [GKW90], wo es möglich ist, sublinearen, aber superkonstanten Nichtdeterminismus zu haben.
In der Arbeit [GKW90] wird gezeigt, daß für jede reguläre Sprache L, welche den Buchstaben # nicht verwendet, die Sprache $(L\#)^*$ genauso effizient von einem dfa erkannt werden kann, wie von einem optimalen nfa mit konstantem Nichtdeterminismus. Nach unserer Definition wird für solche Sprachen also linearer Nichtdeterminismus für effizientere nfa benötigt.
Weiterhin wird in [GKW90] folgendes gezeigt.

Fakt 2.18 *Ein nfa mit s Zuständen, der beschränkten Nichtdeterminismus r hat, erkennt eine Sprache, deren minimaler dfa höchstens s^{2^r} Zustände hat.*
Es gibt eine Sprache L, welche einen minimalen dfa mit 2^n Zuständen hat, während ein nfa mit r Ratebits und Größe $O(2^{r+n/2^r})$ für L existiert und der minimale nfa für L nur $n+1$ Zustände hat.

Außer Einweg Automaten betrachten wir auch Zweiweg Automaten. Hier geben wir die Definition des Modells.

Definition 2.10 *Ein (endlicher) Zweiweg Automat M ist durch ein Tupel der Form $(Q, \Sigma, \delta, q_0, F)$ gegeben. Q ist die endliche Menge von Zuständen, Σ ein endliches Alphabet. Sei $\Gamma = \Sigma \cup \{\$\}$ das Bandalphabet. $\delta : Q \times \Gamma \times Q \times \{-1, 0, +1\} \to [0, 1]$ ist die Transitionsfunktion, $q_0 \in Q$ der Startzustand und $F \subseteq Q$ die Menge der akzeptierenden Zustände. Es muß gelten, daß für alle $q \in Q, a \in \Gamma$:*

$$\sum_{q' \in Q, b \in \{-1,0,+1\}} \delta(q, a, q', b) = 1.$$

Die Größe von M ist $|Q|$.

1. *M ist deterministisch, wenn $\delta(q, a, q', b) \in \{0, 1\}$ für alle q, q', a, b.*
2. *Wenn M nichtdeterministisch ist, so muß gelten $\delta(q, a, q', b) \in \{0, 1/c_{qa}\}$ (für ein $c_{qa} \geq 1$) für alle q, q', a, b.*
3. *im allgemeinen ist M probabilistisch, δ induziert eine Verteilung auf Zuständen und $\{-1, 0, +1\}$, gegeben q, a.*

M rechnet auf einem Wort $x_1 \dots x_n$ in der folgenden Art: M besitzt einen Lesekopf, der zu Beginn auf x_1 plaziert ist. Dann wählt M gegeben den Zustand q und das Zeichen x_i an der Kopfposition i den nächsten Zustand q' und die Bewegungsrichtung b des Kopfes gemäß der Verteilung, welche durch $\delta(q, x_i, \cdot, \cdot)$ gegeben ist. b bestimmt die Kopfbewegung nach links, stehenbleiben, oder nach rechts. Links und rechts vom Eingabewort befindet sich jeweils das Zeichen \$. Versucht der Automat, an diesem Zeichen vom Eingabewort weg weiterzulaufen, so bleibt der Kopf an seiner Position stehen.
Ein Wort wird von einer Berechnung akzeptiert, wenn irgendwann am linken oder rechten Ende der Eingabe das Zeichen \$ in einem akzeptierenden Zustand besucht wird.

1. *Ist M deterministisch, dann ist die Menge der mit Wahrscheinlichkeit 1 akzeptierten Worte die erkannte Sprache.*

2. *Ist M nichtdeterministisch, dann ist die von M erkannte Sprache die Menge der mit positiver Wahrscheinlichkeit akzeptierten Worte.*

3. *Ist M probabilistisch mit beschränktem Fehler, dann ist die von M erkannte Sprache die Menge der Worte, welche mit Wahrscheinlichkeit $2/3$ akzeptiert werden. Jedes Wort im Komplement dieser Sprache muß mit Wahrscheinlichkeit mindestens $2/3$ nicht akzeptiert werden.*

Eine Kreuzungsfolge zwischen den Eingabebuchstaben x_i und x_{i+1} ist die Folge der Zustände, in denen in einer Berechnung von x_i nach x_{i+1} gewechselt wird oder umgekehrt. Ein probabilistischer oder deterministischer Zweiweg Automat ist k-kreuzungsbeschränkt, wenn in jeder seiner Berechnungen jede Kreuzungsfolge höchstens k Elemente hat. Ein nichtdeterministischer Zweiweg Automat ist k-kreuzungsbeschränkt, wenn folgendes gilt: Versucht der Automat in einer Berechnung, eine längere Kreuzungsfolge als k zu erzeugen, so wird die Berechnung abgebrochen und die Eingabe verworfen.
Ein Zweiweg Automat führt in einer Berechnung k Reversals aus, wenn der Lesekopf k mal die Bewegungsrichtung ändert, wobei der Start der Berechnung mitgerechnet wird. Ein probabilistischer oder deterministischer Zweiweg Automat ist k-reversalbeschränkt, wenn er in allen seinen Berechnungen nur höchstens k Reversals ausführt. Ein nichtdeterministischer Zweiweg Automat ist k-reversalbeschränkt, wenn er in jeder Berechnung nach mehr als k Reversals nur noch verwerfen kann.
Eine Konfiguration des Automaten auf einer Eingabe bestehe aus dem Zustand q, einem Buchstaben x_i und einer Kopfposition i. Die nichtdeterministische Ratekomplexität (gemessen in Bits) eines nichtdeterministischen Zweiweg Automaten in einer Konfiguration ist der aufgerundete Logarithmus der Kardinalität der Menge der Nachfolgekonfigurationen der Konfiguration. Der Automat rät s Bits in einer Berechnung, wenn die Summe der Ratekomplexitäten der Konfigurationen in den Berechnungsschritten s ist.

Ein nichtdeterministischer Zweiweg Automat hat beschränkten Nichtdeterminismus $s(n)$, wenn für alle Eingaben der Länge n und alle Berechnungen auf den Eingaben höchstens $s(n)$ Bits geraten werden.

Offensichtlich sind k-kreuzungsbeschränkte Zweiweg Automaten mindestens so mächtig wie k-reversalbeschränkte Automaten. Jede Berechnung, die k-reversalbeschränkt ist, ist auch k-kreuzungsbeschränkt. Für Ergebnisse zur Reversalkomplexität siehe [CY91].
Die Beziehungen zwischen den Größen deterministischer und nichtdeterministischer Zweiweg Automaten sind ein wichtiges offenes Problem der Automatentheorie. Sipser [S80] hat gezeigt, daß für den Spezialfall sogenannter Sweepingautomaten ein exponentieller Größenunterschied zwischen deterministischen und nichtdeterministischen Automaten für eine Sprache besteht. Andererseits ist bekannt, daß nichtdeterministische Zweiweg Automaten nur reguläre Sprachen erkennen können (siehe [HU79]), während probabilistische Zweiweg Automaten mit beschränktem Fehler auch nichtreguläre Sprachen erkennen können, allerdings nur in exponentieller Zeit (siehe [Fr81],[DS89]). Las Vegas Zweiweg Automaten werden in [HrS99] untersucht.

2.1.3 Schaltkreise und Formeln

In diesem Abschnitt definieren wir die Modelle der Booleschen Schaltkreise und Formeln, welche in den Kapiteln 3 und 4 betrachtet werden. Fragen der Uniformität von Familien solcher Schaltkreise werden nicht betrachtet.

Definition 2.11 *Die Architektur eines Booleschen Schaltkreises ist ein gerichteter azyklischer Graph. Die Knoten mit Ingrad 0 sind entweder Eingaben, in welchem Fall sie mit einer Eingabevariablen x_i markiert sind, oder Konstanten, in welchem Fall sie mit 0 oder 1 markiert sind. Knoten mit Ingrad $k \geq 1$ sind Gatter und tragen als Beschriftung eine k-stellige Boolesche Funktion $g : \{0,1\}^k \to \{0,1\}$. Der Ingrad eines Knotens heißt auch fan-in, der Ausgrad auch fan-out. Als Ausgabegatter wird ein bestimmtes Gatter festgelegt.*
Die Größe eines Booleschen Schaltkreises ist die Anzahl seiner Knoten, die Tiefe die Länge eines längsten Pfades im Schaltkreis von einer Eingabe zu einer Ausgabe.
Ein Boolescher Schaltkreis rechnet auf einer Zuweisung von Werten zu den Variablen $x_1, \ldots, x_n$ wie folgt. Bei der Berechnung wird allen Knoten sukzessive ein Wert aus $\{0,1\}$ zugewiesen. Eingaben und Konstanten erhalten den entsprechenden Wert. Ein mit einer Funktion g beschrifteter innerer Knoten, der Vorgänger mit bereits berechneten Werten $v_1, \ldots, v_k$ hat, bekommt den Wert $g(v_1, \ldots, v_k)$ zugewiesen. Der Wert des Ausgabegatters ist die Ausgabe des Schaltkreises und demgemäß berechnet der Schaltkreis eine Funktion $f : \{0,1\}^n \to \{0,1\}$. Wenn nichts anderes gesagt wird, haben alle Schaltkreise in dieser Arbeit nur Gatter mit fan-in höchstens 2.

Während es wohlbekannt ist, daß fast alle Booleschen $f : \{0,1\}^n \to \{0,1\}$ nur Schaltkreise der Größe $\Theta(2^n/n)$ haben (siehe z.B. [BS90]), sind nichtlineare untere Schranken für explizit definierte Funktionen bisher nicht bekannt. Deshalb werden eingeschränkte Modelle von Schaltkreisen betrachtet.

Definition 2.12 *Ein monotoner Schaltkreis ist ein Boolescher Schaltkreis, der ausschließlich die Gattertypen UND ($\wedge$) und ODER ($\vee$) verwendet. Wir betrachten fan-in 2 und, wenn besonders vermerkt, unbeschränkten fan-in.*

Hohe untere Schranken für die Größe und die Tiefe monotoner Schaltkreise sind in mehreren Arbeiten enthalten, siehe [BS90] für einen Überblick. In unserem Kontext besonders interessant ist die Methode von Karchmer und Wigderson [KW90], welche einen Zusammenhang zwischen Runden und Kommunikation pro Runde in einem Kommunikationsspiel zu Tiefe und Größe von Booleschen Schaltkreisen herstellt, siehe auch [NW93]. Eine Hierarchie für monotone Schaltkreise mit polynomieller Größe und polylogarithmischer Tiefe wird von Raz und McKenzie [RM97] bewiesen.
Probabilistische monotone Schaltkreise werden in dieser Arbeit nicht betrachtet, da der Probabilismus mit Hilfe der Nichtuniformität effizient eliminiert werden kann. Als nächstes definieren wie nichtdeterministische monotone Schaltkreise.

Definition 2.13 *Ein nichtdeterministischer monotoner Schaltkreis besitzt als zusätzliche nichtdeterministische Eingabevariablen $a_1, \ldots, a_s, \overline{a_1}, \ldots, \overline{a_s}$ und akzeptiert eine Eingabe x, wenn es eine Belegung a der zusätzlichen Variablen gibt, so daß die Eingabe x, a akzeptiert wird.*

Die Ratebits werden in obiger Definition negiert und unnegiert gegeben, ansonsten sind sie wirkungslos.
Die Größe Boolescher Formeln ist ein wichtiges Komplexitätsmaß. Formeln sind ein grundlegendes Berechnungsmodell und eine universelle Datenstruktur für Boolesche Funktionen. Der Logarithmus der Formellänge ist asymptotisch äquivalent zur Tiefe von Schaltkreisen mit konstantem fan-in. Andererseits sind Formeln eingeschränkte Schaltkreise, für deren Größe superlineare untere Schranken bekannt sind.

Definition 2.14 *Eine Boolesche Formel ist ein Boolescher Schaltkreis mit fan-in 2 und fan-out 1. Die Booleschen Eingabevariablen $x_1, \ldots, x_n$ dürfen beliebig oft gelesen werden. Die Gatterfunktionen sind beliebig, Konstanten 0,1 dürfen gelesen werden.*
Die Größe (oder Länge) einer deterministischen Formel ist die Anzahl der nichtkonstanten Blätter.

Es kann gezeigt werden, daß für Boolesche Funktionen der Logarithmus der optimalen Formellänge in einer linearen Beziehung zur optimalen Schaltkreistiefe steht (siehe [BS90, W87]).

Probabilistische Formeln wurden in [V84], [B85], [DZ97] mit der Absicht betrachtet, effiziente (deterministische) monotone Formeln für die Majoritätsfunktion probabilistisch zu konstruieren.
Das gewöhnlich betrachtete Modell einer probabilistischen Formel ist eine Wahrscheinlichkeitsverteilung auf deterministischen Formeln. Da aber Formeln auch als Datenstruktur interessant sind, führen wir ein kompakteres Modell ein. „Faire“ probabilistische Formeln sind Formeln, die auf Eingabevariablen und zusätzlichen Zufallsvariablen arbeiten. Das Modell, bei dem eine probabilistische Formel eine Verteilung auf deterministischen Formeln ist, nennen wir „starke“ probabilistische Formeln.

Definition 2.15 *Eine faire probabilistische Formel ist eine Boolesche Formel, die auf Eingabevariablen und zusätzlichen Zufallsvariablen $r_1, \ldots, r_m$ arbeitet, eine starke probabilistische Formel ist eine Wahrscheinlichkeitsverteilung D auf (deterministischen) Booleschen Formeln. Faire bzw. starke probabilistische Formeln F berechnen eine Boolesche Funktion f mit beschränktem Fehler, wenn*

$$\Pr[F(x) \neq f(x)] \leq 1/3.$$

Faire bzw. starke probabilistische Formeln F sind Monte Carlo Formeln für f (d.h. haben einseitigen Fehler), wenn

$$\Pr[F(x) = 0 | f(x) = 1] \leq 1/2 \text{ und } \Pr[F(x) = 1 | f(x) = 0] = 0.$$

Eine Las Vegas Formel besteht aus zwei Booleschen Formeln. Eine der Formeln gibt die Ausgabe, die andere (verifizierende) Formel zeigt durch ihre Akzeptanz an, ob die Berechnung korrekt ist. Beide arbeiten auf derselben Eingabe. Es gibt also vier verschiedene Ausgaben, von denen zwei als „?“ zu interpretieren sind (die verifizierende Formel verwirft), und die anderen als 0 bzw. 1. Eine Las Vegas Formel F berechnet f, wenn die Ausgaben 0 oder 1 immer korrekt sind und

$$\Pr[F(x) = ?] \leq 1/2.$$

Die Größe einer fairen probabilistischen Formel ist die Anzahl ihrer nichtkonstanten Blätter.
Die Größe einer starken probabilistischen Formel ist die erwartete Größe einer deterministischen Formel bezüglich der Verteilung D.

Es ist leicht zu sehen, daß man die Fehlerwahrscheinlichkeit probabilistischer Formeln auf beliebig kleine Konstanten drücken kann, wobei die Größe entsprechend um einen konstanten Faktor steigt. Deshalb werden wir in asymptotischen Aussagen auch beliebige Konstanten als Fehlerwahrscheinlichkeiten betrachten.

Eine starke probabilistische Formel kann man in eine deterministische Formel umwandeln. Bei Monte Carlo Formeln steigt dabei die Größe um einen Faktor von $O(n)$: man wählt $O(n)$ zufällige Formeln gemäß der Verteilung, und verbindet sie durch ein ODER Gatter. Mit einer Anwendung der Chernov Ungleichung folgt die Korrektheit, siehe Abschnitt 2.2.1. Starke Formeln mit beschränktem Fehler derandomisiert man, indem man ebenfalls $O(n)$ zufällige Formeln wählt, und sie durch eine approximative Majoritätsfunktion verbindet. Diese Funktion gibt auf n Booleschen Variablen die Ausgabe 1, wenn mindestens $2n/3$ der Variablen den Wert 1 haben, und gibt die Ausgabe 0, wenn höchstens $n/3$ der Variablen den Wert 1 haben. Die approximative Majoritätsfunktion kann von deterministischen Formeln der Länge $O(n^2)$ berechnet werden, siehe [V84], [B85]. Daher steigt so die Größe um den Faktor $O(n^2)$ an.
Es sei bemerkt, daß starke probabilistische Formeln auch sublineare Länge haben können, was bei fairen probabilistischen Formeln, die von allen Eingaben abhängen, nicht möglich ist. Wir werden aber in Korollar 3.5 zeigen, daß starke probabilistische Formeln sublineare Größe nur dann haben können, wenn sie partielle Funktionen berechnen oder nicht von allen Eingabe abhängen. Die approximative Majorität kann beispielsweise von starken probabilistischen Formeln der Länge 1 berechnet werden, indem man den Wert einer zufälligen Variable zur Ausgabe macht.

Definition 2.16 *Eine nichtdeterministische Formel mit s Ratebits ist eine Formel mit zusätzlichen Eingabevariablen $a_1, \ldots, a_s$. Die Formel akzeptiert eine Eingabe x, wenn es eine Belegung der zusätzlichen Variablen a gibt, so daß (a, x) akzeptiert wird.*
Die Größe einer nichtdeterministischen Formel ist die Anzahl ihrer nichtkonstanten Blätter.

Einige wohlbekannte Resultate geben untere Schranken für die Formellänge an. Die Methode von Nečiporuk ist dabei die Methode, welche für Formeln, bei denen alle zweistelligen Gatterfunktionen erlaubt sind, die höchsten unteren Schranken ergibt (siehe [N66, BS90]). Für andere Methoden zum Beweis unterer Schranken siehe z.B. [BS90] und [BNS92]; eine Charakterisierung der Formellänge für Formeln mit Gatterfunktionen UND, ODER und NICHT durch die Komplexität eines bestimmten Kommunikationsspiels wird in [KW90] angegeben, untere Schranken mit dieser Methode sind vor allem für monotone Formeln bekannt.
Die Nečiporuk Methode werden wir in Abschnitt 3.3.3 mit Hilfe der Einweg Kommunikationskomplexität definieren. Hier geben wir zunächst die Standarddefinition, siehe z.B. [W87].
Sei f eine Funktion auf den n Variablen aus $X = \{x_1, \ldots, x_n\}$. Für eine Teilmenge $S \subseteq X$ sei eine Subfunktion auf S eine von f durch Fixierung der Variablen in $X - S$ induzierte Funktion auf den Variablen in S. Die Menge der Subfunktionen auf S heiße die Menge der S-Subfunktionen von f.

Fakt 2.19 (Nečiporuk) *Sei f eine Boolesche Funktion auf n Variablen. Weiter sei $S_1, \ldots, S_k$ eine Partition der Eingabevariablen und s_i die Anzahl der S_i-Subfunktionen von f. Dann hat jede deterministische Formel für f eine Länge von mindestens*

$$(1/4) \sum_{i=1}^{k} \log s_i.$$

Es ist leicht zu sehen, daß die Nečiporuk Funktion $(1/4) \sum_{i=1}^{k} \log s_i$ niemals größer als $n^2/\log n$ ist [W87].

Definition 2.17 *Die Funktion „indirekte Speicher-Adressierung" ISA ist wie folgt definiert: es gibt drei Blöcke von Eingaben U, X, Y mit $|U| = \log n - \log\log n$, $|X| = |Y| = n$. U adressiert einen Block der Länge $\log n$ in X, welcher ein Bit in Y adressiert. Dieses Bit ist die Ausgabe, also $ISA(U, X, Y) = Y_{X_U}$.*

Die folgende Aussage wird z.B. in [W87] und [Z95] bewiesen.

Fakt 2.20 *Jede deterministische Formel für ISA hat die Länge $\Omega(n^2/\log n)$. Es gibt eine deterministische Formel für ISA mit der Länge $O(n^2/\log n)$.*

2.2 Weitere Grundlagen

2.2.1 Wahrscheinlichkeitstheorie

In diesem Abschnitt geben wir die benötigten Definitionen aus der Wahrscheinlichkeitstheorie. Das meiste Material kann beispielsweise in [MR95] gefunden werden.

Definition 2.18 *Eine σ-Algebra $(\Omega, \mathcal{F})$ besteht aus einem Raum Ω und einer Menge $\mathcal{F}$ von Teilmengen des Raums, so daß folgende Bedingungen gelten:*

1. $\emptyset \in \mathcal{F}$
2. $E \in \mathcal{F} \Rightarrow \overline{E} \in \mathcal{F}$
3. $E_1, E_2, \ldots \in \mathcal{F} \Rightarrow E_1 \cup E_2 \cup \ldots \in \mathcal{F}$.

Die Menge von Mengen $\mathcal{F}$ ist abgeschlossen unter Komplementierung, Vereinigung, und daher auch unter Schnittbildung. Elemente von $\mathcal{F}$ heißen auch Ereignisse.

Definition 2.19 *Für eine σ-Algebra $(\Omega, \mathcal{F})$ ist ein Wahrscheinlichkeitsmaß $\Pr : \mathcal{F} \to [0, 1]$ eine Funktion mit den folgenden Eigenschaften:*

1. $\Pr(\Omega) = 1$.

2. *Für paarweise disjunkte* $E_1, E_2, \ldots \in \mathcal{F}$*:* $\Pr(E_1 \cup E_2 \cup \ldots) = \sum_i \Pr(E_i)$.

Ein Wahrscheinlichkeitsraum ist ein Tripel $(\Omega, \mathcal{F}, \Pr)$*, wobei* $(\Omega, \mathcal{F})$ *eine* σ*-Algebra ist und* $\Pr$ *ein Wahrscheinlichkeitsmaß.*

Definition 2.20 *Für einen Wahrscheinlichkeitsraum* $(\Omega, \mathcal{F}, \Pr)$ *ist die bedingte Wahrscheinlichkeit von* $E_1 \in \mathcal{F}$ *gegeben* $E_2 \in \mathcal{F}$ *definiert durch*

$$\Pr(E_1|E_2) = \frac{\Pr(E_1 \cap E_2)}{\Pr(E_2)},$$

falls $\Pr(E_2) > 0$.

Fakt 2.21 (Bayes Regel) *Sei* $E_1, \ldots, E_k$ *eine Partition von* Ω*. Dann gilt für jedes Ereignis* E*:*

$$\Pr(E_i|E) = \frac{\Pr(E|E_i)\Pr(E_i)}{\sum_{j=1}^{k} \Pr(E|E_j)\Pr(E_j)}.$$

Definition 2.21 *Eine Menge von Ereignissen* $\{E_i | i \in I\}$ *ist unabhängig, wenn für alle Teilmengen* $S \subseteq I$ *gilt:*

$$\Pr(\bigcap_{i \in S} E_i) = \prod_{i \in S} \Pr(E_i).$$

Definition 2.22 *Eine (diskrete) Zufallsvariable ist eine Funktion* $X : \Omega \to S$*, wobei* $S \subset \mathbb{R}$ *abzählbar ist und für alle* $x \in S$ *gilt:* $\{\omega \in \Omega | X(\omega) = x\} \in \mathcal{F}$*. Das Argument der Zufallsvariablen wird meistens nicht mit aufgeführt. Üblicherweise wird zu einer Zufallsvariablen eine Menge* $E \subseteq S$ *mit dem Ereignis* $\{\omega \in \Omega | X(\omega) \in E\} \in \mathcal{F}$ *identifiziert und ebenfalls als Ereignis bezeichnet.*
Zufallsvariablen X, Y *mit Wertebereichen* S_X *bzw.* S_Y *sind unabhängig, wenn für alle* $x \in S_X, y \in S_Y$ *die Ereignisse* $X = x$ *und* $Y = y$ *unabhängig sind.*
Die Dichtefunktion (oder Verteilung) einer Zufallsvariable X *ist definiert durch* $p_X(x) = \Pr(X = x)$.
Sei X *eine Zufallsvariable. Der Erwartungswert von* X *ist* $E[X] = \sum_{x \in S_X} x \cdot p_X(x)$*, der bedingte Erwartungswert ist* $E[X|Y = y] = \sum_x x \cdot \Pr(X = x|Y = y)$*, und* $E[X|Y]$ *ist die Zufallsvariable* $f(Y)$ *mit* $f(y) = E[X|Y = y]$.
Die Varianz von X *ist* $Var[X] = E[X^2] - E[X]^2$.

Fakt 2.22 *Für alle Zufallsvariablen* X, Y*:* $E[X + Y] = E[X] + E[Y]$.
Für unabhängige Zufallsvariablen X, Y*:* $Var[X + Y] = Var[X] + Var[Y]$.

Als ein Beispiel betrachten wir unabhängige Zufallsvariablen $X_1, \ldots, X_n$, wobei die Zufallsvariable X_i die folgende Verteilung habe: $X_i = 1$ mit Wahrscheinlichkeit p, $X_i = 0$ mit Wahrscheinlichkeit $1 - p$ für ein $p \in [0, 1]$. Also liegt eine Bernoulli Verteilung vor. Sei $X = X_1 + \cdots + X_n$. X hat eine Binomialverteilung mit den Parametern p und n. Der Erwartungswert von X ist pn, die Varianz ist $np(1 - p)$.
Die Binomialverteilung entsteht bei einem Experiment, bei dem Bälle aus einem Topf gezogen werden, der pn schwarze und $(1-p)n$ weiße Bälle enthält, wobei die gezogenen Bälle jedesmal wieder zurückgelegt werden. Bei der hypergeometrischen Verteilung liegt dasselbe Experiment zugrunde, wenn gezogene Bälle nicht zurückgelegt werden.
Es seien also M schwarze und $N-M$ weiße Bälle gegeben, und wir wählen n von den Bällen zufällig ohne Zurücklegen. Sei X die Zufallsvariable, welche die Anzahl gezogener schwarzer Bälle anzeigt. Dann ist der Erwartungswert von X gleich $E[X] = n\frac{M}{N}$. Die Varianz ist

$$Var[X] = n\frac{M}{N}(1 - \frac{M}{N})\frac{N-n}{N-1}.$$

Abweichungen von Erwartungswert können allgemein wie folgt beschränkt werden.

Fakt 2.23 (Markov) *Für alle nichtnegativen Zufallsvariablen X und alle positiven k:*
$\Pr(X \geq kE[X]) \leq 1/k$.

Bessere Schranken erhält man oft mit folgender Ungleichung.

Fakt 2.24 (Chebyshev) *Für alle Zufallsvariablen X und alle positiven k:*
$\Pr(|X - E[X]| \geq k\sqrt{Var[X]}) \leq 1/k^2$.

Für Summen von unabhängigen 0,1-wertigen Zufallsvariablen kann eine stärkere Ungleichung gezeigt werden.

Fakt 2.25 (Chernov) *Seien $X_1, \ldots, X_n$ unabhängige Zufallsvariablen, wobei $X_i = 1$ mit Wahrscheinlichkeit p_i und $X_i = 0$ mit Wahrscheinlichkeit $1 - p_i$ für ein $p_i \in [0, 1]$. Sei $X = \sum_i X_i$. Dann ist für $0 < \epsilon \leq 1$:*

$$\Pr(X \leq (1 - \epsilon)E[X]) \leq e^{-\epsilon^2 E[X]/2},$$

$$\Pr(X \geq (1 + \epsilon)E[X]) \leq e^{-\epsilon^2 E[X]/3}.$$

Sind Zufallsvariablen nicht unabhängig, ist es oft schwer, gute Schranken für Abweichungen von Erwartungswert zu zeigen. Ein wichtiger Ansatz, beschränkte Abhängigkeit auszunutzen, ist mit Martingalen verbunden. Als nächstes führen wir einige Begriffe ein, welche in der Definition von Martingalen benutzt werden.

Definition 2.23 *Es sei eine σ-Algebra $(\Omega, \mathcal{F})$ mit $\mathcal{F} = \mathcal{P}(\Omega)$ gegeben. Eine Folge $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_m$ von Teilmengen von $\mathcal{P}(\Omega)$ heißt ein* Filter*, wenn*

1. $\mathcal{F}_0 = \{\emptyset, \Omega\}$
2. $\mathcal{F}_m = \mathcal{P}(\Omega)$
3. $(\Omega, \mathcal{F}_i)$ *für alle i eine σ-Algebra ist.*

Wenn $E_1, E_2, \ldots$ disjunkte Ereignisse sind, so daß $\mathcal{F}$ die minimale Menge von Teilmengen ist, welche $E_1, E_2, \ldots$ und $\emptyset$ enthält und unter Komplementierung und Vereinigung abgeschloßen ist, dann sagen wir, daß die Ereignisse $E_1, E_2, \ldots$ die Menge $\mathcal{F}$ erzeugen, und $E_1, E_2, \ldots$ werden als Elementarereignisse bezeichnet.
Ein Filter kann so verstanden werden: Zu Ω wird eine Folge von sich verfeinernden Partitionen gewählt, so daß die Elemente der $i$ten Partition $\mathcal{F}_i$ erzeugen. Die erzeugenden Ereignisse von $\mathcal{F}_m$ sind dann einelementig.

Definition 2.24 *Für eine σ-Algebra $(\Omega, \mathcal{F})$ und eine Zufallsvariable X ist der bedingte Erwartungswert $E[X|\mathcal{F}]$ definiert durch $E[X|Y]$ für eine Zufallsvariable Y, die auf den Elementarereignissen von $\mathcal{F}$ jeweils unterschiedliche Werte annimmt.*

Die Definition von $E[X|\mathcal{F}]$ hängt offenbar nicht von den spezifischen Werten ab, welche Y auf den Elementarereignissen hat, sondern nur davon, daß Y jeweils einen anderen Wert auf jedem Elementarereignis hat.

Definition 2.25 *Für eine σ-Algebra $(\Omega, \mathcal{F})$ ist eine Zufallsvariable $X : \Omega \to S$ genau dann $\mathcal{F}$-meßbar, wenn für alle $x \in S$ das Ereignis „$X = x$“ in $\mathcal{F}$ liegt.*

Definition 2.26 *Sei $(\Omega, \mathcal{F})$ eine σ-Algebra und $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_m$ ein Filter.*
Weiterhin seien $X_0, X_1, \ldots$ Zufallsvariablen so daß jedes X_i $\mathcal{F}_i$-meßbar ist. Dann ist die Folge $X_0, X_1, \ldots$ ein Martingal, wenn für alle $i \geq 0$ gilt, daß

$$E[X_{i+1}|\mathcal{F}_i] = X_i.$$

Wir geben jetzt einen einfachen Weg an, ein Martingal zu erzeugen.

Fakt 2.26 *Sei $(\Omega, \mathcal{F}, \Pr)$ ein Wahrscheinlichkeitsraum mit einem Filter $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_m$.*
X sei eine Zufallsvariable über dem Wahrscheinlichkeitsraum und $Z_i = E[X|\mathcal{F}_i]$. Die Folge der $Z_0, Z_1, \ldots, Z_m$ ist ein Martingal. Solche Folgen heißen auch Doob Martingale.

Folgende Ungleichung erlaubt es, Abweichungen vom Erwartungswert für Martingale zu beschränken.

Fakt 2.27 (Azuma) *Sei $Z_0, Z_1, \ldots$ ein Martingal, bei dem für alle $i > 0$*

$$|Z_i - Z_{i-1}| \leq c_i,$$

wobei diese Distanz den maximalen Unterschied zwischen den Zufallsvariablen angibt. Dann gilt für alle $t > 0$ und $\lambda > 0$:

$$Pr(|Z_t - Z_0| \geq \lambda) \leq 2 \exp\left(-\frac{\lambda^2}{2\sum_{i=1}^{t} c_i^2}\right).$$

Ein weiterer wichtiger Begriff ist die Distanz zwischen Verteilungen bzw. Dichtefunktionen von Zufallsvariablen.

Definition 2.27 *Die Distanz zwischen den Dichtefunktionen p_X und p_Y von Zufallsvariablen X, Y mit Wertebereich $\{1, \ldots, n\}$ ist*

$$||p_X - p_Y|| = \sum_{i=1}^{n} |p_X(i) - p_Y(i)|.$$

Im nächsten Abschnitt wird folgendes nützlich sein.

Fakt 2.28 (Jensen) *Sei $f : \mathbb{R} \to \mathbb{R}$ eine konkave Funktion und X eine Zufallsvariable mit reellen Werten. Dann gilt*

$$E[f(X)] \leq f(E[X]).$$

2.2.2 Informationstheorie

Wir definieren nun einige wichtige Begriffe der Informationstheorie [Bl87, G90].

Definition 2.28 *Sei $(\Omega, \mathcal{F}, \Pr)$ ein Wahrscheinlichkeitsraum und X eine Zufallsvariable mit Wertebereich $S = \{x_1, \ldots, x_n\}$ und Dichtefunktion p_X.*
Die Entropie von X ist $H(X) = -\sum_{x \in S} p_X(x) \log p_X(x)$.
Die Entropie von X gegeben ein Ereignis E ist
$H(X|E) = -\sum_{x \in S} \Pr(X = x|E) \log \Pr(X = x|E)$.
Die bedingte Entropie von X gegeben eine Zufallsvariable Y über demselben Wahrscheinlichkeitsraum ist $H(X|Y) = \sum_y \Pr(Y = y) H(X|Y = y)$, wobei die Summme über die Werte von Y läuft. Damit ist $H(X|Y) = H(XY) - H(Y)$.
Die relative Entropie zwischen zwei Zufallsvariablen X, Y mit einem gemeinsamen Wertebereich $S = \{x_1, \ldots, x_n\}$ ist gegeben durch
$R(X|Y) = \sum_{x \in S} \Pr(X = x) \log(\Pr(X = x)/\Pr(Y = x))$. Gilt für ein x, daß $\Pr(Y = x) = 0$ und $\Pr(X = x) > 0$, so ist die relative Entropie unendlich.

Die Information zwischen X und Y ist $H(X:Y) = H(X) - H(X|Y)$.
Die bedingte Information zwischen X und Y, gegeben Z, ist $H(X:Y|Z) = H(XZ) + H(YZ) - H(Z) - H(XYZ)$.
Für $\alpha \in [0,1]$ wird auch $H(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$ definiert.
Alle obigen Definitionen benutzen die Konvention $0 \log 0 = 0$.

Das nächste Ergebnis wird in [NW93] und [RW89] benutzt, um die Wahrscheinlichkeit von Ereignissen zu einer Zufallsvariable mit hoher Entropie durch deren uniforme Wahrscheinlichkeit (d.h. unter der Gleichverteilung) zu approximieren.

Fakt 2.29 *Wenn $E \subseteq S = \{x_1, \ldots, x_n\}$ ein Ereignis zu einer Zufallsvariable X mit dem Wertebereich S ist, und wenn $q = |E|/n \leq 1/5$ und*

$$\Delta = \sqrt{\frac{4(\log n - H(X))}{q}} \leq \frac{1}{10},$$

so gilt $|\Pr(E) - q| \leq q\Delta$.

Beweis: Wir zeigen, daß $\Pr(E) \leq q(1+\Delta)$, der andere Fall $\Pr(E) \geq q(1-\Delta)$ kann analog gezeigt werden.
Die Ereignisse $\{X = x_i\}$ bezeichnen wir im folgenden als Elementarereignisse. Zuerst verschieben wir die Wahrscheinlichkeiten von Pr zu einem Maß $\Pr'$, indem die Wahrscheinlichkeiten der nq größten Elementarereignisse umverteilt werden, so daß alle die gleiche Wahrscheinlichkeit haben. Dann wird die Gesamtwahrscheinlichkeit der restlichen Elementarereignisse so umverteilt, daß alle die gleiche Wahrscheinlichkeit haben. Offensichtlich wird so die Summe a der Wahrscheinlichkeiten der größten qn Elementarereignisse (für die offenbar $a \geq \Pr(E)$ gilt) nicht verändert. Die Entropie der Zufallsvariablen X' mit der neuen Dichtefunktion kann durch die Transformation nicht gegenüber der Entropie von X gefallen sein, also gilt $\log n - H(X') \leq \log n - H(X)$. Angenommen

$$\Pr{}'(X = x) = \begin{cases} (1+\xi)/n & \text{für die } qn \text{ größten Elementarereignisse} \\ \frac{1-q(1+\xi)}{(1-q)n} & \text{für die anderen Elementarereignisse.} \end{cases}$$

Die zweite Wahrscheinlichkeit ist gleich $(1 - \frac{\xi q}{1-q})/n$. Es gilt

$$\begin{aligned} \log n - H(X) &\geq \log n + nq \cdot (1+\xi)/n \cdot \log((1+\xi)/n) \\ &+ (1-q)n \cdot (1 - \frac{\xi q}{1-q})/n \cdot \log((1 - \frac{\xi q}{1-q})/n) \\ &\geq q(1+\xi)\log(1+\xi) + (1-q)\log(1 - \xi q/(1-q)). \end{aligned}$$

Mit Hilfe von Taylorreihen kann man nun zeigen, daß für $\alpha \geq -1/10$:

$$\ln(1+\alpha) = \alpha - \alpha^2/2 + R_\alpha,$$

wobei $|R_\alpha| \leq |\alpha|^3$ und $R_\alpha \cdot \alpha \geq 0$. Wir erhalten unter der Annahme $\xi \leq 1/10$ und $q \leq 1/5$:

$$\begin{aligned}
\log n - H(X) &\geq \log e \cdot [q(1+\xi)(\xi - \xi^2/2) \\
&+ (1-q)(-\xi q/(1-q) + \xi^2 q^2/(2(1-q)^2) - \xi^3 q^3/(1-q)^3)] \\
&\geq \log e \cdot [(q+q\xi)(\xi - \xi^2/2) - (\xi q - \xi^2 q^2/2 + \xi^3 q^3/(1-q)^2)] \\
&\geq \log e \cdot [q\xi^2/2 - q\xi^3/2 - q^3\xi^3/(1-q)^2].
\end{aligned}$$

Angenommen $\xi \leq 1/10$ und $q \leq 1/5$, so erhalten wir $\log n - H(X) \geq q\xi^2/4$ bzw. $\xi \leq \sqrt{\frac{4(\log n - H(X))}{q}} = \Delta$.
Da aber $\log n - H(X')$ streng monoton steigend in ξ ist, war die Annahme $\xi \leq 1/10$ gerechtfertigt: wäre $\xi > 1/10$, so wäre $\sqrt{\frac{4(\log n - H(X))}{q}} \geq \sqrt{\frac{4(\log n - H(X'))}{q}} > 4\sqrt{\frac{4(\log n - H(X_{1/10}))}{q}} \geq 1/10$, wenn $X_{1/10}$ die Variable X' für $\xi = 1/10$ ist. Das ist aber ein Widerspruch zur Annahme an Δ.
Insgesamt erhalten wir $\Pr(E) \leq a \leq q(1+\Delta)$. □
Das folgende Resultat ist eine vereinfachte Version von Fanos Ungleichung (siehe [G90]).

Fakt 2.30 *Sind X, Y zwei Boolesche Zufallsvariablen mit $\Pr(X \neq Y) \leq \epsilon$, dann gilt $H(X:Y) \geq H(X) - H(\epsilon)$.*

Beweis: Sei $Z = 1 \iff X = Y$ und $Z = 0 \iff X \neq Y$. Dann gilt $H(X|Y) = H(XY) - H(Y) = H(ZY) - H(Y) \leq H(Z) \leq H(\epsilon)$. □
Das nächste Lemma ist ähnlich, allerdings in einer „Las Vegas Variante“.

Lemma 2.3 *Gegeben seien eine Zufallsvariable X mit endlichem Wertebereich S sowie eine Zufallsvariable Y mit Wertebereich $S \cup \{x_?\}$, so daß $\Pr(Y = x | X = x) \geq 1 - \epsilon$ für alle $x \in S$, $\Pr(Y = x | X \neq x) = 0$ für alle $x \neq x_?$ und $\Pr(Y = x_? | X = x) \leq \epsilon$ für alle $x \in S$. Dann gilt $H(X:Y) \geq (1-\epsilon)H(X)$.*

Beweis: $H(X:Y) = H(X) - H(X|Y)$. Sei $\delta = \Pr(Y = x_?) \leq \epsilon$ und $\epsilon_x = \Pr(Y = x_? | X = x) \leq \epsilon$ und $p_x = \Pr(X = x)$.

$$\begin{aligned}
H(X|Y) &\leq (1-\delta)H(X|Y \neq x_?) + \delta H(X|Y = x_?) \\
&= \delta H(X|Y = x_?) \\
&= -\delta \sum_x \Pr(X = x | Y = x_?) \log(\Pr(X = x | Y = x_?)) \\
&= -\delta \sum_x (\epsilon_x p_x/\delta) \log(\epsilon_x p_x/\delta) \\
&\leq -\epsilon \sum_x p_x \log p_x + \delta \sum_x (\epsilon_x p_x/\delta) \log(\delta/\epsilon_x) \\
&\leq \epsilon H(X) + \delta \log \sum_x p_x \text{ mit Jensens Ungleichung} \\
&\leq \epsilon H(X).
\end{aligned}$$

□

Wir benötigen noch die Kraftsche Ungleichung [MO00].

Fakt 2.31 *Sei code* : $\{1,\ldots,k\} \to \{0,1\}^*$ *eine injektive Funktion und* d_i *bezeichne die Länge von code*(i)*. Dann gilt*

$$\sum_{i=1}^{k} 2^{-d_i} \leq 1.$$

Das nächste Resultat (siehe z.B. [Bl87]) verbindet die Distanz zwischen Dichtefunktionen mit der relativen Entropie. Ein verallgemeinerter Beweis für Quanteninformationstheorie wird in Abschnitt 5.2.2 gegeben.

Fakt 2.32 *Für alle Zufallsvariablen* X, Y *mit gemeinsamem Wertebereich gilt*

$$R(X|Y) \geq \frac{1}{2\ln 2}||p_X - p_Y||^2.$$

2.2.3 Die VC-Dimension

Wir benötigen das kombinatorische Konzept der Vapnik-Chervonenkis Dimension [VC71].

Definition 2.29 *Eine Menge* S *wird von einer Menge Boolescher Funktionen* $\mathcal{F}$ *zerschmettert, wenn es für jedes* $R \subseteq S$ *eine Funktion* $f \in \mathcal{F}$ *gibt, so daß für alle* $x \in S$*:* $f(x) = 1 \iff x \in R$*.*
Die Größe einer größten von $\mathcal{F}$ *zerschmetterten Menge ist die VC-Dimension* $VC(\mathcal{F})$ *von* $\mathcal{F}$*.*

Fakt 2.33 *Sei* $\mathcal{F}$ *eine Menge Boolescher Funktionen* $f : X \to \{0,1\}$*. Dann gilt*

$$2^{VC(\mathcal{F})} \leq |\mathcal{F}| \leq (|X|+1)^{VC(\mathcal{F})}.$$

Beweis:
Für die erste Ungleichung beobachte man, daß eine zerschmetterte Menge mit $d = VC(\mathcal{F})$ Elementen mindestens 2^d Funktionen in $\mathcal{F}$ erzwingt, welche die 2^d Teilmengen erzeugen.
Die zweite Ungleichung wird per Induktion über die Größe von X bewiesen, wobei X die Menge der Eingaben der Funktionen ist. Für $|X| = 1$ ist die Behauptung trivial.
Sei also $|X| > 1$. Man nehme eine Eingabe $x \in X$ und partitioniere die Funktionen in zwei Mengen. $\mathcal{F}_0$ enthalte alle f so daß $f(x) = 0$ gilt und es eine Funktion $g \in \mathcal{F}$ mit $g(x) = 1$ und $g(y) = f(y)$ für alle $y \neq x$ gibt. Sei $\mathcal{F}_1 = \mathcal{F} - \mathcal{F}_0$.
Natürlich kann man $\mathcal{F}_0$ so betrachten, als hätten die Funktionen in $\mathcal{F}_0$ die Menge $X - \{x\}$ als Definitionsbereich der Größe $|X| - 1$. Außerdem ist $VC(\mathcal{F}_0) \leq VC(\mathcal{F}) - 1$:

Denn wenn $\mathcal{F}_0$ die Menge $S \subseteq X - \{x\}$ zerschmettert, dann zerschmettert $\mathcal{F}$ auch $S \cup \{x\}$.
Sei $T \subseteq S$. Wenn $x \notin T$, dann gibt es eine Funktion $f \in \mathcal{F}_0$, so daß $f(y) = 1 \iff y \in T$. Ansonsten gibt es eine Funktion $f \in \mathcal{F}_0$ mit $f(y) = 1 \iff y \in T - \{x\}$. Dann erfüllt die Funktion g gegeben für f in der Definition von $\mathcal{F}_0$ die Eigenschaft $g(y) = 1 \iff y \in T$.
$\mathcal{F}_1$ kann man als eine Menge von Funktionen über $X - \{x\}$ betrachten, weil alle $f, g \in \mathcal{F}_1$ sich auf Eingaben aus $X - \{x\}$ paarweise unterscheiden. Auch gilt $VC(\mathcal{F}_1) \leq VC(\mathcal{F})$.
Daher folgt

$$\begin{aligned} |\mathcal{F}| &= |\mathcal{F}_0| + |\mathcal{F}_1| \\ &\leq (|X| - 1 + 1)^{VC(\mathcal{F})-1} + (|X| - 1 + 1)^{VC(\mathcal{F})} \\ &= |X|^{VC(\mathcal{F})-1}(|X| + 1) \\ &< (|X| + 1)^{VC(\mathcal{F})}. \end{aligned}$$

□

Wir betrachten nun die VC-Dimension auch für zweistellige Funktionen.

Definition 2.30 *Für eine Funktion $f : X \times Y \to \{0, 1\}$ sei $\mathcal{F} = \{g | \exists x \in X : \forall y \in Y : g(y) = f(x, y)\}$. Dann definieren wir $VC(f) = VC(\mathcal{F})$.*

$D^{(1)}(f)$ ist die deterministische Einweg Kommunikationskomplexität der (zweistelligen) Funktion f. Wir werden sehen, daß bei obiger Definition $\lceil \log |\mathcal{F}| \rceil = D^{(1)}(f)$ gilt (Fakt 3.1). Daher ist $VC(f) \leq D^{(1)}(f) \leq \lceil \log(|Y| + 1) \cdot VC(f) \rceil$.

Kapitel 3

Probabilistische Kommunikation

3.1 Überblick

Probabilistische Kommunikation ist ein natürliches Modell der Kommunikation, da es sowohl realisierbar als auch mächtig ist. Wie bereits gesehen, kann probabilistische Kommunikation exponentiell effizienter als deterministische Kommunikation sein (Fakt 2.8). Wir beschreiben im folgenden Resultate, welche sich mit der Natur beschränkter Interaktion in der Kommunikationskomplexität beschäftigen.
Als neues Ergebnis geben wir eine Verbesserung einer Rundenhierarchie für probabilistische Kommunikationskomplexität gegenüber dem Resultat in [NW93] an. In gewisser Hinsicht rechtfertigen die Resultate dieses Kapitels, daß ein Kapitel über deterministische Kommunikation fehlt. Deterministische Einweg Kommunikation kann trivial charakterisiert werden, und die Ergebnisse zur probabilistischen Rundenhierarchie in Abschnitt 3.2 führen auch zu einer deterministischen Hierarchie. Man kann sagen, daß probabilistische Kommunikation im wesentlichen fast genauso abhängig von Interaktion ist, wie deterministische Kommunikation und sich damit wesentlich von nichtdeterministischer Kommunikation unterscheidet. Diese Abhängigkeit scheint eine wichtige Eigenschaft realistischer Kommunikationsmodelle zu sein.
Der erste Abschnitt des Kapitels enthält Techniken zum Beweis unterer Schranken für probabilistische Einweg Kommunikation. Erstaunlicherweise kann dieses Maß (im Fall beschränkten Fehlers) bisher jedoch nicht gut kombinatorisch charakterisiert werden, während solche Charakterisierungen für deterministische und nichtdeterministische Einweg Kommunikation trivial sind. Die einzige bekannte generelle Methode für untere Schranken für die probabilistische Einweg Kommunikation basiert auf der VC-Dimension aus Abschnitt 2.2.3.

Danach beschreiben und verbessern wir eine Rundenhierarchie für probabilistische Kommunikationskomplexität von Nisan und Wigderson [NW93], und führen dabei die Pointer Jumping Funktion f_k ein, die zentral für Betrachtungen über beschränkte Interaktion ist. Bei dieser Funktion muß im wesentlichen der $k+2$te Knoten eines Pfades in einem bipartiten Graphen mit $2n$ Knoten berechnet werden. Für diese Funktion gilt trivialerweise $D^{(k)}(f_k) \leq k \log n$. Wir leiten das beste bisher bekannte probabilistische Protokoll für f_k her, bei dem B beginnt und k Runden benutzt werden. Hierbei verwenden wir eine Kombination von Ideen aus [NW93] und [PRV99]. Es ist bekannt, daß $D^{(B,k)}(f_k) = O(n)$ wenn k konstant [PRV99] und daß $R^{(B,k)}(f_k) = O(\frac{n}{k} \log n + k \log n)$ [NW93].

Theorem 3.1 $R^{(B,k)}(f_k) = O(\frac{n}{k} \cdot (\log^{(k/2)} n + \log k) + k \log n)$.

Nisan und Wigderson [NW93] haben $R^{(B,k,pub)}(f_k) = \Omega(n/k^2 - k \log n)$ gezeigt. Das folgende Ergebnis kann mit einer Modifikation ihres Beweises gezeigt werden.

Theorem 3.2 $R^{(B,k,pub)}(f_k) = \Omega(n/k + k)$.

Wie die obere Schranke zeigt, gibt es eine Familie von probabilistischen Protokollen, welche f_k in Abhängigkeit von k effizienter berechnen können als $O(n)$, obwohl B die Kommunikation beginnt und nur k Runden erlaubt sind. Wir untersuchen in einem weiteren Abschnitt, wie der Informationsgewinn, den die Spieler in jeder Runde machen, genau gemessen werden kann, was in vorherigen Beweisen nicht möglich war. Wir erhalten eine weitere untere Schranke für Pointer Jumping mit einer neuen informationstheoretischen Methode. Diese Schranke ist zwar asymptotisch schlechter als das obige Ergebnis, aber sowohl methodisch interessant, als auch von den konstanten Faktoren der unteren Schranke, welche erheblich geringer sind als im Beweis von Theorem 3.2.

Theorem 3.3 $R_\epsilon^{(B,k,pub)}(f_k) \geq \frac{(1-2\epsilon)^2}{2k^2} \cdot n - k \log n$.

Abschnitte 3.3.1 und 3.3.2 beschreiben Anwendungen auf probabilistische Automaten. In Fakt 3.2 wird eine untere Schranke für die probabilistische Einweg Kommunikationskomplexität mittels des VC-Dimension gegeben. Eine ihrer Anwendungen ist das folgende Resultat.

Korollar 3.2 *Sei L eine reguläre Sprache.*
Dann hat jeder probabilistische Einweg Automat mit beschränktem Fehler für L mindestens $2^{\Omega(VC(L))}$ Zustände.

Weiterhin wird gezeigt, daß der Ansatz, die Größe von pfa durch probabilistische Einweg Kommunikation zu beschränken, manchmal im Vergleich zur wirklichen Größe exponentiell kleinere Schranken hervorbringt.

Für probabilistische Zweiweg Automaten wird die folgende Hierarchie gezeigt.

Theorem 3.4 *Für alle k und n gibt es eine Sprache $L_k \subseteq \{0,1\}^n$, welche von einem deterministischen k-reversalbeschränkten Zweiweg Automaten mit $O(kn)$ Zuständen erkannt wird. Probabilistische $(k-1)$-kreuzungsbeschränkte Zweiweg Automaten für L_k haben mindestens $2^{\Omega(n/(k^2 \log n))}$ Zustände.*

Danach wird in Abschnitt 3.3.3 die Länge probabilistischer Formeln eingehend untersucht. Als untere Schranken werden die Nečiporuk Funktion [N66] (siehe auch [BS90]) sowie eine neue Funktion hergeleitet. Diese neue Funktion ist mit der Nečiporuk Funktion verwandt, und entsteht informell gesprochen durch eine Ersetzung des Logarithmus' der Größe der Menge der Subfunktionen durch deren VC-Dimension.

Korollar 3.3 *Die VC-Nečiporuk Funktion ist eine asymptotische untere Schranke für die Länge starker probabilistischer Formeln mit beschränktem Fehler.*
Die (konventionelle) Nečiporuk Funktion ist asymptotisch eine untere Schranke für die Länge von starken Las Vegas Formeln für totale Funktionen.

Weiterhin wird der folgende Unterschied zwischen probabilistischen Formeln mit und ohne Fehler gezeigt. Es wird auch bewiesen, daß dieser Unterschied maximal ist unter der Annahme, daß die untere Schranke mit der (konventionellen) Nečiporuk Methode gezeigt wird.

Korollar 3.4 *Es gibt eine Funktion, die von fairen Monte Carlo Formeln der Länge $O(N)$ berechnet wird, für die aber jede starke Las Vegas Formel die Länge $\Omega(N^{3/2})$ hat, d.h. es gibt einen Unterschied von $\Omega(N^{1/2})$ zwischen Las Vegas und Monte Carlo Formellänge.*
Es gibt ebenfalls einen Unterschied von $\Omega(N^{1/2})$ zwischen Monte Carlo Formellänge und der Länge von probabilistischen Formeln mit beschränktem Fehler.

Als einen Kandidaten für eine Funktion, bei der ein größerer Unterschied bestehen könnte, untersuchen wir verallgemeinerte Matrixmultiplikation.
Die Resultate dieses Kapitels sind zum Teil in [Kl97] und [Kl00a] publiziert.

3.2 Resultate zur Kommunikationskomplexität

3.2.1 Probabilistische Einweg Kommunikation

Die deterministische Einweg Kommunikationskomplexität einer Funktion kann wie folgt charakterisiert werden [Hr97]. Sei $row(f)$ die Anzahl der verschiedenen Zeilen der Kommunikationsmatrix von f.

Fakt 3.1 $D^{(1)}(f) = \lceil \log row(f) \rceil$.

Es ist also relativ einfach, die deterministische Einweg Kommunikationskomplexität abzuschätzen. Wenn man als Beispiel die Index Funktion anschaut, so gilt offenbar $D^{(B,1)}(IX_n) = \log n$. Es ist aber mit Fakt 3.1 leicht zu sehen, daß $D^{(1)}(IX_n) = n$, da es 2^n verschiedene Zeilen in der Kommunikationsmatrix von IX_n gibt. In [KNR95] wird gezeigt, daß auch $R^{(1,pub)}(IX_n) = \Omega(n)$, siehe auch [Ab93]. Daher ist Einweg Kommunikation eine starke Einschränkung. Eine generelle untere Schranke wird ebenfalls in [KNR95] gezeigt.

Fakt 3.2 $R^{(1,pub)}(f) = \Omega(VC(f))$

Wir verallgemeinern dieses Resultat auf Quanten Einweg Protokolle in Kapitel 5. Da aber bekannt ist [KNR95], daß es Funktionen gibt, bei denen $VC(f) = O(1)$, aber $R^{(1,pub)}(f) = \Omega(n)$ gilt, ist obige Methode unter Umständen sehr schlecht. Es ist ein offenes Problem, eine geeignete kombinatorische Methode zum Beweis unterer Schranken für probabilistische Einweg Kommunikationskomplexität mit beschränktem Fehler zu finden, die für alle Funktionen gute Resultate liefert. Eine alternative Methode wird ebenfalls in Kapitel 5 diskutiert.
Las Vegas Kommunikation für totale Funktionen kann quadratisch effizienter sein als deterministische Kommunikation (Fakt 2.10/2.11). Die entsprechende Aussage für Einweg Protokolle ist falsch [DHRS97].

Fakt 3.3 *Für alle totalen Funktionen f gilt:*
$R^{(1,pub)}_{0,1/2}(f) \geq D^{(1)}(f)/2$.

Auch dieses Resultat wird in Kapitel 5 für Quantenkommunikation verallgemeinert.

3.2.2 Probabilistische k-Runden Protokolle

Die Pointer Jumping Funktion

In [PS84] wurde vermutet (und für $k = 2$ bewiesen), daß Protokolle mit k Runden bestimmte Funktionen wesentlich effizienter berechnen können als Protokolle mit $k - 1$ Runden. Ein erster allgemeiner Beweis befindet sich in [DGS87] für deterministische Protokolle. Ein existentieller Beweis einer solchen Rundenhierarchie für probabilistische Protokolle befindet sich in [HR93], eine Hierarchie für die sogenannte Pointer Jumping Funktion wird von Nisan und Wigderson [NW93] gezeigt. Ponzio et.al. [PRV99] geben eine genauere Analyse sowohl einer nicht Booleschen Variante als auch der Booleschen Variante dieser Funktion für den Fall von $k \leq \log^* n$ Runden an. Wir definieren zunächst die Pointer Jumping Funktion und gehen dann auf das Resultat von [NW93] ein.

Definition 3.1 *Seien V_A und V_B disjunkte Mengen von jeweils n Knoten. Sei $F_A = \{f_A | f_A : V_A \to V_B\}$ und $F_B = \{f_B | f_B : V_B \to V_A\}$.*
Weiter sei für feste f_A, f_B definiert, daß
$$f(v) = f_{f_A,f_B}(v) = \begin{cases} f_A(v) & \text{wenn } v \in V_A, \\ f_B(v) & \text{wenn } v \in V_B. \end{cases}$$
Definiere $f^{(0)}(v) = v$ und $f^{(k)}(v) = f(f^{(k-1)}(v))$.
Dann ist $g_k : F_A \times F_B \to (V_A \cup V_B)$ definiert durch $g_k(f_A, f_B) = f^{(k+1)}_{f_A,f_B}(v_1)$, wobei $v_1 \in V_A$ ein fixer Knoten ist.
Die Funktion $f_k : F_A \times F_B \to \{0,1\}$ ist das XOR aller Bits in der Binärdarstellung der Ausgabe von g_k.

Nisan und Wigderson haben in [NW93] bewiesen, daß f_k eine probabilistische k Runden Kommunikationskomplexität von $\Omega(n/k^2 - k \log n)$ hat, wenn B zuerst spricht. Die deterministische k Runden Kommunikationskomplexität von g_k ist offensichtlich $k \log n$, wenn A beginnt.

Protokolle für Pointer Jumping

In [NW93] wird ein probabilistisches Protokoll für g_k mit einer Kommunikation von $O((n/k) \log n + k \log n)$ beschrieben, das k Runden hat, wobei B beginnt. Ponzio et.al. [PRV99] zeigen, daß die deterministische Kommunikation von f_k in dieser Situation $O(n)$ ist, falls $k = O(1)$, dabei ist n wie auch im folgenden immer die Anzahl der Knoten, d.h. die Eingabelänge von g_k und f_k ist $2n \log n$.

Fakt 3.4 *Wenn $k = O(1)$, dann $D^{(B,k)}(f_k) = O(n)$.*

Wir beschreiben nun ein Protokoll, das für alle Werte von $k = \omega(1)$ die bestehenden Protokolle verbessert. Wir kombinieren dazu Ideen aus [NW93] und [PRV99].

Theorem 3.1 $R^{(B,k)}_{0,\epsilon}(g_k) = O(\frac{n}{k \cdot \epsilon} \cdot (\log^{(k/2)} n + \log k) + k \log n)$.

Beweis: B rät zufällig mit öffentlichen Zufallsbits $\lceil (4/\epsilon) \cdot (n/k) \rceil$ Knotennummern von 1 bis n. Für jeden geratenen Knoten v kommuniziert B die ersten $\lceil \log^{(k/2)} n + 3 \log k \rceil$ Bits des Pointers $f_B(v)$.
In Runde $t \geq 1$ kommuniziert der jeweils nächste Spieler den Knoten $v_t = f^{(t-1)}(v_1)$. Ist Spieler A an der Reihe ist, so prüft A, ob v_t in Bs Knotenliste der ersten Runde ist. Diese Liste ist A bekannt, da die Knotennummern öffentliche Zufallsbits sind. Ist v_t in der Liste, so kennt A $\log^{(k/2)} n + 3 \log k$ Bits von $f(v_t)$. Dies geschieht mit Wahrscheinlichkeit $1 - \epsilon$ während einer der ersten $k/2$ Runden: da die geratenen Knoten relativ zu den Knoten des Pfades zufällig sind, wird in in jeder geraden Runde mit Wahrscheinlichkeit $4/(\epsilon k)$ die Liste „getroffen“. Erwartet hat man also nach $\epsilon k/2$ Runden einen

solchen Erfolg, mit Wahrscheinlichkeit höchstens ϵ ist dies nach $k/2$ Runden immer noch nicht der Fall. Dann gibt das Protokoll auf.
Von der Runde i ab, in der A die ersten $\log^{(k/2)} n + 3\log k$ Bits von $f(v_i)$ erfährt, kommunizieren die Spieler in jeder Runde $i+t$ für alle möglichen Werte von $f(v_{i+t})$ jeweils die ersten $\log^{(k/2-t)} n + 3\log k$ Bits. Weil es höchstens $n/(\log^{(k/2-t)} n \cdot k^3)$ solche Werte gibt, reichen $O(n/k^2)$ Bits Kommunikation dafür aus. In der letzten Runde ist v_{k+2} gefunden. Insgesamt ist die Kommunikation höchstens

$$k \log n + O((1/\epsilon) \cdot (n/k)(\log^{(k/2)} n + 3\log k)) + k \cdot O(n/k^2).$$

Mit Fakt 2.4 erhält man ein Protokoll mit privatem Zufall. □

Korollar 3.1 *Für $k \geq 2\log^*(n)$ gilt $R^{(B,k)}(g_k) = O((\frac{n}{k} + k)\log k)$.*

Untere Schranken

Nisan und Wigderson [NW93] betrachten den Protokollbaum *deterministischer* Protokolle mit k Runden für f_k, bei denen B die Kommunikation beginnt, ein kleiner Fehler bezüglich der Gleichverteilung gemacht wird und die Kommunikation höchstens $\epsilon n - k\log n$ beträgt. Sie zeigen, daß solche Bäume für eine geeignet kleine Konstante ϵ nicht existieren. Nach Yaos Lemma (Fakt 2.3) impliziert dies eine untere Schranke für die probabilistische Kommunikationskomplexität von f_k mit dem entsprechenden Fehler.
Jeder Knoten z des Protokollbaums ist mit einer Teilmatrix der Kommunikationsmatrix assoziiert, welche genau die Eingaben enthält, welche diesen Knoten im Protokollbaum erreichen. Zu der Teilmatrix gehören Zufallsvariablen F_A^z, F_B^z, welche uniform auf den Zeilen bzw. Spalten der Teilmatrix verteilt sind. F_A^z ist dabei als ein Vektor von Zufallsvariablen $F_A^z(1), \ldots, F_A^z(n)$ zu verstehen. Wir nehmen an, daß in jeder Runde $t > 1$ der Knoten $v_t = f^{(t-1)}(v_1)$ kommuniziert wird. Erzwingt man eine solche Eigenschaft, werden höchstens $k\log n$ zusätzliche Bits kommuniziert, und die Gesamtkommunikation ist dann höchstens ϵn.
v_{t+1} als $t+1$ter Knoten des Pfades einer zufälligen Eingabe ist an einem Knoten z in Tiefe $t \geq 1$ im Baum eine Zufallsvariable, v_t ist durch vorige Nachrichten fixiert. c_z bezeichne die Anzahl von Bits, welche vom Protokoll bis z gesendet sind.
Nisan und Wigderson nennen einen Knoten z im Protokollbaum *gut*, wenn die Zufallsvariablen auf den Eingaben, welche den Knoten erreichen, folgende Eigenschaften haben (o.B.d.A. gelte, daß v_t auf der Seite von F_B liegt und A die vorige Nachricht gesendet hat, H sei die Entropie Funktion):

1. $H(F_A^z) \geq n\log n - 2c_z$
2. $H(F_B^z) \geq n\log n - 2c_z$

3. $H(F_B^z(v_t)) \geq \log n - \sqrt{\epsilon}/20$

(mit einer analogen Definition in der anderen Situation). Weiterhin sei die Wurzel gut. Das Hauptlemma von [NW93] kann wie folgt formuliert werden.

Fakt 3.5 *Angenommen z ist gut, und w ist a zufälliges Kind von z, wobei Kinder gezogen werden, indem gleichverteilt eine Eingabe $x \in F_A^z \times F_B^z$ gezogen und das durch x bestimmte Kind gewählt wird (die Wahrscheinlichkeit eines Kindes entspricht der Anzahl der Eingaben, die es erreichen). Dann gilt*
$\Pr[w \textit{ ist nicht gut }] \leq 22\sqrt{\epsilon} + \frac{1}{n}$.

Die Wurzel des Protokollbaums ist per Definition gut. Eine zufällige Eingabe in der Teilmatrix eines guten Knotens liegt mit Wahrscheinlichkeit $1 - 22\sqrt{\epsilon} - \frac{1}{n}$ in einem guten Kind. Also gilt für ein konstantes ϵ, daß immer mit Wahrscheinlichkeit $1/2$ ein gutes Kind erreicht wird. In Tiefe $k+1$ wird also ein gutes Blatt mit Wahrscheinlichkeit $1/2^{k+1}$ erreicht (bei gleichverteilter Wahl einer Eingabe). In Tiefe $k+1$ hat aber ein Spieler bereits das Ergebnis bekanntgegeben.
Ein gutes Blatt hat einen Fehler von mindestens $1/2 - O(\epsilon^{1/4})$ auf den Eingaben in seiner Teilmatrix (siehe [NW93]). Daher hat das Protokoll einen Fehler von insgesamt mindestens $\Omega(1/2^k)$ für eine genügend kleine Konstante ϵ. Dies impliziert insbesondere eine untere Schranke für fehlerlose deterministische Protokolle von $\Omega(n) - k \log n$, diese Schranke dann noch auf $\Omega(n)$ verbessert werden, siehe [NW93].
Wir versuchen nun, eine möglichst gute untere Schranke für Protokolle mit Fehler für f_k herzuleiten. Nisan und Wigderson zeigen, wie oben erläutert, daß deterministische Protokolle mit Kommunikation $\epsilon n - k \log n$ einen Fehler von mindestens $c/2^k$ auf der Gleichverteilung haben, für Konstanten ϵ, c.
Aber hätten Protokolle mit $\gamma \cdot n/k - \log n$ Kommunikation einen Fehler von höchstens $1/3$ für konstantes γ, so gäbe es Protokolle mit Kommunikation $\epsilon(\gamma, c)n - k \log n$ und Fehler $c/2^k$ für eine von γ und c abhängige Konstante $\epsilon(\gamma, c)$, deren Wert mit γ monoton fällt (siehe Fakt 2.5), und für genügend kleines γ ergäbe sich ein Widerspruch zu den vorigen Überlegungen.
Für Protokolle mit k Runden ist weiterhin eine untere Schranke von k trivial. Entweder n/k oder k übersteigen $\log n$ asymptotisch, und wir erhalten somit eine untere Schranke von $\Omega(n/k + k)$.

Theorem 3.2 $R^{(B,k,pub)}(f_k) = \Omega(n/k + k)$.

Die folgende untere Schranke für die nicht Boolesche Variante g_k von Pointer Jumping ist in [PRV99] bewiesen.

Fakt 3.6 $R^{(B,k,pub)}(g_k) = \Omega(n \log^{(k-1)} n)$.

Eine weitere untere Schranke für Pointer Jumping

In der oben beschriebenen unteren Schranke für Pointer Jumping werden informationstheoretische Argumente benutzt. Insbesondere wird die Entropie der Zufallsvariable des jeweils „nächsten“ Pointers betrachtet. Dann wird gezeigt, daß diese Variable mit hoher Wahrscheinlichkeit über die Blätter des Protokollbaums hohe Entropie hat. Da die Verteilung auf den Blättern dabei der Gleichverteilung auf allen Eingaben entspricht, kann man also sagen, daß eine zufällige Eingabe im Protokoll mit einer gewissen Wahrscheinlichkeit in eine Situation führt, in der der „letzte“ Pointer hohe Entropie hat. Es stellt sich nun die Frage, ob es nicht der Fall ist, daß man einfacher informationstheoretisch argumentieren kann. Betrachtet man die Zufallsvariable einer gleichverteilt gewählten Eingabe, ist die Information zwischen den ersten t Nachrichten und dem Pointer $F_{A/B}(v_t)$ klein? Für $t = 1$ ist dies sicher wahr, denn die erste Nachricht kommt von B und hat keine Information über $F_A(v_1)$. Obwohl Entropie ein Erwartungswert ist, und man annehmen könnte, daß die Information zwischen Nachrichten und „nächstem“ Pointer immer klein ist, zeigt der Beweis der unteren Schranke nur, daß die Entropie des nächsten Pointers nach Fixieren der Nachrichten mit hoher Wahrscheinlichkeit groß ist, dasselbe gilt für einen ähnlichen Beweis in [PRV99].
Es handelt sich hierbei aber nicht um ein Problem des Beweises. Das Protokoll für die Pointer Jumping Funktion g_k bzw. f_k zu Theorem 3.1 zeigt, daß ab dem Punkt, an dem Spieler A einen Knoten in Bs Knotenliste aus der ersten Runde findet, die Information der Nachrichten über den nächsten Pointer Runde pro Runde stark ansteigt.
Zur Illustration betrachten wir folgendes Beispiel eines 2 Runden Protokolls: B wählt mit öffentlichen Zufallsbits $\epsilon n/\log\log n$ Knoten und sendet für jeden davon die ersten $2\log\log n$ Bits. Wenn der erste Pointer von A, also $f_A(v_1)$, zu einem der geratenen Knoten zeigt, sendet A alle $n/\log^2 n$ möglichen nächsten Pointer $f_A(v_3)$ sowie $v_2 = f_A(v_1)$. Die Information dieser Nachricht zusammen mit Bs Eingabe (welche $v_3 = f_B(v_2)$ enthält) über den Pointer $f_A(v_3)$ erhöht sich auf $\Theta(\log n/\log\log n)$, obwohl man diesen nur mit Wahrscheinlichkeit $\Theta(1/\log\log n)$ voraussagen kann, wenn man die Nachricht und Bs Eingabe besitzt.
Wir betrachten im folgenden ein neues Maß von Information, um einen Wert zu erhalten, der eine Schranke für den Informationsanstieg pro Runde für die Pointer Jumping Funktion darstellt. Dies führt zu einem neuen Beweis für eine untere Schranke für Pointer Jumping. Während die neue untere Schranke asymptotisch schlechter als das Ergebnis ist, welches wir zuvor mit der Methode von [NW93] hergeleitet haben, hat der neue Beweis zwei Vorzüge. Zuerst erhalten wir eine direktere informationstheoretische Argumentation. Zum zweiten ist der konstante Faktor in der unteren Schranke wesentlich besser als bei [NW93]. Dort liegt die untere Schranke ungefähr bei $(n/k) \cdot 10^{-4}$ bei Fehler 1/3.

Da das Hauptproblem ist, daß mit geringer Wahrscheinlichkeit über die Zufallswahlen eines probabilistischen Protokolls die gewöhnliche Information zwischen Nachrichten und „nächstem“ Pointer zu groß wird, betrachten wir ein anderes Informationsmaß, das nie größer als konstant ist. Dieses Maß führen wir nun ein.
Für die gemeinsame Verteilung p_{AB} zweier Zufallsvariablen A und B seien p_A und p_B die Grenzverteilungen, d.h. $p_A(x) = \sum_y p_{AB}(x, y)$. Die Produktverteilung der Grenzverteilungen sei dann $p_A \otimes p_B$, d.h. $p_A \otimes p_B(x, y) = p_A(x) \cdot p_B(y)$. Wir erhalten mit Fakt 2.32.

$$H(A : B) = R(p_{AB}|p_A \otimes p_B) \geq \frac{1}{2 \ln 2}||p_{AB} - p_A \otimes p_B||^2,$$

wobei die Distanz zwischen Verteilungen wie in Definition 2.27 gemessen wird. Also kann die Distanz zwischen der Produktverteilung und der realen Verteilung durch die Information beschränkt werden. Wie nennen den Wert $D(A : B) = ||p_{AB} - p_A \otimes p_B||$ die *Distanzinformation.*
Das nächte Lemma stellt einige Eigenschaften der Distanzinformation zusammen.

Lemma 3.1 *Für alle Verteilungen p_{ABC} gilt:*

1. $D(A : B) = D(B : A)$.
2. $D(AB : C) \geq D(A : C)$.
3. $0 \leq D(A : B) \leq 2$.
4. $D(A : B) \geq ||F(p_{AB}) - F(p_A \otimes p_B)||$ *für alle Funktionen F, welche Verteilungen auf Verteilungen abbilden.*
5. $D(A : B) \leq \sqrt{2H(A : B)}$.

Beweis: Offensichtlich ist die Definition symmetrisch, daher gilt 1. 2. ist eine Konsequenz von 4. Ist $p_{AB} = p_A \otimes p_B$, so ist die Distanzinformation offenbar 0, kleiner als 0 kann eine Distanz aber nicht werden. Die Distanz ist immer kleiner als 2, 2 wird erreicht, wenn die beiden Verteilungen auf disjunkten Mengen positive Werte haben, somit gilt 3. 4. kann per Induktion gezeigt werden. □
p_{AB} sei die gemeinsame Verteilung von Zufallsvariablen A und B mit Werten $a_1, \ldots, a_k$ und $b_1, \ldots, b_l$. $p_B^{(a)}$ sei die (normalisierte) bedingte Verteilung von B, welche durch Fixierung von A auf ein a entsteht. $\Pr(ab)$ ist die Wahrscheinlichkeit von ab, $\Pr(a) = \sum_b \Pr(ab), \Pr(b|a) = \Pr(ab)/\Pr(a)$. $E_{a|b}$ bezeichne bedingte Erwartungswerte über a gegeben b. Die folgenden Eigenschaften der Distanzinformation werden später benutzt.

Lemma 3.2 *1. Sei p_{AB} eine Verteilung, wobei p_B die Verteilung einer Zufallsvariable B auf $\{0,1\}$ mit $\Pr(B=1)=\Pr(B=0)=1/2$ sei. Wenn es eine Abbildung von A auf eine Zufallsvariable X gibt, welche $\Pr(X=B)\geq 1-\epsilon$ und $\Pr(X\neq B)\leq\epsilon$ erfüllt (während dieselbe Abbildung für die Verteilung $p_A\otimes p_B$ eine Verteilung auf X mit $\Pr(X=B)=\Pr(X\neq B)=1/2$ ergibt), so gilt $D(A:B)\geq 1-2\epsilon$.*

2. Für alle p_{AB}, wobei p_B eine uniforme Verteilung U ist, gilt:

$$D(A:B)=E_a||p_B^{(a)}-U||.$$

3. Für alle p_{ABC}, wobei $p_C=U$ eine uniforme Verteilung ist, gilt:

$$D(A:C)=D(AB:C)\Rightarrow$$

$$\forall a \text{ mit } \Pr(a)>0: ||p_C^{(a)}-U||=E_{b|a}||p_C^{(ab)}-U||.$$

Beweis: Für die erste Behauptung beobachte man, daß $D(A:B)\geq D(X:B)$, und letzterer Wert ist durch $1-2\epsilon$ beschränkt.
Die zweite Behauptung folgt aus $D(A:B)=||p_{AB}-p_A\otimes U||$.
Die dritte Behauptung gilt, weil $D(A:C)=D(AB:C)$ impliziert, daß

$$E_a||p_C^{(a)}-U||=E_aE_{b|a}||p_C^{(ab)}-U||.$$

Weiterhin gilt für alle a:

$$||p_C^{(a)}-U||\leq E_{b|a}||p_C^{(ab)}-U||.$$

Wenn also für ein a (mit Wahrscheinlichkeit >0) $||p_C^{(a)}-U||<E_{b|a}||p_C^{(ab)}-U||$ wäre, hätten wir einen Widerspruch. □
Wir geben jetzt eine untere Schranke für Pointer Jumping an, in deren Beweis klar wird, daß die Distanzinformation zwischen Nachrichten und „nächstem Pointer“ Runde pro Runde nur wenig ansteigen kann. Außerdem ist die erhaltene Schranke für kleine k wegen ihrer Konstanten interessant.

Theorem 3.3 $R_\epsilon^{(B,k,pub)}(f_k)\geq\frac{(1-2\epsilon)^2}{2k^2}\cdot n-k\log n.$

Beweis: Wir betrachten also probabilistische Protokolle mit Fehler ϵ, k Runden, bei denen B beginnt. Die Kommunikation sei $\delta n/k^2-k\log n$. Wir werden sehen, daß diese Kommunikation für $\delta<(1-2\epsilon)^2/2$ nicht ausreicht, was die Behauptung beweist.
Wir betrachten die Gleichverteilung auf den Eingaben, d.h. F_A und F_B sind uniform zufällig gewählte linke und rechte Eingaben. Die zugehörige Verteilung auf den Eingaben und den Nachrichten der ersten t Runden sei $p_{F_AF_BM_t}$. Für jede Eingabe gibt es dabei eine Verteilung auf den Nachrichten, da das Protokoll probabilistisch ist.

Wir sind daran interessiert, daß das Protokoll die Eigenschaften $D(M_tF_A : F_B) = D(M_t : F_B)$ und $D(M_tF_B : F_A) = D(M_t : F_A)$ für alle t hat.
Diese Eigenschaft kann in einem Protokoll mit öffentlichem Zufall sehr einfach erhalten werden, wenn man alle Zufallsbits fixiert. M_t enthalte also die gesendeten Nachrichten bis Runde t. Wir betrachten im folgenden die öffentlichen Zufallsbits als beliebig fixiert.

Lemma 3.3 *Das Protokoll erfüllt* $D(M_tF_A : F_B) = D(M_t : F_B)$ *und* $D(M_tF_B : F_A) = D(M_t : F_A)$.

Wir zeigen zuerst $H(M_tF_A : F_B) = H(M_t : F_A)$ und schließen dann das Resultat für das D-Maß. Der Beweis verläuft per Induktion.
Für $t = 1$ ist die Behauptung trivial. Ansonsten ist $M_t = M_{t-1}M$, wobei M die Nachricht der $t$ten Runde ist. Dann ist $H(M_tF_A : F_B) = H(M_{t-1}MF_A : F_B)$. Wir unterscheiden zwei Fälle: A sendet Nachricht M bzw. B sendet Nachricht M.
Im ersten Fall gilt $H(M_{t-1}MF_A : F_B) = H(M_{t-1}F_A : F_B) = H(M_{t-1} : F_B) = H(M_t : F_B)$, weil M aus M_{t-1} und F_A berechnet wird. Der zweite Schritt gilt per Induktion.
Im zweiten Fall wird M von B gesendet. $H(M_{t-1}MF_A : F_B) = H(M_{t-1}MF_B : F_A) - H(M_{t-1}M : F_A) + H(M_{t-1}M : F_B)$. Per Induktion ist das gleich $H(M_{t-1} : F_A) - H(M_{t-1} : F_A) + H(M_{t-1}M : F_B) = H(M_t : F_B)$.
So gilt also $H(M_tF_A : F_B) = H(M_t : F_B)$ und aus Symmetriegründen auch $H(M_tF_B : F_A) = H(M_t : F_A)$. Das ist äquivalent zu $H(F_A : F_B|M_t) = 0$. Letzteres bedeutet, daß für alle Werte von M_t die induzierte Verteilung auf den Eingaben eine Produktverteilung ist.
Nun ist $D(M_tF_A : F_B) = E_m||p^{(m)}_{F_AF_B} - p^{(m)}_{F_A} \otimes U||$, wobei U die Gleichverteilung ist. Daher

$$\begin{aligned} D(M_tF_A : F_B) &= E_m||p^{(m)}_{F_AF_B} - p^{(m)}_{F_A} \otimes U|| \\ &= E_m||p^{(m)}_{F_A} \otimes p^{(m)}_{F_B} - p^{(m)}_{F_A} \otimes U|| \\ &= E_m||p^{(m)}_{F_B} - U|| \\ &= D(M_t : F_B). \end{aligned}$$

□

Wir bezeichnen mit v_1 den Startknoten und setzen $v_t = f^{(t-1)}(v_1)$. Wir verlangen, daß das Protokoll in Runde t den Knoten v_t kommuniziert. Dies erhöht die Kommunikation um $k \log n$ höchstens, die resultierende Kommunikation ist höchstens $\delta n/k^2$.
Die Strategie des Beweises ist eine Induktion über die Runden. Die erste Nachricht wird von B gesendet, offensichtlich gilt $H(M_1F_B : v_2) = 0$, weil B bisher noch keine Nachricht empfangen hat und v_2 von F_A bestimmt wird. Daher gilt auch $D(M_1F_B : v_2) = 0$. Die Induktionsvoraussetzung ist, daß

$D(M_tF_A : v_{t+1})$ bzw. $D(M_tF_B : v_{t+1})$ klein ist. Es kann allerdings sein, daß $H(M_tF_A : v_{t+1})$ größer als eine Konstante ist, dies geschieht z.B. im Protokoll zu Theorem 3.1: mit kleiner Wahrscheinlichkeit über die Zufallswahlen (z.B. $1/\log\log n$) wird die Information groß (z.B. $\log n$). Das kann bei der Distanzinformation nicht passieren, da sie durch 2 beschränkt ist und unwahrscheinliche „Spitzen“ hoher Information abgeschnitten werden.
Da also $H(M_1 : v_2) = 0$ kann man schließen, daß $H(M_2F_A : v_3) = H(M_1F_A : F_B(v_2)) \leq \delta/k^2$. Aber wenn $H(M_t : v_{t+1}) > 0$ kann ein solches Argument nicht weiter verfolgt werden. Stattdessen benutzen wir die Induktionsvoraussetzung $D(M_{t-1}F_{A/B} : v_t) \leq \gamma_{t-1}$ für $\gamma_t = (t-1) \cdot \frac{\sqrt{2\delta}}{k}$.
Wir beschränken zuerst $D(M_{t-1}v_t : F(v_t))$, und erhalten dann dieselbe Schranke für $D(M_{t-1}F_{A/B}v_t : F(v_t))$ mit Lemma 3.3, also für $D(M_tF_{A/B}v_t : F(v_t))$ mit Lemma 3.1.4.
O.B.d.A. nehmen wir an, daß v_t ein Knoten auf Bs Seite ist. Wir dürfen benutzen, daß $D(M_{t-1} : v_t) \leq \gamma_{t-1}$ und daß $\sum_{i=1}^n \frac{1}{n} H(M_{t-1} : F_B(i)) \leq \delta/k^2$, was $\sum_{i=1}^n \frac{1}{n} D(M_{t-1} : F_B(i)) \leq \sqrt{2\delta}/k$ impliziert (nach Lemma 3.1.5).

Lemma 3.4 $D(M_{t-1}v_t : F(v_t)) \leq D(M_{t-1} : v_t) + \sqrt{2\delta}/k$.

Der Einfachheit halber schreiben wir $v = v_t$ und $M = M_{t-1}$.
$D(Mv : F(v)) = E_m E_{v|m} ||p^{(mv)}_{F(v)} - U||$, wobei U die Gleichverteilung ist. Wir wissen, daß $D(M : v) = E_m ||p^{(m)}_v - U|| \leq \gamma_{t-1}$.
Betrachten wir irgendein m, für das $||p^{(m)}_v - U|| = \beta_m$. Dann ist $E_{v|m} ||p^{(m)}_{F(v)} - U|| \leq \beta_m + \sum_{i=1}^n \frac{1}{n} ||p^{(m)}_{F(i)} - U||$. Das gilt, weil höchstens $\beta_m/2$ Gewicht zu Verteilungen von $F(v)$ auf $\{1, \ldots, n\}$ mit größerer Distanz von U als erwartet verschoben wird, und alle Verteilungen voneinander höchstens Distanz 2 haben.

$$\begin{aligned}
D(Mv : F(v)) &= E_m E_{v|m} ||p^{(mv)}_{F(v)} - U|| \\
&= E_m E_{v|m} ||p^{(m)}_{F(v)} - U|| \text{ (wegen Lemma 3.3)} \\
&\leq E_m [\beta_m + \sum_{i=1}^n \frac{1}{n} ||p^{(m)}_{F(i)} - U||] \\
&= D(M : v) + \sum_i \frac{1}{n} D(M : F(i)) \\
&\leq D(M : v) + \sqrt{2\delta}/k \\
&\leq \gamma_t.
\end{aligned}$$

□

Lemma 3.5 $D(M_{t-1}v_tF_A : F_B(v_t)) = D(M_{t-1}v_t : F_B(v_t))$.

Wieder sei $v = v_t$ und $M = M_{t-1}$. Nach Lemma 3.3 erfüllt das Protokoll $D(MF_A : F_B) = D(M : F_B)$. Daher gilt für alle i: $D(MF_Av : F_B(i)) = D(Mv : F_B(i))$, weil v von M und F_A allein abhängt. Nach Lemma 3.2.3 folgt für alle i und alle v:

$$\begin{aligned} ||p^{(v)}_{MF_B(i)} - p^{(v)}_M \otimes U|| &= E_{m|v}E_{f_A|vm}||p^{(vmf_A)}_{F_B(i)} - U|| \\ &= E_{m|v}||p^{(vm)}_{F_AF_B(i)} - p^{(vm)}_{F_A} \otimes U||. \end{aligned}$$

Insbesondere gilt für $v = i$:

$$||p^{(v)}_{MF_B(v)} - p^{(v)}_M \otimes U|| = E_{m|v}||p^{(vm)}_{F_AF_B(v)} - p^{(vm)}_{F_A} \otimes U||$$

und daher

$$\begin{aligned} D(Mv : F_B(v)) = E_v||p^{(v)}_{MF_B(v)} - p^{(v)}_M \otimes U|| &= E_v||p^{(v)}_{MF_AF_B(v)} - p^{(v)}_{MF_A} \otimes U|| \\ &= D(MF_Av : F(v)). \end{aligned}$$

□

Wir können also schließen, daß $D(M_tF_A : F_B(v_t)) \leq D(M_{t-1}F_A : F_B(v_t)) = D(M_{t-1}v_t : F(v_t)) \leq D(M_{t-1} : v_t) + \sqrt{2\delta}/k \leq \gamma_t$. Dies gilt für alle Möglichkeiten, die öffentlichen Zufallsbits zu fixieren.
Nach Runde k gibt ein Spieler das Ergebnis bekannt, also enthalte M_{k+1} die Parität der Binärdarstellung von v_{k+2} mit hoher Wahrscheinlichkeit $1 - \epsilon$ über die Zufallsbits. Daher ist nach Lemma 3.1

$$k \cdot \sqrt{2\delta}/k \geq E[D(M_{k+1} : v_{k+2})] \geq E[D(M_{k+1} : \bigoplus v_{k+2})] \geq 1 - 2\epsilon.$$

Es folgt $\delta \geq (1 - 2\epsilon)^2/2$. □
Der obige Beweis zeigt, daß die Distanzinformation $D(M_t : v_{t+1})$ pro Runde nur um eine kleine additive Konstante steigt. Das gilt nicht für $H(M_t : v_{t+1})$. Das neue Informationsmaß beschreibt also den Informationsgewinn pro Runde besser.

3.3 Anwendungen

3.3.1 Probabilistische Einweg Automaten

Duris, Hromkovič, Rolim und Schnitger [DHRS97] haben beobachtet, daß die Größe eines minimalen deterministischen Einweg Automaten für eine Sprache L durch die Anzahl der Nachrichten in einem optimalen Einweg Protokoll für L charakterisiert werden kann.
Sei $L \subseteq \Sigma^*$ eine Sprache über einem endlichen Alphabet Σ. Die (unendliche) Kommunikationsmatrix von L besitzt Zeilen und Spalten indiziert mit Σ^*. Der Eintrag $M(x, y)$ ist 1 wenn $xy \in L$ und 0 ansonsten. Ein uniformes Einweg Protokoll für L ist ein Einweg Protokoll für das von der Matrix

induzierte Kommunikationsproblem. Begriffe wie Kommunikationskomplexität etc. sind wie zuvor definiert. Die uniforme Einweg Kommunikationskomplexität einer regulären Sprache L ist $D^{(1)}(L)$, eine analoge Notation wird für probabilistische Kommunikation verwendet.
Obwohl die Matrix unendlich ist, besitzt sie für reguläre Sprachen nur endlich viele verschiedene Zeilen, da deren Anzahl genau dem Nerode Index der Sprache entspricht. Andererseits ist die Anzahl verschiedener Zeilen aber auch gleich der Anzahl der verschiedenen Nachrichten in einem optimalen deterministischen Einweg Protokoll für L, wenn A einen Präfix x und B einen Suffix y erhält, und beide die Zugehörigkeit des Wortes xy zu L testen wollen. Da aber der Nerode Index gleich der minimalen Größe eines dfa für L ist, folgt:

Fakt 3.7 *Für jede reguläre Sprache L gilt: Die Größe des minimalen dfa für L ist gleich der Anzahl der verschiedenen Nachrichten in einem optimalen deterministischen uniformen Einweg Protokoll für L.*

Die untere Schranke im obigen Resultat läßt sich leicht auf probabilistische Automaten und Protokolle verallgemeinern.

Fakt 3.8 *Für jede reguläre Sprache L gilt: Die Größe eines minimalen pfa für L ist größer gleich der Anzahl der verschiedenen Nachrichten in einem optimalen uniformen probabilistischen Einweg Protokoll für L.*
Dies gilt für Las-Vegas pfa und Las-Vegas Protokolle, wie auch für pfa und Protokolle mit beschränktem Fehler.

Die VC-Dimension einer Sprache sei definiert als die VC-Dimension der Funktion $f(x, y) = 1 \iff xy \in L$. Mit Fakt 3.2 erhalten wir:

Korollar 3.2 *Sei L eine reguläre Sprache.*
Dann hat jeder probabilistische Einweg Automat mit beschränktem Fehler für L mindestens $2^{\Omega(VC(L))}$ *Zustände.*

Für Las-Vegas pfa wurde in [DHRS97] folgende Konsequenz von Fakt 3.3, Fakt 3.7 und Fakt 3.8 beobachtet:

Fakt 3.9 *Für jede reguläre Sprache L:*
Ein minimaler Las Vegas pfa für L hat mindestens $\Omega(\sqrt{d})$ *Zustände, wenn d die Größe des minimalen dfa für L ist.*

Die probabilistische Kommunikationskomplexität scheint die besten unteren Schranken für die Größe probabilistischer Automaten zu liefern. Aber während die untere Schranke für dfa für alle Sprachen exakt ist, kann die untere Schranke für pfa manchmal sehr schlecht sein.

Lemma 3.6 *Es gibt eine reguläre Sprache L, so daß die Größe probabilistischer Einweg Automaten für L mindestens $2^{\Omega(n)}$ ist, während es für L ein probabilistisches Einweg Protokoll mit $O(n^2)$ Nachrichten gibt (bei beschränktem Fehler).*

Beweis:
Sei $L = \{xyz : |x| = |y| = |z| = n$, und $x \neq z$ oder es gibt i mit $x_i = y_i = 1\}$.
Zuerst geben wir ein Protokoll mit $O(n^2)$ Nachrichten an.
Die Spieler A und B berechnen die Längen l_I, l_{II} ihrer Eingaben. A kommuniziert l_I und B verwirft, falls $l_I + l_{II} \neq 3n$. Im folgenden nehmen wir also $l_I + l_{II} = 3n$ an.
Fall 1: $l_I \leq n$.
A benutzt das probabilistische Einweg Protokoll für das Gleichheitsproblem aus Fakt 2.8 auf ihrer Eingabe und sendet die Nachricht des Protokolls (sowie l_I). B setzt das Protokoll auf den ersten l_I Bits von z fort und akzeptiert, falls das Protokoll verwirft, oder wenn die letzten $n - l_I$ Bits von x ungleich den letzten $n - l_I$ Bits von z sind. Ansonsten akzeptiert B genau dann, wenn $\bigvee_{i=1}^{n} y_i \wedge z_i$ gilt.
Wenn also $x \neq z$, so akzeptiert das Protokoll mit Wahrscheinlichkeit mindestens $1 - \epsilon$. Wenn aber $x = z$, so akzeptiert B genau dann, wenn $\bigvee_{i=1}^{n} y_i \wedge z_i$ gilt, d.h. wenn $\bigvee_{i=1}^{n} x_i \wedge y_i$ gilt.
Fall 2: $n < l_I \leq 2n$.
A benutzt das probabilistische Einweg Protokoll für das Gleichheitsproblem auf den x-Bits und sendet die Nachricht (sowie l_I). Außerdem sendet A das Bit $b = \bigvee_{i=1}^{l_I - n} x_i \wedge y_i$.
B setzt das Protokoll für Gleichheit auf z fort und akzeptiert, wenn das Protokoll verwirft. Ansonsten berechnet B den Wert $c = b \vee \bigvee_{i=l_I - n+1}^{n} y_i \wedge z_i$. Wenn $c = 1$, so akzeptiert B, sonst verwirft B.
Wenn also $x \neq z$, so akzeptiert das Protokoll mit Wahrscheinlichkeit mindestens $1 - \epsilon$. Sonst akzeptiert B genau dann, wenn $\bigvee_{i=1}^{n} x_i \wedge y_i$ gilt.
Fall 3: $2n < l_I \leq 3n$.
A benutzt das probabilistische Einweg Protokoll für das Gleichheitsproblem auf den letzten $3n - l_I$ Bits seiner x-Eingabe und sendet die Nachricht (sowie l_I). Desweiteren berechnet A den Wert $b = \bigvee_{i=1}^{n} x_i \wedge y_i$ und sendet b. A vergleicht auch die ersten $l_I - 2n$ Bits von x und z und sendet $c = 1$ bei Ungleichheit und $c = 0$ sonst.
Wenn $b = 1$, dann akzeptiert B. Wenn $c = 1$ so akzeptiert B. Sonst setzt B das Gleichheitsprotokoll auf seiner Eingabe fort und akzeptiert genau dann, wenn das Gleichheitsprotokoll verwirft.
Das Protokoll benutzt $O(n^2)$ Nachrichten, weil das Protokoll zu Fakt 2.8 $O(n)$ Nachrichten benutzt, und das Kommunizieren von l_I einen Faktor von n kostet.
Nun geben wir eine untere Schranke für die Anzahl der Zustände eines probabilistischen Automaten für L an. Dabei schränken wir uns auf Eingaben

der Form xyx ein, wobei $|x| = |y|$. Der pfa muß also nur testen, ob $\bigvee_{i=1}^{n} x_i \wedge y_i$ gilt.
Wir simulieren einen probabilistischen endlichen Automaten M für L mit s Zuständen durch ein probabilistisches Zweiweg Protokoll, bei dem A das Wort x und B das Wort y erhält. A simuliert den Automaten und kommuniziert den erhaltenen Zustand an B, B fährt mit der Simulation fort, sendet den erhaltenen Zustand an A, und A beendet die Simulation. Also erhalten wir ein probabilistische Protokoll mit beschränktem Fehler, 2 Runden und $2\lceil \log s \rceil$ Bits Kommunikation. Da aber das Disjunktheitsproblem gelöst wird, ist die Kommunikation nach Fakt 2.9 mindestens $\Omega(n)$ und die Größe ist $s = 2^{\Omega(n)}$. □

3.3.2 Probabilistische Zweiweg Automaten

Theorem 3.4 *Für alle k und n gibt es eine Sprache $L_k \subseteq \{0,1\}^n$, welche von einem deterministischen k-reversalbeschränkten Zweiweg Automaten mit $O(kn)$ Zuständen erkannt wird. Probabilistische $(k-1)$-kreuzungsbeschränkte Zweiweg Automaten für L_k haben mindestens $2^{\Omega(n/(k^2 \log n))}$ Zustände.*

Beweis: Wir verwenden die Pointer Jumping Funktion f_k auf $n/\log n$ Knoten (wie in Abschnitt 3.2.2 definiert). Dabei wird für die Worte der Sprache lediglich vorausgesetzt, daß die Pointer der linken Knoten in einem Eingabewort in irgendeiner festen Reihenfolge vor den Pointern der rechten Seite stehen (welche wieder einer feste Reihenfolge haben).
Ein k-reversalbeschränkter Automat beginnt beim ersten Pointer, liest ihn ein, läuft zur Position des nächsten Pointers usw. Dabei wechselt er nur k mal (inklusive des Startes) die Bewegungsrichtung. Der Automat ist aus k Teilautomaten zusammengesetzt. Jeder Teilautomat beginnt an der Position des nächsten zu lesenden Pointers, liest ihn, läuft zu angezeigten Position. Um dies zu implementieren enthält der Automat einen Pfad der Länge $O(n)$, in dem die Eingabe nur „überlesen“ wird. Nach Einlesen der Pointers (mit $O(n/\log n)$ Zuständen) „springt“ er an die richtige Stelle in dem Pfad, um genau die gesuchte Position zu erreichen. Der Pointer selbst kann dabei vergessen werden. Es reichen also für jede Stufe $O(n)$ Zustände aus, insgesamt $O(kn)$.
Andererseits kann ein probabilistischer $(k-1)$-kreuzungsbeschränkter Automat mit s Zuständen von einem probabilistischen Protokoll mit $k-1$ Runden und Kommunikation $(k-1) \cdot \lceil \log s \rceil$ Bits Kommunikation simuliert werden, das seine Eingaben genau wie bei f_k erhält. Also ist s mindestens 2^c mit $c = \Omega(n/(k^2 \log n))$ nach Theorem 3.2.

3.3.3 Die Länge probabilistischer Formeln

In diesem Abschnitt verallgemeinern wir die Nečiporuk Methode von deterministischen Formeln auf probabilistische Formeln. Dabei beobachten wir eine Verbindung der Methode zur Einweg Kommunikation und damit zur VC-Dimension. Informell gesagt ersetzen wir den Logarithmus der Anzahl der Subfunktionen durch die VC-Dimension der Menge der Subfunktionen und erhalten eine untere Schranke für probabilistische Formeln.
Unsere unteren Schranken gelten für das Modell starker probabilistischer Formeln. Korollar 3.4 gibt eine Funktion an, bei der sogar starker Probabilismus mit zweiseitigem Fehler nicht hilft, die Größe zu senken. Alle oberen Schranken werden für faire Formeln angegeben.
Wir zeigen, daß die Nečiporuk Schranke für totale Funktionen höchstens einen Faktor von $O(\sqrt{n})$ größer sein kann als die probabilistische Formellänge (Theorem 3.6). Wenn also eine totale Funktion kleine starke probabilistische Formeln hat, kann die konventionelle Nečiporuk Schranke nicht allzu groß werden.
Andererseits beschreibt Korollar 3.6 eine Funktion, für die faire probabilistische Formeln mit einseitigem Fehler um einen Faktor $\Theta(\sqrt{n})$ kleiner sind als starke Las Vegas Formeln. Ein analoger Unterschied besteht zwischen Formeln mit zweiseitigem Fehler und solchen mit einseitigem Fehler.
Die untere Schranke für Las Vegas Formeln benutzt die neue Beobachtung, daß die konventionelle Nečiporuk Methode auch asymptotische untere Schranken für Las Vegas Formeln liefert.
Schließlich schlagen wir eine Funktion vor, welche eine effiziente Monte Carlo Formel besitzt, und bei welcher es wahrscheinlich ist, daß deterministische Formeln fast quadratisch größer als diese Formel sein müssen. Ein Beweis einer solchen unteren Schranke benötigte allerdings neue Methoden zum Beweis unterer Schranken für die Länge deterministischer Formeln.

Eine Methode für untere Schranken

Wir leiten zunächst die Nečiporuk Schranke neu mit Hilfe der Einweg Kommunikation her, und verallgemeinern sie dann auf probabilistische Formeln.

Definition 3.2 *Sei f eine Boolesche Funktion auf n Eingaben und sei $y_1 \dots y_k$ eine Partition der Eingabevariablen in k Mengen.*
Wir betrachten k Kommunikationsprobleme für $i = 1, \dots, k$. Spieler B erhält jeweils alle Eingaben in y_i und Spieler A alle anderen Eingaben. Die deterministische Einweg Kommunikationskomplexität von f unter dieser Aufteilung der Eingaben heiße $D^{(1)}(f_i)$. Die probabilistische Einweg Kommunikationskomplexität von f (mit beschränktem Fehler und öffentlichen Zufallsbits) unter dieser Aufteilung der Eingaben heiße $R^{(1,pub)}(f_i)$.
Die probabilistische Nečiporuk Funktion ist $(1/4)\sum_i R^{(1,pub)}(f_i)$.

Es ist leicht zu sehen, daß $(1/4)\sum_i D^{(1)}(f_i)$ mit der üblichen Nečiporuk Funktion übereinstimmt und daher eine untere Schranke für deterministische Formellänge ist (vergleiche Fakt 2.19).

Theorem 3.5 *Die probabilistische Nečiporuk Funktion ist für die Länge von starken probabilistischen Formeln mit beschränktem Fehler eine untere Schranke.*

Beweis: Für jede Partition $y_1, \ldots, y_k$ der Eingaben zeigen wir, wie eine starke probabilistische Formel F in den k Kommunikationsspielen simuliert werden kann. Sei F_i die von F induzierte Verteilung auf denjenigen Subbäumen der deterministischen Formeln zu F, die als Blätter die Variablen in y_i enthalten und dazu alle Pfade von diesen Blättern bis zur Wurzel enthalten. Wir wollen die Formel in Spiel i so simulieren, daß die probabilistische Einweg Kommunikation durch die erwartete Anzahl der Blätter in F_i beschränkt ist.
Gegeben ist eine probabilistische Formel F. Die Spieler bestimmen nun eine deterministische Formel F' mit den öffentlichen Zufallsbits. Spieler A weiß alle Eingaben außer denen in y_i. Damit wird auch ein Subbaum F_i' fixiert. V_i enthalte die Knoten in F_i', welche 2 Vorgänger in F_i' haben und P_i alle Pfade, welche in V_i oder an einem Blatt anfangen, und in V_i oder an der Wurzel aufhören, aber keinen weiteren Knoten aus V_i enthalten. Es reicht aus, wenn A für jeden solchen Pfad 2 Bits sendet, die angeben, ob das letzte Gatter des Pfades $0, 1, g$, oder $\neg g$ berechnet, für die Funktion g des ersten Gatters des Pfades. Dann kann B die Formel alleine auswerten.
Es gibt aber höchstens $2|V_i| + 1$ solche Pfade, da der fan-in der Formel 2 ist. Damit ist die gesamte Kommunikation höchstens $4|V_i| + 2$. Die Menge der Blätter L_i mit Variablen aus y_i hat $|V_i| + 1$ Elemente, und daher ist $R^{(1,pub)}(f_i) \leq 4|V_i| + 2 < 4|L_i|$ und $1/4 \sum_i R^{(1,pub)}(f_i)$ ist eine untere Schranke für die Länge $E[\sum_i |L_i|] = \sum_i E[|L_i|]$ der probabilistischen Formel. □
Sei $VC(f_i)$ die VC-Dimension des Kommunikationsproblems f_i. Wir nennen $\sum_i VC(f_i)$ die VC-Nečiporuk Funktion.

Korollar 3.3 *Die VC-Nečiporuk Funktion ist eine asymptotische untere Schranke für die Länge starker probabilistischer Formeln mit beschränktem Fehler.*
Die konventionelle Nečiporuk Funktion ist eine asymptotische untere Schranke für die Länge von starken Las Vegas Formeln für totale Funktionen.

Beweis: Nach Fakt 3.2 ist die VC-Dimension eine asymptotische untere Schranke für die probabilistische Einweg Kommunikationskomplexität einer Funktion bei öffentlichem Zufall und beschränktem Fehler.
Nach Fakt 3.3 gilt, daß Las Vegas Einweg Protokolle für totale Funktionen höchstens einen konstanten Faktor effizienter sein können als optimale deterministische Einweg Protokolle. Das gilt ebenfalls bei öffentlichem Zufall. □

Nach Fakt 2.20 ist $\Theta(n^2/\log n)$ die deterministische Formellänge der Funktion Indirekte Speicher Adressierung (ISA) aus Definition 2.17. Wir wenden unsere Methode an, um eine untere Schranke derselben Größenordnung für starke probabilistische Formeln mit beschränktem Fehler zu zeigen. Daher ist ISA eine konkrete Funktion, bei der starker Probabilismus die Formellänge nicht wesentlich reduzieren kann.

Korollar 3.4 *Jede starke probabilistische Formel für die ISA Funktion (mit beschränktem Fehler) hat eine Länge von mindestens* $\Omega(n^2/\log n)$.

BEWEIS: ISA hat Eingaben Y, X, U und berechnet Y_{X_U}, siehe Definition 2.17. Zuerst definieren wir die Partition. Wir teilen die Eingaben in X in $n/\log n$ Blöcke mit jeweils $\log n$ Bits auf, alle anderen Eingaben kommen in einen Block. In einem der Kommunikationsspiele erhält A alle Eingaben außer einem Block von X. S sei die Menge der möglichen Werte der Variablen in diesem Block. Diese Menge wird zerschmettert: Sei $R \subseteq S$ und $R = \{r_1, \ldots, r_m\}$. Man läßt nun den Zeiger U auf den Block in der Eingabe von B zeigen, und setzt dann $Y_i = 1 \iff i \in R$.
Also ist die VC-Dimension von f_i genau $|S| = n$. Da es $n/\log n$ solche Spiele gibt, folgt das Ergebnis. □
Die nächste Aussage wäre trivial für deterministische oder faire probabilistische Formeln, aber für starke probabilistische Formeln können Funktionen, die von allen Eingaben abhängen, sublineare Komplexität haben. Die approximative Majoritätsfunktion ist 0, wenn die Eingabe weniger als $n/3$ Einsen enthält, und 1, wenn die Eingabe mehr als $(2/3)n$ Einsen enthält, ansonsten undefiniert. Diese partielle Funktion kann von starken probabilistischen Formeln der Länge 1 berechnet werden, indem man zufällig eine Variable zieht. Für totale Funktionen ergibt sich aber:

Korollar 3.5 *Jede starke probabilistische Formel, welche eine totale Funktion berechnet, die von* n *Variablen abhängt, hat Länge* $\Omega(n)$.

BEWEIS: Wir teilen die Eingaben in n Blöcke von je einer Variable auf, von denen A $n-1$ und B einen erhält. Da die Funktion von der Eingabe von B abhängt, ist die deterministische Kommunikation mindestens 1. Wäre die probabilistische Kommunikation 0, so hätte das Protokoll für eine Eingabe Fehler $1/2$ und wäre nicht korrekt. □
Fakt 2.33 zeigt, daß für eine Funktion $f : X \times Y \to \{0,1\}$ gilt, daß $D^{(1)}(f) \leq \lceil VC(f) \cdot \log(|Y|+1) \rceil$. Das führt zu:

Theorem 3.6 *Für alle totalen Funktionen* $f : \{0,1\}^n \to \{0,1\}$ *mit einer starken probabilistischen Formel der Länge* s *und für alle Partitionen der Eingaben von* f *gilt:*

$$\frac{\sum D^{(1)}(f_i)}{s} = O(\sqrt{n}).$$

BEWEIS: Offensichtlich ist $D^{(1)}(f_i) \leq n$ für alle i. Das es in einer Partition höchstens $\sqrt{n}$ Blöcke mit mehr als $\sqrt{n}$ Variablen gibt, tragen diese zur Nečiporuk Funktion $\sum D^{(1)}(f_i)$ höchstens $n\sqrt{n}$ bei. Alle kleineren Blöcke erfüllen $D^{(1)}(f_i) \leq \lceil \sqrt{n} \cdot VC(f_i) \rceil$. Also gilt insgesamt $\sum D^{(1)}(f_i) \leq O(\sqrt{n}(n + \sum VC(f_i))) = O(\sqrt{n}s)$. □
Wenn eine totale Funktion effiziente probabilistische Formeln besitzt (mit linearer Länge), ergibt die Nečiporuk Methode keine quadratische untere Schranke.

Eine Funktion, für die Monte Carlo Probabilismus hilft

Wir betrachten nun eine Funktion, für die Monte Carlo Probabilismus so viel hilft, wie nach obigen Betrachtungen mit der Nečiporuk Methode maximal nachweisbar ist. Wir erhalten einen solchen Unterschied sogar für faire probabilistische Formeln.

Definition 3.3 *Die Matrixprodukt Funktion MP erhält zwei $n \times n$-Matrizen $T^{(1)}, T^{(2)}$ über $\mathbb{Z}_2$ als Eingaben und akzeptiert genau dann, wenn deren Produkt nicht die Nullmatrix ist.*

Theorem 3.7 *Die MP Funktion kann von einer fairen Monte Carlo Formel der Länge $O(n^2)$ berechnet werden.*

BEWEIS: Wir bestimmen zuerst mit Hilfe einiger Zufallsvariablen einen Vektor als Fingerabdruck jeder Matrix. Nachher multiplizieren wir die Fingerabdrücke und erhalten ein Bit. Dieses ist immer 0, wenn das Produkt der Matrizen die Nullmatrix ist, ansonsten 1 mit Wahrscheinlichkeit 1/4. Wir erhalten also eine Monte-Carlo Formel.
Es seien $r^{(1)}, r^{(2)}$ Zufallsstrings aus n Bits. Die Fingerabdrücke sind definiert als $F^{(1)}[i] = \bigoplus_{l=1}^{n} r^{(1)}[l]T^{(1)}[l,i]$ und $F^{(2)}[i] = \bigoplus_{l=1}^{n} T^{(2)}[i,l]r^{(2)}[l]$. Dann sei $b = \bigoplus_{i=1}^{n} F^{(1)}[i] \wedge F^{(2)}[i]$. Offenbar kann b von einer Formel mit linearer Länge berechnet werden.
Angenommen $T^{(1)}T^{(2)} = 0$. Dann gilt $r^{(1)}T^{(1)}T^{(2)}r^{(2)} = 0$ für alle $r^{(1)}$ und $r^{(2)}$.
Wenn aber $T^{(1)}T^{(2)} \neq 0$ ist, dann gibt es i, j so daß $\bigoplus_k T^{(1)}[i,k]T^{(2)}[k,j] = 1$. Seien alle Zufallsbits außer $r^{(1)}[i]$ und $r^{(2)}[j]$ beliebig fixiert. Gleichgültig, wie die Werte der anderen Summen aussehen, ergibt ein Wert von $r^{(1)}[i]$ und $r^{(2)}[j]$ das Ergebnis $b = 1$, das geschieht zufällig mit Wahrscheinlichkeit 1/4. □

Theorem 3.8 *Für die MP Funktion gilt eine untere Schranke von $\Omega(n^3)$ für die Länge starker Las Vegas Formeln.*

BEWEIS: Wir benutzen die Nečiporuk Methode. Zunächst zur Definition der Partition der Eingaben. Es werden n Blöcke b_j mit den Bits $T^{(2)}(i,j)$ für alle

$i = 1, \ldots, n$ sowie ein Block für die restlichen Eingaben benutzt. Also erhalte A alle Bits außer den n Bits der $j$ten Spalte $T^{(2)}(\cdot, j)$ von $T^{(2)}$, welche B erhalte. Wir zeigen, daß MP nun eine Einweg Kommunikationskomplexität von $\Omega(n^2)$ hat. Die Nečiporuk Methode gibt uns damit eine untere Schranke von $\Omega(n^3)$ für die Länge von deterministischen und starken Las Vegas Formeln. O.B.d.A. besitze B die Bits $T^{(2)}(i, 1)$.
Wir konstruieren eine Menge von Werten der Eingabevariablen von A. Sei U ein Unterraum von $\mathbb{Z}_2^n$ und T_U eine Matrix mit $T_U x = 0 \iff x \in U$. Für jedes U wählen wir T_U als $T^{(1)}$ und $T^{(2)}(i, j) = 0$ für alle i und für $j \geq 2$. Wenn es nun $2^{\Omega(n^2)}$ paarweise verschiedene Unterräume gibt, so erhalten wir so viele Werte der Eingabevariablen. Aber jeder dieser Werte entspricht einer eigenen Zeile der Kommunikationsmatrix, denn alle $T^{(1)}$ haben verschiedene Kerne. Daher ist die deterministische Einweg Kommunikation $\Omega(n^2)$.
Um zu sehen, daß es tatsächlich $2^{\Omega(n^2)}$ paarweise verschiedene Unterräume von $\mathbb{Z}_2^n$ gibt zählen wir die Unterräume mit Dimension kleiner gleich $n/2$. Es gibt 2^n Vektoren. Es gibt $\binom{2^n}{n/2}$ Möglichkeiten, $n/2$ paarweise verschiedene Mengen von $n/2$ Vektoren zu ziehen. Jede solche Menge ergibt einen Unterraum mit Dimension kleiner gleich $n/2$. Jeder solche Unterraum wird von höchstens $\binom{2^{n/2}}{n/2}$ Mengen von $n/2$ paarweise verschiedenen Vektoren aus dem Unterraum aufgespannt. Daher wird jeder Unterraum höchstens so oft gezählt und es gibt mindestens

$$\frac{\binom{2^n}{n/2}}{\binom{2^{n/2}}{n/2}} \geq 2^{\Omega(n^2)}$$

paarweise verschiedene Unterräume von $\mathbb{Z}_2^n$. □

Korollar 3.6 *Es gibt eine Funktion, die von fairen Monte Carlo Formeln der Länge $O(N)$ berechnet wird, für die aber jede starke Las Vegas Formel Länge $\Omega(N^{3/2})$ hat, d.h. es gibt einen Unterschied von $\Omega(N^{1/2})$ zwischen Las Vegas und Monte Carlo Formellänge.*
Es gibt ebenfalls einen Unterschied von $\Omega(N^{1/2})$ zwischen Monte Carlo Formellänge und der Länge von probabilistischen Formeln mit beschränktem Fehler.

Beweis: Für den zweiten Teil betrachten wir die folgende Funktion mit vier Matrizen als Eingaben. Es soll die Parität von der MP Funktion auf den ersten beiden Matrizen und dem Komplement von MP auf den anderen Matrizen berechnet werden.
Eine faire probabilistische Formel kann das Problem offensichtlich mit Länge $O(n^2)$ lösen. Hat man aber eine Monte Carlo Formel, so fixiert man die ersten zwei Eingabematrizen einmal, so daß ihr Produkt 0 ist, und einmal, so daß dies nicht der Fall ist. Man erhält so Monte Carlo Formeln für MP und das Komplement von MP. Läßt man beide auf derselben Eingabe rechnen und

kombiniert das Ergebnis geeignet, so erhält man eine Las Vegas Formel und damit die untere Schranke.
Für die Konstruktion einer Las Vegas Formel sei F die Monte Carlo Formel für MP und G die Monte Carlo Formel für $\neg MP$. Dann sind F und $\neg G$ Formeln für MP, so daß F niemals fehlerhaft akzeptiert, aber mit Wahrscheinlichkeit $1/2$ korrekt ist und $\neg G$ niemals fehlerhaft verwirft, aber mit Wahrscheinlichkeit $1/2$ korrekt ist. Angenommen der Funktionswert ist 0. Dann wird F verwerfen. Mit Wahrscheinlichkeit $1/2$ wird auch $\neg G$ verwerfen, ansonsten wird aufgegeben. Angenommen der Funktionswert ist 1. Dann wird $\neg G$ akzeptieren. Mit Wahrscheinlichkeit $1/2$ wird auch F akzeptieren, ansonsten wird aufgegeben. Also wird immer mit Wahrscheinlichkeit $1/2$ korrekt gerechnet, sonst aufgegeben. □
Eine interessante Eigenschaft der Formel aus dem Beweis von Theorem 3.7 ist, daß sie jede Eingabe genau einmal liest, allerdings ihre Zufallseingaben mehrmals. MP kann nicht von einer deterministischen Formel berechnet werden, die ihre Eingaben nur je einmal liest, denn eine solche Formel hätte lineare Größe. In Abschnitt 5.5.3 werden wir sehen, daß eine faire probabilistische Formel, die ihre Zufallseingaben nur je einmal liest, durch die Nečiporuk Funktion geteilt durch $\log n$ in der Länge beschränkt ist. Daher sind für MP Zufallseingaben, welche nur einmal gelesen werden dürfen, praktisch wirkungslos.

Verallgemeinerte Matrix Multiplikation

Wir haben gesehen, daß wir bessere Schranken als in Korollar 3.6 nicht mit der Nečiporuk Methode nachweisen können. Im folgenden suchen wir nach einer Funktion, die ein Kandidat für einen größeren Unterschied zwischen deterministischer und probabilistischer Formellänge ist.

Definition 3.4 *Das Boolesche verallgemeinerte Matrix Produkt ist definiert auf k-dimensionalen $n \times \cdots \times n$ Matrizen $T^{(1)}, \ldots, T^{(k)}$:*
$VP(T^{(1)}, \ldots, T^{(k)})[i_1, \ldots, i_k]$
$= \bigoplus_{j_1,\ldots,j_{k-1}} (T^{(1)}[j_1, \ldots, j_{k-1}, i_1] \wedge \cdots \wedge T^{(k)}[j_1, \ldots, j_{k-1}, i_k])$.
Die verallgemeinerte Matrix Produkt Ungleichheitsfunktion (VMPU) hat als Eingaben k-dimensionale $n \times \cdots \times n$ Matrizen $T^{(i)}$ und E über $\mathbb{Z}_2$ und entscheidet, ob das Produkt der $T^{(i)}$ ungleich E ist:

$$\bigvee_{i_1,\ldots,i_k} E[i_1, \ldots, i_k] \oplus VP(T^{(1)}, \ldots, T^{(k)})[i_1, \ldots, i_k].$$

Theorem 3.9 *Die $VMPU$ Funktion kann von einer fairen Monte Carlo Formel der Länge $O(2^k \cdot k \cdot n^k)$ berechnet werden.*

BEWEIS: Wieder nehmen wir von jeder verallgemeinerten Matrix einen Fingerabdruck der Dimension $k-1$, multiplizieren diese und erhalten ein Bit, welches einen Ungleichheitstest mit Wahrscheinlichkeit $1/2^k$ erlaubt.
Sei $T^{(i)}$ die $i$te verallgemeinerte Matrix und $r_1^{(i)}, \ldots, r_n^{(i)}$ seien Zufallsbits (insgesamt werden kn Zufallsbits verwendet). Dann ist der Fingerabdruck $F^{(i)}$ definiert als
$F^{(i)}[j_1, \ldots, j_{k-1}] = \bigoplus_{l=1}^{n} r_l^{(i)} T^{(i)}[j_1, \ldots, j_{k-1}, l]$.
Es sei $b = \bigoplus_{j_1,\ldots,j_{k-1}} (F^{(1)}[j_1, \ldots, j_{k-1}] \wedge \cdots \wedge F^{(k)}[j_1, \ldots, j_{k-1}])$.
Dann ist b nach Auflösen mit dem Distributivgesetz gleich

$$\bigoplus_{i_1,\ldots,i_k} r_{i_1}^{(1)} \wedge \cdots \wedge r_{i_k}^{(k)} \wedge \bigoplus_{j_1,\ldots,j_{k-1}} (T^{(1)}[j_1, \ldots, j_{k-1}, i_1] \wedge \cdots \wedge T^{(k)}[j_1, \ldots, j_{k-1}, i_k]).$$

Als Ausgabe wird $b \oplus e$ verwendet mit $e = \bigoplus_{i_1,\ldots,i_k} r_{i_1}^{(1)} \wedge \cdots \wedge r_{i_k}^{(k)} \wedge E[i_1, \ldots, i_k]$
Angenommen $VP(T^{(1)}, \ldots, T^{(k)}) = E$. Dann ist
$\bigoplus_{j_1,\ldots,j_{k-1}} (T^{(1)}[j_1, \ldots, j_{k-1}, i_1] \wedge \cdots \wedge T^{(k)}[j_1, \ldots, j_{k-1}, i_k]) = E[i_1, \ldots, i_k]$
für alle $i_1, \ldots, i_k$. Seien die $r_l^{(i)}$ beliebige Bits. Es gilt $b = e$.
Angenommen $VP(T^{(1)}, \ldots, T^{(k)}) \neq E$. In diesem Fall gibt es $i_1, \ldots, i_k$ so daß

$$\bigoplus_{j_1,\ldots,j_{k-1}} (T^{(1)}[j_1, \ldots, j_{k-1}, i_1] \wedge \cdots \wedge T^{(k)}[j_1, \ldots, j_{k-1}, i_k]) \neq E[i_1, \ldots, i_k]. \tag{3.1}$$

Seien alle $r_l^{(j)}$ außer den $r_{i_j}^{(j)}$ beliebig fixiert. Wir zeigen, daß mindestens ein Wert der restlichen k Zufallsbits zu $b \neq e$ führt. Wir sagen, daß eine Summe $S_{l_1,\ldots,l_k} = \bigoplus_{j_1,\ldots,j_{k-1}} (T^{(1)}[j_1, \ldots, j_{k-1}, l_1] \wedge \cdots \wedge T^{(k)}[j_1, \ldots, j_{k-1}, l_k])$ von einem Wert der restlichen Zufallsbits *erzeugt* wird, wenn $S_{l_1,\ldots,l_k} \neq E[l_1, \ldots, l_k]$ und die Zufallsbits $r_{l_i}^{(i)}$ alle den Wert 1 haben (einige dieser Zufallsbits sind eventuell schon fixiert). Jede Summe außer (3.1) wird von einer geraden Anzahl von Werten erzeugt, (3.1) wird von genau einem Wert erzeugt. Also muß es einen Wert geben, der eine ungerade Anzahl von Summen erzeugt, und daher zu $b \neq e$ führt.
Es ist klar, daß $b \oplus e$ von einer Formel der Länge $O(kn^k)$ berechnet werden kann. Um Fehlerwahrscheinlichkeit $1/2$ zu erreichen, verwenden wir 2^k unabhängige Versuche und verbinden sie durch einen ODER Baum. □
Wir haben keine gute untere Schranke für die Länge deterministischer Formeln für $VMPU$. Daher betrachten wir eine leicht unterschiedliche Funktion.

Definition 3.5 *Die verallgemeinerte Matrix Produkt Nichtdisjunktheits Funktion (VMPND) hat als Eingaben k-dimensionale $n \times \cdots \times n$ Matrizen $T^{(i)}$ und E über $\mathbb{Z}_2$ und entscheidet, ob*

$$\bigvee_{i_1,\ldots,i_k} E[i_1, \ldots, i_k] \wedge VP(T^{(1)}, \ldots, T^{(k)})[i_1, \ldots, i_k]$$

Theorem 3.10 *$VMPND$ hat starke Las Vegas Formeln der Länge $\Omega(n^{2k-1})$. Jede starke probabilistische Formel mit beschränktem Fehler für $VMPND$ hat Länge $\Omega(n^{2k-2}/k)$.*

Beweis: Der erste Teil kann mit der Nečiporuk Methode bei einer Partition mit Blocklänge kn gezeigt werden. Dann sind aber Formeln mit beschränktem Fehler ebenfalls ineffizient: da die VC-Dimension höchstens einen Faktor kn kleiner als die deterministische Einweg Kommunikation sein kann, folgt Teil 2 mit Fakt 2.33.
Nun zu der unteren Schranke. Spieler B besitzt im Kommunikationsspiel $(j_1, \ldots, j_{k-1})$ für alle p und l die Eingaben $T^{(p)}[j_1, \ldots, j_{k-1}, l]$, Spieler A hat alle restlichen Eingaben. Somit gibt es n^{k-1} Spiele. Der Einfachheit halber nehmen wir an, daß wir das Spiel $(j_1, \ldots, j_{k-1}) = (1, \ldots, 1)$ spielen.
Wir betrachten die Menge der Werte der Eingaben von A, wenn E beliebig ist und alle anderen Eingaben 0. A muß für jede solche Eingabe eine andere Nachricht senden: zwei verallgemeinerte Matrizen $E^{(1)}$ und $E^{(2)}$ mögen sich an der Position $(i_1, \ldots, i_k)$ unterscheiden. Werden die Eingaben von B genau an den Positionen $T^{(l)}[1, \ldots, 1, i_l]$ auf 1 gesetzt und sonst auf Null, so erhalten wir zwei verschiedene Ausgaben. Damit ist die deterministische Einweg Kommunikation mindestens n^k und die Formellänge mindestens $\Omega(n^{2k-1})$.
□
Wir vermuten, daß deterministische Formeln für $VMPU$ nicht kürzer sind als solche für $VMPND$. In der Kommunikationskomplexität ist das Disjunktheitsproblem schwer für Probabilismus, während Ungleichheit einfach ist. Deterministisch sind jedoch beide schwer. Wenn also die Intuition aus der Kommunikationstheorie richtig ist, so sollte $VMPU$ für beliebig großes, aber konstantes k ein Kandidat sein, um einen fast quadratischen Unterschied zwischen deterministischer und probabilistischer Formellänge zu zeigen. Ein solcher Beweis würde jedoch wegen Theorem 3.6 neue Techniken erfordern.

Kapitel 4

Nichtdeterministische Kommunikation

4.1 Überblick

Einige der fundamentalen offenen Fragen der Komplexitätstheorie betreffen die Mächtigkeit von Nichtdeterminismus im Vergleich zu Determinismus. Während in den Modellen der Turingmaschinen und der Registermaschinen fast keine Resultate bekannt sind, die zeigen, daß Nichtdeterminismus tatsächlich hilft (außer z.B. [PST83] und [Aj99]), ist Nichtdeterminismus in der Theorie der Kommunikationskomplexität wesentlich besser verstanden. Das Thema des beschränkten Nichtdeterminismus ist intensiv erforscht worden, siehe [GLM96] für einen Überblick. Man betrachtet Nichtdeterminismus als eine beschränkt vorhandene Resource. Einer nichtdeterministischen Berechnung stehen hier nur „wenige“ nichtdeterministische Entscheidungen zur Verfügung. Andere Einschränkungen an nichtdeterministische Berechnungen, wie beschränkte Ambiguität [KNSW94, GLW92] wollen wir hier nicht untersuchen.
Die Forschung im Bereich des beschränkten Nichtdeterminismus' konzentriert sich auf die Bereiche Turingmaschinen, Schaltkreise und endliche Automaten. Eine generelle Definition von beschränktem Nichtdeterminismus befindet sich in [CC97]. Für Modelle wie beschränkt nichtdeterministische Turingmaschinen oder Schaltkreise kennt man jedoch bisher keine unteren Schranken für konkrete Funktionen.
In diesem Kapitel geben wir untere Schranken für explizit gegebene Funktionen in verschiedenen Modellen von Berechnungen mit beschränktem Nichtdeterminismus an. Die beschränkt langen nichtdeterministischen Rateworte kann man auch als beschränkt lange Beweise verstehen, die manchmal zu erhöhtem Verifizierungsaufwand führen.
Die Untersuchung von beschränktem Nichtdeterminismus in der Kommunikationskomplexität wurde initiiert in [HrS96]. Während probabilistische

Protokolle nur relativ wenige Ratebits benötigen, nämlich $\log n + O(1)$ (siehe Fakt 2.4), kann eine scheinbar geringfügige Einschränkung der Anzahl der nichtdeterministischen Ratebits die Kommunikation drastisch erhöhen, siehe Fakt 2.14.
Eine wichtige Eigenschaft der nichtdeterministischen Kommunikation ist die Tatsache, daß Protokolle mit nur einer Runde optimal sind. Das ist ein scharfer Kontrast zu deterministischen und probabilistischen Protokollen (siehe Fakt 2.1 und Abschnitt 3.2). Das zentrale Resultat dieses Kapitels (Theorem 4.2) zeigt, daß Kommunikation mit beschränktem Nichtdeterminismus sehr stark von Interaktion abhängt. Diese Eigenschaft ist der gemeinsame Ursprung der Resultate in diesem Kapitel.
Wir beginnen wieder mit Einweg Kommunikation. Ein fast optimaler Unterschied zwischen beschränkt nichtdeterministischer Einweg Kommunikation und unbeschränkt nichtdeterministischer Kommunikation ist das Ergebnis von Theorem 4.1.

Theorem 4.1 *Es gibt eine Funktion $OD_{n,s}$ auf $\Theta(ns \log n)$ Eingaben, so daß folgendes gilt:*

$$N^{(1)}_{O(s \log n)}(OD_{n,s}) = O(s \log n).$$

$$D^{(B,1)}(OD_{n,s}) = O(s \log n).$$

Es gibt eine Konstante $\epsilon > 0$ so daß für $s \leq n$ gilt

$$N^{(1)}_{\epsilon s}(OD_{n,s}) = \Omega(ns \log n).$$

Eine wichtige Eigenschaft der Funktion zu diesem Resultat ist eine asymmetrische Aufteilung der Eingaben, in der A weitaus mehr Eingaben als B erhält. Das wird für eine Anwendung auf Formellänge wichtig sein.
Dann wenden wir uns der Kommunikation mit mehr Runden zu. Eine Rundenhierarchie für Kommunikation mit beschränktem Nichtdeterminismus ist in [HrS96] zu finden, aber nur für eine weniger als logarithmische Anzahl von Ratebits. Wir geben für alle s, k eine Funktion auf n Eingaben an, die deterministisch in k Runden mit Kommunikation $O(sk \log n)$ berechnet werden kann, wenn A beginnt, während jedes Protokoll mit k Runden, bei dem B startet und welches s nichtdeterministische Ratebits verwendet, Kommunikation $\Omega(n/(s^2k^2 \log n))$ benötigt.

Theorem 4.2 *Für alle s, k und alle n gibt es eine Boolesche Funktion f^s_k mit n Eingaben und*

- $D^{(k)}(f^s_k) = O(sk \log n)$
- $N^{(B,k)}_s(f^s_k) = \Omega(n/(s^2k^2 \log n))$
- $N^{(B,k)}_{O(s \log n)}(f^s_k) = O(sk \log n)$.

Die Funktion ist eine Variante von Pointer Jumping, bei der man $\Theta(s)$ Pfaden in einem bipartiten Graphen bis zum $k+2$ten Knoten folgen muß. Man kann leicht eine Runde sparen, indem man für jeden Pfad eine Kante mit $\log n$ nichtdeterministischen Bits rät. Sind nun z.B. s, k polylogarithmisch in n, so ergibt sich ein exponentieller Unterschied zwischen der beschränkt nichtdeterministischen $k-1$-Runden Kommunikationskomplexität und der deterministischen k-Runden Kommunikationskomplexität.
Abschnitt 4.3 enthält Anwendungen der Resultate zur Kommunikationskomplexität. Zuerst zeigen wir, daß die uniforme nichtdeterministische Kommunikationskomplexität nicht immer eine gute untere Schranke für die Größe nichtdeterministischer Einweg Automaten ist. Dann zeigen wir, daß eine Sprachversion von $D_{n,n}$ (siehe Definition 2.7) eine bestimmte konstante Anzahl von Ratebits zur Konstruktion effizienter nichtdeterministischer Einweg Automaten erfordert. Genauer, bis zu einem konstanten Wert (der polynomiell in der optimalen nfa-Größe ist) bleibt die Größe eines nfa nahe an der minimalen dfa Größe, um dann schnell exponentiell in die Nähe der optimalen nfa Größe zu fallen, wenn die Anzahl der Ratebits um einen logarithmischen Faktor steigt. Dieses Resultat ist eine Ergänzung zu den Resultaten in [GKW90], wo gezeigt wird, daß manchmal nur unbeschränkt viele Ratebits helfen, und manchmal die Größe kontinuierlich mit jedem zusätzlichen Ratebit sinkt, bis das Optimum (fast) erreicht ist.

Theorem 4.3 *Es gibt eine Sprache $D_N \subseteq \{0,1\}^N$, so daß D_N von einem nfa mit $O(\sqrt{N \log N})$ Ratebits und poly(N) Zuständen erkannt werden kann, während das Erkennen von D_N mindestens $2^{\Omega(\sqrt{N/\log N})}$ Zustände erfordert, wenn ein nfa nur $\epsilon\sqrt{N/\log N}$ Ratebits benutzt für eine hinreichend kleine Konstante $\epsilon > 0$. Der minimale dfa für D_N hat die Größe $2^{\Theta(\sqrt{N \log N})}$.*

Wir betrachten dann nichtdeterministische Zweiweg Automaten. Es gibt erstaunlicherweise eine Hierarchie über die maximal erlaubte Länge der Kreuzungsfolgen k (oder die Anzahl der Reversals), in der die Größe nichtdeterministischer Automaten (mit unbeschränktem Nichtdeterminismus) bei der Restriktion von k ansteigt. Für Automaten mit beschränktem Nichtdeterminismus ist die entsprechende Hierarchie allerdings viel stärker ausgeprägt und ergibt exponentielle Unterschiede. Reversal Komplexität für Zweiweg nfa wird auch z.B. in [Hr91] betrachtet. Für Turing Maschinen enthält das Reversalmaß gewöhnlich auch Reversals auf den Arbeitsbändern, siehe z.B. [CY91]. Andere Einschränkungen der Bewegungen von Zweiweg Automaten werden z.B. in [Da96] untersucht.
Die maximalen Unterschiede zwischen den Größen nichtdeterministischer Automaten mit verschiedenen Kreuzungsfolgenlängen sind wie folgt beschränkt. Ein k-kreuzungsbeschränkter nichtdeterministischer Automat mit q Zuständen kann von einem Einweg nfa mit $O(q^k)$ Zuständen simuliert werden. Ein k-reversalbeschränkter nichtdeterministischer Automat mit q Zu-

ständen kann von einem $k-j$-reversalbeschränkten nichtdeterministischen Automaten mit $O(q^{j+2})$ Zuständen simuliert werden, siehe Lemma 4.7.
Die Hierarchie für unbeschränkten Nichtdeterminismus ist wie folgt.

Theorem 4.4 *Es gibt eine von k unabhängige Sprache $L \subseteq \{0,1\}^n$, so daß jeder nichtdeterministische k-kreuzungsbeschränkte Automat für das Erkennen von L mindestens $\Omega(N^{1/k})$ Zustände benötigt. Ein deterministischer k-reversalbeschränkter Automat mit $O(N^{1/k}\log^2 N)$ Zuständen kann L erkennen. $N = \Theta(2^{n/2})$ ist die Größe des minimalen Einweg dfa für L.*

Bei beschränktem Nichtdeterminismus ergibt sich eine stärkere Hierarchie:

Theorem 4.5 *Für alle s, k, n mit $n \geq s$ gibt es eine Sprache $L_{s,k} \subseteq \{0,1\}^n$, welche von einem deterministischen k-reversalbeschränkten Automaten mit $kn^{O(s)}$ Zuständen erkannt wird. Zum Erkennen von $L_{s,k}$ benötigt jeder nichtdeterministische $(k-1)$-kreuzungsbeschränkte Automat mit nur s nichtdeterministischen Ratebits $2^{\Omega(n/(s^2k^3\log n))}$ Zustände.*

In Abschnitt 3.3.3 haben wir die Nečiporuk Methode mit Hilfe von asymmetrischer Einweg Kommunikation beschrieben. In Abschnitt 4.3.3 leiten wir eine untere Schranke für die Länge von Formeln mit beschränktem Nichtdeterminismus her. Dazu entwerfen wir eine geignete Variante des iterierten Disjunktheitsproblems.

Theorem 4.6 *Es gibt eine Boolesche Funktion $AD_{n,s}$ (mit $\log n \leq s \leq n$) auf $N = O(ns\log n)$ Eingaben, so daß jede Formel mit s nichtdeterministischen Bits für $AD_{n,s}$ eine Länge von mindestens $\Omega(n^2 s\log n)$ hat. $AD_{n,s}$ kann von einer Formel mit $O(s\log n)$ nichtdeterministischen Bits und Länge $O(ns^2\log n)$ berechnet werden.*

Insbesondere sei für $0 < \epsilon \leq 1/2$ $s = n^{\frac{\epsilon}{1-\epsilon}}$, dann ist die untere Schranke $\Omega(N^{2-\epsilon}/\log^{1-\epsilon} N)$ bei $N^\epsilon/\log^\epsilon N$ erlaubten Ratebits. $O(N^\epsilon \log^{1-\epsilon} N)$ Ratebits reichen, um eine Länge von $O(N^{1+\epsilon}/\log^\epsilon N)$ zu erzielen.
Wegen Beschränkungen des Nečiporuk Ansatzes kann man nur untere Schranken von bestenfalls n^2/s für s Ratebits zeigen, wesentlich bessere Ergebnisse benötigten also neue Techniken. Im Fall von unbeschränktem Nichtdeterminismus wird die Asymmetrie der Eingabeaufteilung unwichtig und nur triviale untere Schranken sind mit unserer Methode möglich. Das ist kein Zufall, da allgemeine nichtdeterministische Formeln so mächtig sind wie nichtdeterministische Schaltkreise. Aber selbst für deterministische Schaltkreise sind für konkrete Funktionen keine superlinearen unteren Schranken bekannt.
Schließlich untersuchen wir die Tiefe monotoner Schaltkreise. Karchmer und Wigderson haben gezeigt, daß eine Äquivalenz zwischen der Kommunikationskomplexität eines bestimmten Spiels und der Tiefe monotoner Schaltkreise besteht [KW90]. Diese Äquivalenz kann auch über Tiefe (bei unbeschränktem fan-in) und Logarithmus des fan-in gegen Runden und Kommunikation pro Runde dargestellt werden, siehe [NW93] und [K88].

Es ist einfach zu sehen, daß Tiefe bis zum Logarithmus der Größe reduziert werden kann, wenn unbeschränkter Nichtdeterminismus vorhanden ist, dabei steigt die Größe nicht wesentlich an. Wir zeigen, daß dies auch im Fall monotoner Schaltkreise gilt.

Theorem 4.7 *Ein monotoner nichtdeterministischer Schaltkreis mit c Gattern kann in eine äquivalente monotone nichtdeterministische Formel mit Tiefe $\log c + O(1)$ und $O(c)$ Gattern umgewandelt werden. Bei unbeschränktem fan-in reicht Tiefe 2 bei Größe $O(c)$.*

Wir konstruieren eine Familie von Funktionen auf n Variablen für jedes d (mit $\sqrt{n} \geq d \geq \log n$) so daß jede Funktion in monotoner Tiefe $\Theta(d)$ berechenbar ist, wenn höchstens $\epsilon n/d$ nichtdeterministische Bits für ein konstantes $\epsilon > 0$ erlaubt sind. Sind n/d nichtdeterministische Bits vorhanden, kann die monotone Tiefe auf $O(\log n)$ gesenkt werden. Die untere Schranke benutzt die untere Schranke für das Matching Problem aus [RW92].

Theorem 4.8 *Seien d, n so, daß $\sqrt{n} \geq d \geq \log n$. Dann gibt es eine explizite Boolesche Funktion g_n^d auf n Variablen sowie eine Konstante $\epsilon > 0$ mit den folgenden Eigenschaften: g_n^d kann von einer monotonen deterministischen Formel mit Tiefe $O(d)$ berechnet werden, und jeder nichtdeterministische monotone Schaltkreis mit $\epsilon n/d$ Ratebits braucht Tiefe $\Omega(d)$. g_n^d kann in monotoner Tiefe $O(\log n)$ mit n/d Ratebits berechnet werden.*

s,t-connectivity ist ein wichtiges monotones Problem, bei dem zu entscheiden ist, ob ein gegebener Graph mit zwei ausgezeichneten Knoten s und t einen Pfad von s nach t besitzt. Die deterministische monotone Tiefe des Problems ist $\Theta(\log^2 n)$ (siehe [KW90]). Intuitiv kann man vermuten, daß viel Information über den Pfad geraten werden muß, um die Tiefe drastisch zu reduzieren. Wir bestätigen dieses.

Theorem 4.9 *Sei d die Tiefe optimaler monotoner Schaltkreise für s,t-connectivity mit $(n/k)\log n$ nichtdeterministischen Ratebits.*
Dann ist $\Omega(\log^2 k + \log n) = d = O(\log n \log k)$.

Wir vermuten, daß die „wirkliche“ Schranke $\Theta(\log n \log k)$ ist.
Nisan und Wigderson haben eine Separation der Tiefenhierarchie monotoner Schaltkreise mit konstanter Tiefe und unbeschränktem fan-in bei polynomieller Größe, welche zuerst in [KPY84] bewiesen wurde, von der deterministischen Rundenhierarchie der Kommunikationskomplexität abgeleitet [NW93]. Wir verallgemeinern das Ergebnis für den Fall von beschränktem Nichtdeterminismus.

Theorem 4.10 *Für alle $k \geq 3$ und $s \geq n$ gibt es eine Funktion q_k^s mit $N = \Theta(sn^{k-1})$ Eingaben, die von deterministischen monotonen Formeln mit fan-in $O(s)$, Tiefe k und Größe $O(N)$ berechnet werden kann.*

Jeder monotone Schaltkreis für q_k^s mit Tiefe $k-1$, unbeschränktem fan-in und s/k nichtdeterministischen Bits hat Größe $2^{\Omega((N/s)^{1/(k-1)}/k)}$.
q_k^s kann von monotonen Schaltkreisen mit unbeschränktem fan-in, $O(s \log n)$ nichtdeterministischen Bits, Tiefe $k-1$ und Größe $O(N)$ berechnet werden. Der Fall $k = 3$ ist eine Ausnahme mit der Größe $O(N \log N)$.

Nichtdeterminismus erlaubt Vereinfachungen der Struktur von Berechnungen in vielen Berechnungsmodellen. Die Schlußfolgerung unserer Resultate ist, daß man dafür mit einem hohem Verbrauch von Nichtdeterminismus rechnen muß. Ist diese Resource beschränkt, so treten Runden- und Tiefenphänomene wie in der deterministischen Welt auf. Diese Phänomene tauchen in unseren Beispielen deshalb auf, weil Nichtdeterminismus Kommunikationsstrukturen vereinfachen kann, aber nur, wenn in ausreichendem Maße Ratebits vorhanden ist.
Die Ergebnisse dieses Kapitels sind zum großen Teil in [Kl98] und (Lemma 4.6) [HKKSS00] publiziert.

4.2 Resultate zur Kommunikationskomplexität

In gewisser Hinsicht ist alles über nichtdeterministische Kommunikation mit beschränkter Interaktion bereits in Fakt 2.1 gesagt: Eine Kommunikationsrunde ist ausreichend, man benötigt keine Interaktion. Aber dies wird mit einem hohen Verbrauch an nichtdeterministischen Bits erkauft und ist nicht mehr wahr, wenn nur noch beschränkter Nichtdeterminismus zur Verfügung steht. In diesem Fall besteht eine starke Abhängigkeit der Kommunikation von der Anzahl der Runden (je nachdem, wie viele Ratebits vorhanden sind).

4.2.1 Eine Funktion mit asymmetrischer Eingabeaufteilung und hoher Einweg Kommunikation

In diesem Abschnitt untersuchen wir nichtdeterministische Einweg Kommunikationskomplexität bei beschränktem Nichtdeterminismus.
Es ist einfach zu beobachten, daß wenn Spieler B nur m Eingabebits besitzt und Spieler A m nichtdeterministische Bits hat, die Einweg Kommunikation gleich der nichtdeterministischen Kommunikationskomplexität ist: Die nichtdeterministische Kommunikationskomplexität ist in diesem Fall höchstens m, und daher kann A den gesamten Dialog raten.
Für eine Anwendung auf Formellänge untersuchen wir Funktionen mit asymmetrischer Aufteilung der Eingabebits, d.h. bei denen A viel mehr Eingabebits erhält als B. Für nichttriviale Resultate muß dann die Anzahl der nichtdeterministischen Bits kleiner als die Anzahl der Eingaben eines Spielers sein.

Eine andere Beobachtung ist, daß man mit s nichtdeterministischen Bits die Kommunikation (ausgehend von der deterministischen Kommunikation d) bestenfalls auf $d/2^s$ drücken kann. Ist also s sublogarithmisch, so folgen starke untere Schranken oft bereits von Schranken für deterministische Kommunikation, z.B. ist $N_{\epsilon \log n}(\neg EQ) \geq n^{1-\epsilon}$, während $N_{\log n}(\neg EQ) = O(\log n)$. Andererseits gilt aber:

Lemma 4.1

$$N_s^{(1)}(f) = c \Rightarrow N_c^{(1)} \leq c.$$

Beweis: Bei c Bits Kommunikation können nur 2^c verschiedenen Nachrichten verwendet werden (für alle Eingaben). Um eine der Nachrichten zu raten reichen daher immer c nichtdeterministische Bits aus. Alle Ratebits sind privat. □

Also ist es sinnlos, mehr zu raten als zu kommunizieren. Wir sind daran interessiert, festzustellen, wie groß der Unterschied zwischen nichtdeterministischer Einweg Kommunikation mit s Ratebits und unbeschränkt nichtdeterministischer Einweg Kommunikation sein kann. Dazu betrachten wir den zu erreichenden Unterschied als eine Funktion G.

Korollar 4.1 *Sei $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ eine Boolesche Funktion und $G : \mathbb{N} \to \mathbb{N}$ eine monoton steigende Funktion mit*
$N^{(1)}(f) = N(f) = c$ und $N_s(f) = G(c)$ für ein s.
Dann gilt $N^{(1)}_{G^{-1}(n)}(f) \leq c$ und daher $s \leq G^{-1}(n)$.

Beweis: Es ist $G(c) \leq n$ und somit $c \leq G^{-1}(n)$. □

Der Bereich der Werte von s, in dem ein Unterschied G zwischen $N(f)$ und $N_s^{(1)}(f)$ beweisbar ist, ist daher eingeschränkt. Wenn z.B. ein exponentieller Unterschied $G(x) = 2^x$ gezeigt werden soll, so muß $s \leq \log n$ sein. Wenn $G(x) = r \cdot x$, dann muß $s \leq n/r$ gelten.

Wir zeigen nun einen Unterschied zwischen nichtdeterministischer Einweg Kommunikation mit s Ratebits und unbeschränkt nichtdeterministischer Einweg Kommunikation. Zuerst definieren wir die dabei betrachtete Funktion.

Definition 4.1 *$OD_{n,s}$ bezeichne folgende Boolesche Funktion für $1 \leq s \leq n$:*

$$\begin{aligned} OD_{n,s}(x_1, \ldots, x_n; x_{n+1}) = 1 \iff & \forall i : x_i \in \mathcal{P}(n^3, s) \\ & \wedge \exists i : |\{j | j \neq i; x_i \cap x_j \neq \emptyset\}| \geq s. \end{aligned}$$

Wir betrachten die Aufteilung der Eingaben, bei der B die Menge x_{n+1} und A alle anderen Mengen erhält. Die oberen Schranken des folgenden Lemmas sind trivial, weil B nur $O(s \log n)$ Eingaben erhält.

Lemma 4.2

$$N^{(1)}_{O(s \log n)}(OD_{n,s}) = O(s \log n).$$

$$D^{(B,1)}(OD_{n,s}) = O(s \log n).$$

Die folgende untere Schranke ergibt einen fast optimalen Unterschied zwischen nichtdeterministischer Kommunikation und beschränkt nichtdeterministischer Einweg Kommunikation. Beschränkt nichtdeterministische Einweg Kommunikation wird auch in [HrSa00] untersucht.

Theorem 4.1 *Es gibt eine Konstante $\epsilon > 0$, so daß für $s \leq n$ gilt*

$$N^{(1)}_{\epsilon s}(OD_{n,s}) = \Omega(ns \log n).$$

Beweis: Wir müssen zeigen, daß alle nichtdeterministischen Einweg Protokolle mit ϵs Ratebits für $OD_{n,s}$ eine hohe Kommunikation haben. Ein nichtdeterministisches Einweg Protokoll mit ϵs Ratebits und Kommunikation c ist äquivalent zu einer Überdeckung der Kommunikationsmatrix mit $2^{\epsilon s}$ Booleschen Matrizen mit den folgenden Eigenschaften: Jeder 1-Eintrag der Kommunikationsmatrix ist in mindestens einer der Matrizen ein 1-Eintrag, kein 0-Eintrag der Kommunikationsmatrix ist in einer der Matrizen ein 1-Eintrag, die Menge der verschiedenen Zeilen, welche in den Matrizen vorkommen, hat eine Größe von höchstens 2^c. Wir zeigen die untere Schranke unter Verwendung der Eigenschaft, daß jede einzelne Matrix nur maximal 2^c verschiedene Zeilen hat. Diese Eigenschaft korrespondiert zu einem stärkeren Modell von Protokollen mit beschränktem, aber öffentlichem Nichtdeterminismus.
Wir konstruieren eine Teilmatrix der Kommunikationsmatrix mit nützlichen Eigenschaften, und zeigen dann das Theorem für das „einfachere“ Problem. Hierzu partitionieren wir das Universum $\{1, \ldots, n^3\}$ in n disjunkte Mengen $U_1, \ldots, U_n$ mit $|U_i| = n^2 = m$. Wir wählen Vektoren von n Teilmengen des Universums, so daß die $i$te Teilmenge jeweils aus U_i ist. Also sind alle n Teilmengen eines Vektors paarweise disjunkt. Nun muß das Protokoll feststellen, ob die Menge von Spieler B sich mit s der Mengen von Spieler A nichtleer schneidet.
Wir schränken die Menge der Eingaben weiter ein. Es gibt $\binom{m}{s}$ Teilmengen von U_i der Größe s. Wir wählen eine Menge solcher Teilmengen, so daß je zwei von ihnen nicht mehr als $s/2$ gemeinsame Elemente haben. Dazu beginnen wir mit irgendeiner Teilmenge, und entfernen alle Teilmengen in „Distanz“ höchstens $s/2$. Dies setzen wir so lange wie möglich fort. Wir erhalten eine Menge von Teilmengen von U_i, deren Elemente eine paarweise

Distanz von $s/2$ haben. In jedem Schritt werden höchstens $\binom{s}{s/2}\binom{m}{s/2}$ Teilmengen entfernt, daher erhalten wir mindestens

$$\frac{\binom{m}{s}}{\binom{s}{s/2}\binom{m}{s/2}} \geq \left(\frac{m}{s}\right)^{s/2} / 2^{3s/2} \tag{4.1}$$

Mengen.
Wir ziehen die Eingaben von A wieder als Vektoren von Mengen, wobei wir jetzt an Position i eine der eben konstruierten Teilmengen von U_i ziehen. Diese Eingaben indizieren die Zeilen der Teilmatrix der Kommunikationsmatrix. Die Spalten der Teilmatrix sind eingeschränkt auf Elemente von $U_1 \times \cdots \times U_n$, bei denen nur s Positionen besetzt sind. Wir nennen die konstruierte Teilmatrix M.
Wir nehmen nun an, es gäbe ein Protokoll für das eingeschränkte Problem. Durch Fixieren des Rateworts erhalten wir eine Matrix M', welche mindestens $1/2^r$ der Einsen von M überdeckt, wobei $r = \epsilon s$. Wir werden zeigen, daß eine solche Matrix notwendigerweise viele verschiedene Zeilen hat, was dazu führt, daß die Kommunikation hoch ist.
Zu jeder Zeile von M gehört ein Vektor mit n Positionen, an denen Mengen stehen. i heiße eine *Differenz Position* zweier solcher Vektoren von Mengen, wenn beide an Position i unterschiedliche Einträge/Mengen aus U_i haben. Diese unterschiedlichen Teilmengen von U_i haben nach Konstruktion sogar nicht mehr als $s/2$ gemeinsame Elemente.
Wir sagen, daß eine Menge von Zeilen k Differenz Positionen hat, wenn es k Positionen gibt, so daß jede der Positionen für die Vektoren von mindestens zwei der Zeilen eine Differenz Position ist.
Wir zeigen nun, daß jede Zeile mit vielen Einsen mit vielen Zeilen von M nicht übereinstimmt, das heißt Einsen enthält, welche diese nicht haben. Da M' einseitigen Fehler hat, müssen die Zeilen von M' daher entweder dünn mit Einsen besetzt sein, oder mit nur wenigen Zeilen von M verträglich sein. Man beobachte, daß jede Zeile von M genau $\binom{n}{s}s^s$ Einsen enthält.

Lemma 4.3 *Sei z eine Zeile von M', die mehrfach in M' auftaucht. Die Zeilen von M, an deren Stelle in M die Zeile z in M' auftaucht, mögen δn Differenz Positionen haben. Dann enthält z höchstens $2\binom{n}{s}s^s/2^{\delta s/6}$ Einsen.*

Beweis: Es seien also mehrere Zeilen von M mit δn Differenz Positionen gegeben, so daß die Einsen von z in allen diesen Zeilen vorkommen. Es sei C die Menge von $\binom{n}{s}s^s$ Spalten/Mengen, die Einsen in der ersten solchen Zeile seien. Alle anderen Spalten sind unerlaubt und dürfen in z nicht Eins sein. Eine Spalte in C wird zufällig gezogen, indem zuerst s aus n Positionen gezogen werden, und dann eines aus s Elementen an jeder Position. Sei $k = \delta s$.
Zuerst zählen wir die Spalten in C, welche mit höchstens $k/2$ der Mengen U_i an den Differenz Positionen einen nichtleeren Schnitt haben. Eine Spalte

in C wird gezogen, indem zuerst s aus n Positionen gezogen werden (und dann ein Element an der Position). Betrachten wir das Experiment, daß s mal unabhängig eine aus n Positionen gezogen wird, wobei eventuell eine Position mehrfach gezogen werden kann. Erwartet werden nun $\delta s = k$ der Differenz Positionen gezogen. Mit Chernovs Ungleichung gilt dann aber, daß mit Wahrscheinlichkeit höchstens

$$e^{-\frac{1}{4\cdot 2}\cdot k} \leq 2^{-\delta s/6}$$

höchstens $k/2$ Differenz Positionen vorkommen. Wählt man nun stattdessen eine zufällige Spalte in C, so ist diese Wahrscheinlichkeit noch kleiner, da hier die Positionen ohne Zurücklegen gezogen werden. Also tragen die Spalten in C, die weniger als $k/2$ Differenz Positionen treffen, höchstens $2^{-\delta s/6}\binom{n}{s}s^s$ Einsen zu z bei.
Nun betrachten wir die Spalten/Mengen in C, welche mindestens $k/2$ Universen U_i an Differenz Positionen schneiden. Wie groß ist die Wahrscheinlichkeit für eine zufällige solche Spalte, daß ihre Menge auf alle Zeilen „paßt“ ? Dabei paßt eine Menge auf alle der Zeilen bzw. deren Vektoren von Mengen, wenn jedes Element an einer besetzten Position in der Menge in allen Mengen der Vektoren an der Position liegt. An jeder Differenzposition gibt es aber zwei Vektoren von Mengen, welche sich an der Position unterscheiden, und dabei ist die Distanz der zwei Mengen an dieser Position mindestens $s/2$.
Halten wir also eine beliebige Auswahl der Positionen fest, so daß $k/2$ Distanzpositionen getroffen werden. Jetzt wird eine Spalte gewählt, indem eines von s Elementen an jeder Position ausgesucht wird. Aber wenn die Position eine Differenz Position ist, so ist die Bedingung, mit allen Zeilen übereinzustimmen, nur für höchstens $s/2$ der Elemente erfüllbar. Daher sind nur $2^{-k/2}$ aller Wahlen von Spalten zulässig. So können nur höchstens $\binom{n}{s}s^s/2^{k/2}$ solcher Spalten eine Eins in z sein.
Insgesamt kann nur ein Bruchteil von $2^{-\delta s/6+1}$ aller Spalten in C eine Eins in z sein. □
Mindestens die Hälfte aller Einsen in M' liegen in Zeilen mit $\geq \binom{n}{s}s^s/2^{r+1}$ Einsen. Nach Lemma 4.3 wissen wir, daß eine solche Zeile nur auf einer Menge von Zeilen von M erlaubt ist, die nicht mehr als δn Differenz Positionen hat, wobei $r + 1 = \delta s/6 - 1$. Also kann eine solche Zeile nicht mehr als alle Einsen in $\binom{m}{s}^{\delta n}$ Zeilen von M bedecken, und damit nur $\binom{m}{s}^{\delta n}\binom{n}{s}s^s$ Einsen. Weil aber nach (4.1) mindestens $(m/s)^{sn/2}\binom{n}{s}s^s/(2^{3sn/2}2^{r+1})$ Einsen von solchen Zeilen bedeckt werden, sind

$$\frac{(m/s)^{sn/2}\binom{n}{s}s^s}{\binom{m}{s}^{\delta n}\binom{n}{s}s^s 2^{3sn/2}2^{r+1}}$$
$$\geq \frac{(m/s)^{sn/2}}{(em/s)^{6\epsilon sn+12n}2^{3sn/2}2^{\epsilon s+1}}$$

$$= 2^{\Omega(sn \log n)}$$

Zeilen notwendig (bei $\epsilon = 1/20$ und $n \geq s \geq 400$). □

4.2.2 Eine Rundenhierarchie

Die Pointer Jumping Funktion aus Abschnitt 3.2.2 erfordert es, einem Pfad der Länge $k+1$ in einem bipartiten Graphen mit Ausgrad 1 zu folgen. Ein nichtdeterministisches Protokoll mit $\log n$ Ratebits, in dem B die Kommunikation beginnt, kann nun einfach die erste Kante raten, und diese sowie die zweite Kante an A schicken. Wurde richtig geraten, so kann man die erste Runde sparen und nach $k-1$ Runden fertig sein, ohne die Kommunikation im Vergleich zur Situation, wenn A anfängt, zu erhöhen. Wenn wir eine Rundenhierarchie für (superlogarithmisch) beschränkten Nichtdeterminismus nachweisen wollen, so brauchen wir also eine schwierigere Funktion. Wir verallgemeinern dazu Pointer Jumping (die folgende Funktion wurde in [Kl98] eingeführt).

Definition 4.2 *Seien V_A und V_B disjunkte Mengen von jeweils n Knoten. Sei $F_A = \{f_A | f_A : V_A \to V_B\}$ und $F_B = \{f_B | f_B : V_B \to V_A\}$.*
Weiter sei für feste f_A, f_B definiert, daß
$f(v) = f_{f_A,f_B}(v) = \begin{cases} f_A(v) & \text{wenn } v \in V_A, \\ f_B(v) & \text{wenn } v \in V_B. \end{cases}$
Definiere $f^{(0)}(v) = v$ und $f^{(k)}(v) = f(f^{(k-1)}(v))$.
Dann ist $g_k^s : V_A^s \times F_A \times F_B \to (V_A \cup V_B)^s$ definiert als $g_k^s(v_1^1, \ldots, v_1^s, f_A, f_B) = (f_{f_A,f_B}^{(k+1)}(v_1^1), \ldots, f_{f_A,f_B}^{(k+1)}(v_1^s))$.
Die Funktion $f_k^s : V_A^s \times F_A \times F_B \to \{0,1\}$ ist das XOR der Bits in der Binärkodierung der Ausgabe von g_k^s.

Obige Definition verallgemeinert Pointer Jumping, indem nun s Pfaden parallel gefolgt werden muß, auf einem einzelnen Graphen. Spieler A erhält ein Element von F_A, Spieler B ein Element von F_B, und beide kennen s Startknoten. Intuitiv gesehen muß Spieler B $s \log n$ Bits raten, um die erste Runde produktiv zu nutzen.
Ein probabilistisches Protokoll für Pointer Jumping von [NW93] (siehe Abschnitt 3.2.2) kann zu einem Las Vegas Protokoll mit Aufgabewahrscheinlichkeit 1/2 und damit zu einem nichtdeterministischen Protokoll für g_k^1 verändert werden, das $O(\log n)$ nichtdeterministische Bits benutzt, und Kommunikation $O((n/k+k)\log n)$ hat, bei k Runden, wenn B startet. Das Protokoll läßt sich allerdings anscheinend nicht sinnvoll für g_k^s verallgemeinern. In [PRV99] wird ein Protokoll beschrieben, das g_k^s berechnet, und besser als die triviale $n \log n$ Schranke ist. Sei $h^{(k)}(n,s) = \log n$ für $k \leq 0$ und $h^{(k)}(n,s) = \log h^{(k-1)}(n,s) + \log s$ sonst.

Fakt 4.1 $D^{(B,k)}(g_k^s) = O(n \cdot h^{(k-2)}(n,s))$.

Das folgende Lemma gibt eine untere Schranke.

Lemma 4.4 $N^{(B,k)}_{s/3200}(f^s_k) \geq \frac{n}{2s^2(k+1)^2} - 2s(k+2)\log n$.

Beweis: Wie in der Analyse von [NW93] für das normale Pointer Jumping Problem betrachten wir Protokolle für f^s_k mit folgenden Eigenschaften. Es gibt k Runden, B beginnt die Kommunikation. In Runde $t \geq 1$ werden die Knoten $v^i_t = f^{(t-1)}(v^i_1)$ für alle i kommuniziert, d.h. allen Pfaden mit einer Runde „Verspätung“ gefolgt. Die Forderung der letzten Eigenschaft kostet $sk \log n$ Bits zusätzliche Komunikation. Insgesamt sei die Kommunikation dann $(\epsilon/2)n - s(k+2)\log n$ Bits mit $\epsilon = 1/(s^2(k+1)^2)$. Wir werden zeigen, daß diese Kommunikation nicht ausreicht, was die untere Schranke zeigt.
Ein nichtdeterministisches Protokoll mit $s/3200$ Ratebits führt auf $2^{s/3200}$ deterministische Protokolle, welche Sprachen erkennen, die zusammen f^s_k überdecken. Wir betrachten den Protokollbaum eines solchen deterministischen Protokolls. Jeder Knoten z des Baums wird mit einer Teilmatrix $F^z_A \times F^z_B$ von Eingaben markiert, welche mit der Kommunikation auf dem Pfad von der Wurzel zu z konsistent sind. Sei $\delta = \sqrt{\epsilon}/20$. Es seien c_z Bits bis zum Erreichen von Knoten z kommuniziert.
Unser Ziel ist es, zu zeigen, daß mit Wahrscheinlichkeit mindestens $1 - 2^{-s/3195+1}$ eine gemäß der Gleichverteilung zufällig gezogene Eingabe ein Blatt des Protokollbaums erreicht, in dem die Entropie der Endknoten von $\Omega(s)$ Pfaden, jeweils unter der Bedingung, daß die anderen Pfade fixiert sind, hoch ist. Das erlaubt dann zu zeigen, daß die deterministische Komplexität mit hohem, aber einseitigem Fehler, groß sein muß, und somit auch die nichtdeterministische Komplexität bei einer beschränkten Anzahl von Ratebits.
Wir untersuchen nun einen der s Pfade, denen gefolgt werden soll, sei dies Pfad i. Alle anderen Pfade enthalten zusammen höchstens $(s-1)(k+1)$ Kanten. Die Zufallsvariable Y_i entspreche den Möglichkeiten, diese übrigen $s-1$ Pfade zu fixieren. Sei Ω die Menge aller Eingaben. Y_i bildet eine gleichverteilt gezogene Eingabe auf $s-1$ der Pfade in der Eingabe ab, beginnend mit $s-1$ zufälligen Startknoten v^j_i mit $j \neq i$. Also ist Y_i uniform verteilt auf den Möglichkeiten, alle $s-1$ Pfade außer i zu fixieren. Ein elementares Ereignis zu der Variable Y_i besteht aus allen Eingaben, welche eine Menge von $s-1$ Pfaden gemeinsam haben. Ereignisse von Y_i sind dann Vereinigungen der Elementarereignisse. Manche Ereignisse entsprechen dem Fixieren von $r \leq s-1$ Pfaden. Solche Ereignisse sollen *pfadfixierende Ereignisse* heißen.
Ein Knoten z in Tiefe t im Protokollbaum heiße *sehr gut* für i, wenn $1 \leq i \leq s$ und

1. $H(F^z_A|Y_i) \geq n\log n - 2c_z - 2s(k+2)\log n$
2. $H(F^z_B|Y_i) \geq n\log n - 2c_z - 2s(k+2)\log n$

3. $H(F_B^z(v_t^i)|Y_i) \geq \log n - \delta$

mit einer analogen Definition, wenn A an der Reihe ist. An der Wurzel ($t = 0$) sei der dritte Teil als $H(v_1^i|Y_i) \geq \log n - \delta$ definiert.

Behauptung 4.1 *Sei E ein pfadfixierendes Ereignis zu Y_i. Eine Eingabe, welche unter der Bedingung E gleichverteilt gewählt wird, erreicht kein für i sehr gutes Blatt mit Wahrscheinlichkeit höchstens $(k+1)(22\sqrt{\epsilon} + 1/n^s)$.*

Beweis: Es sei bemerkt, daß in der Behauptung eine zweifache Bedingung an die F_A und F_B betrachtet wird. Zuerst wird die Wahl der Eingaben bzw. Knoten im Protokollbaum unter der Annahme des Ereignisses E betrachtet, und zweitens sind die Entropien über die Zufallsvariable Y_i bedingt. Zunächst zeigen wir, daß die ersten beiden Teile der Eigenschaft "sehr gut" in Tiefe j im Protokollbaum mit Wahrscheinlichkeit mindestens $1 - j/n^s$ gelten.
Man betrachte den Protokollbaum. Wir schränken den Baum auf solche Eingaben, welche E erfüllen, ein. Der neue Protokollbaum hat eine Wurzel, zu welcher alle mit E konsistenten Eingaben gehören. Da E pfadfixierend ist, erhalten wir so eine Teilmatrix $G_A \times G_B$ der Kommunikationsmatrix, denn es werden einfach einige der Pointer in F_A und F_B fixiert. Die Matrix der Eingaben an Knoten z, welche E erfüllen, heiße $G_A^z \times G_B^z$, dabei bezeichnen G_A^z und G_B^z auch uniform auf den Zeilen/Spalten verteilte Zufallvariablen. Der eingeschränkte Protokollbaum wird weiterhin zu den Kommunikationen im alten Protokoll assoziiert, dabei sei wie bisher die Länge der Kommunikation bis Knoten z mit c_z bezeichnet. Das eingeschränkte Protokoll kommuniziert in Runde t alle v_t^i, die Kommunikation hierzu wird in c_z mitgerechnet, auch wenn durch E fixierte Knoten kommuniziert werden.
Es gilt an der Wurzel $H(G_A) = H(G_B) \geq n \log n - s(k+2) \log n$, weil durch E höchstens $(s-1)(k+1)$ Kanten fixiert sind.
Wir gehen nun abwärts im Baum. Sei w ein zufälliges Kind eines Knoten z mit $H(G_A^z) \geq H(G_A) - 2c_z$ und $H(G_B^z) \geq H(G_B) - 2c_z$. Das Kind wird gewählt mit der Wahrscheinlichkeit, die sich aus dem Verhältnis der Anzahl der Eingaben in dem Kind zu der Anzahl der Eingaben im Elternknoten ergibt. Angenommen A sendet in z eine Nachricht, und a_w sei die Länge der Nachricht, welche zu w führt. Dann ist $c_w = c_z + a_w$.
$\Pr(H(G_B^w) < H(G_B) - 2c_w) = 0$, weil B nichts gesendet hat.
$\Pr(H(G_A^w) < H(G_A) - 2c_w) \leq 1/n^s$ wie folgt. Sei μ die Gewichtsfunktion der Gleichverteilung auf den Eingaben, die E erfüllen. Ein Kind wird mit Wahrscheinlichkeit $\mu(G_A^w)/\mu(G_A^z)$ gewählt und

$$H(G_A^w) = H(G_A) + \log \mu(G_A^w).$$

Daher gilt

$$\begin{aligned}
&\Pr(H(G_A^w) < H(G_A) - 2c_w) \\
&\leq \Pr(\mu(G_A^w) < 2^{-2c_w}) \\
&\leq \Pr(\mu(G_A^w)/\mu(G_A^z) < 2^{-2a_w}) \\
&\leq \sum_w 2^{-2a_w} \\
&\leq \frac{1}{n^s} \sum_w 2^{-a_w} \leq \frac{1}{n^s}.
\end{aligned}$$

Der vorletzte Schritt ist wahr, weil an Knoten z immer s Kanten kommuniziert werden und daher $a_w \geq s \log n$ gilt. Die letzte Ungleichung folgt mit der Kraftschen Ungleichung Fakt 2.31 (jede Nachricht „kodiert“ eine Zahl, und die Länge der Nachricht ist eine Kodierungslänge).
Die Wahrscheinlichkeit, einen Knoten in Tiefe j zu erreichen, der $H(G_A^w) < n \log n - 2c_w - s(k+2) \log n$ oder $H(G_B^w) < n \log n - 2c_w - s(k+2) \log n$ erfüllt, ist also höchstens j/n^s. Es gilt $H(F_A^w) \geq H(G_A^w)$ für alle w. Weiterhin gilt $H(F_A^w|Y_i) \geq H(F_A^w) - \log |Y_i|$, weil

$$H(X) \geq H(X|Y_i) \geq H(X) - H(Y_i) \geq H(X) - \log |Y_i|$$

für alle X und Y_i, dabei sei $|Y_i|$ die Größe des Wertebereichs von Y_i.
Daher gelten die ersten beiden Teile der Eigenschaft sehr guter Knoten in Tiefe j mit Wahrscheinlichkeit mindestens $1 - j/n^s$ (offensichtlich ist $\log |Y_i| \leq (s-1)(k+2) \log n$).
Nun ist noch der dritte Teil der Eigenschaft sehr gut zu untersuchen. Wir zeigen, daß bei einer unter der Bedingung E gezogenen Eingabe mit hoher Wahrscheinlichkeit in Tiefe t im Protokollbaum ein Knoten z erreicht wird, an dem v_{t+1}^i hohe Entropie (unter den Eingaben in $G_A^z \times G_B^z$) hat, und folgern dann Eigenschaft drei für Blätter. Beginnen wir an der Wurzel des Protokollbaums. Weil der Startknoten zufällig ist, gilt $H(v_1^i|Y_i) = \log n$, auch unter dem Ereignis E.
Als nächstes untersuchen wir ein zufälliges Kind w eines Knotens z in Tiefe t. O.B.d.A. sende B an z. z habe die Eigenschaft, daß $H(v_{t+1}^i) = H(G_B(v_t^i)) \geq \log n - \delta$ und daß $H(G_B^z) \geq H(G_B) - 2c_z$ und $H(G_A^z) \geq H(G_A) - 2c_z$. Nach obigem gilt dann mit Wahrscheinlichkeit $1 - 1/n^s$, daß auch $H(G_B^w) \geq H(G_B) - 2c_w$ und $H(G_A^w) \geq H(G_A) - 2c_w$. Diese Eigenschaften nehmen wir im folgenden als gegeben an.
Wir benutzen Fakt 2.29, der uns erlaubt, die Wahrscheinlichkeit von Ereignissen zu einer Verteilung mit hoher Entropie durch deren Wahrscheinlichkeit unter der Gleichverteilung zu approximieren.
Wieder sei a_w die Länge der Nachricht, welche von z zu w führt, also $c_w = c_z + a_w$. Wir betrachten nur Kinder w, bei denen $\sum_{v \in V_A} H(G_A^w(v)) \geq H(G_A^w) \geq n \log n - 2c_w - s(k+2) \log n \geq n \log n - \epsilon n$ gilt. Wenn ein

v gemäß der Gleichverteilung gezogen wird, dann ist für alle solchen w $\Pr_U(H(G_A^w(v)) < \log n - \delta) \leq \epsilon/\delta$ aufgrund der Markov Ungleichung.
Aber $H(v_{t+1}^i|E) = H(G_B^z(v_t^i)) \geq \log n - \delta$. Bei der Wahl eines Kindes w von z wird v_{t+1}^i also unter einer Verteilung mit hoher Entropie gewählt. Mit Fakt 2.29 gilt folgendes:

$$\begin{aligned}
&\Pr(H(G_A^w(v_{t+1}^i)) < \log n - \delta) \\
&\leq \epsilon/\delta \left(1 + \sqrt{\frac{4\delta}{\epsilon/\delta}}\right) \\
&= 22\sqrt{\epsilon}.
\end{aligned}$$

Also ist mit Wahrscheinlichkeit $1 - 22\sqrt{\epsilon} - 1/n^s$ die Entropie von $G_A^w(v_{t+1}^i)$ mindestens $\log n - \delta$ und $H(G_A^w), H(G_B^w) \geq n \log n - 2c_w - s(k+2) \log n$, wenn die analogen Eigenschaften für z gelten.
In Tiefe k wird also mit Wahrscheinlichkeit $1 - k22\sqrt{\epsilon} - k/n^s$ ein Knoten z mit $H(G_A^z(v_k^i)) \geq \log n - \delta$ und $H(G_A^z), H(G_B^z) \geq n \log n - 2c_z - s(k+2) \log n$ erreicht. Ist dies der Fall, so wird mit Wahrscheinlichkeit $1 - 1/n^s$ ein Blatt w in Tiefe $k+1$ erreicht, bei dem $H(G_A^w), H(G_B^w) \geq n \log n - 2c_w - s(k+2) \log n$ gilt. Weiterhin gilt in diesem Fall auch $H(F_A^w|Y_i), H(F_B^w|Y_i) \geq n \log n - 2c_w - 2s(k+2) \log n \geq n \log n - \epsilon n$.
Nun interessiert uns die Wahrscheinlichkeit, bei gleichverteilter Wahl einer Eingabe unter der Bedingung E ein Blatt in Tiefe $k+1$ zu erreichen, das sehr gut ist. Wir nehmen an, daß die ersten zwei Teile der Eigenschaft sehr gut gelten und daß ein Elternknoten z mit $H(G^z(v_k^i) \geq \log n - \delta$ erreicht wird. All das gilt mit Wahrscheinlichkeit $1 - (k+1)/n^s - k22\sqrt{\epsilon}$.
Für alle w mit den ersten zwei Eigenschaften der Eigenschaft sehr gut gilt $H(F_A^w|Y_i) \geq n \log n - 2c_w - 2s(k+1) \log n \geq n \log n - \epsilon n$. Dann ist wieder $\sum_{v \in V_A} H(F_A^w(v)|Y_i) \geq H(F_A^w|Y_i) \geq n \log n - \epsilon n$ für solche w. Wenn ein v gemäß der Gleichverteilung gezogen wird, dann ist für alle solchen w $\Pr_U(H(F_A^w(v)|Y_i) < \log n - \delta) \leq \epsilon/\delta$ aufgrund der Markov Ungleichung.
Aber $H(v_{k+1}^i|E) = H(F_B^z(v_k^i)|E) = H(G_B^z(v_k^i)) \geq \log n - \delta$. Bei der Wahl eines Kindes w unter der Bedingung E wird v_{k+1}^i also unter einer Verteilung mit hoher Entropie gewählt. Mit Fakt 2.29 gilt folgendes:

$$\begin{aligned}
&\Pr(H(F_A^w(v_{k+1}^i)|Y_i) < \log n - \delta) \\
&\leq \epsilon/\delta \left(1 + \sqrt{\frac{4\delta}{\epsilon/\delta}}\right) \\
&= 22\sqrt{\epsilon}.
\end{aligned}$$

Daher wird mit Wahrscheinlichkeit $1 - (k+1)(22\sqrt{\epsilon} + 1/n^s)$ ein sehr gutes Blatt erreicht. □

Nun analysieren wir die Anzahl der i, für die ein zufälliges Blatt sehr gut ist. Wir betrachten das Experiment, eine Eingabe uniform zufällig zu ziehen. Die Zufallsvariable X_i sei 1, wenn die Eingabe ein Blatt erreicht, welches nicht sehr gut für i ist. Es sei $X = \sum X_i$, wobei das Experiment weiterhin ist, eine Eingabe zu ziehen.
Ein Filter (siehe Definition 2.23) auf dem Raum der Ereignisse kann durch eine Folge von einander verfeinernden Partitionen erzeugt werden, wobei jede Partition i einer Menge von Elementarereignissen von $\mathcal{F}_i$ entspricht.
Wir betrachten den Filter, der wie folgt erzeugt wird: $\mathcal{F}_0$ ist trivial, d.h. enthält die Menge aller Eingaben sowie die leere Menge. Für $\mathcal{F}_1$ partitionieren wir die Menge aller Eingaben anhand der Möglichkeiten, den ersten Pfad inklusive seines Startknotens zu fixieren. Für $\mathcal{F}_i$ partitionieren wir vorherige Partition anhand der Möglichkeiten, den $i$ten Pfad inklusive seines Startknotens zu fixieren, d.h. wir partitionieren die Menge aller Eingaben anhand der Möglichkeiten, Pfade 1 bis i inklusive der Startknoten zu fixieren. $\mathcal{F}_{s+1}$ ist dann $\mathcal{P}(\Omega)$.
Die Zufallsvariable unseres Interesses ist $X = \sum X_i$, die Anzahl der Pfade, für welche ein zufälliges Blatt nicht sehr gut ist. Es sei $Z_0 = E[X]$ und $Z_i = E[X|\mathcal{F}_i]$, d.h. der Erwartungswert der Summe und die Zufallsvariable des Erwartungswertes der Summe, abhängig davon, wie die ersten i Pfade fixiert sind. Wir beschränken die Wahrscheinlichkeit, daß $|Z_s - Z_0|$, d.h. die Differenz zwischen X und seinem Erwartungswert, groß ist, mit Azumas Ungleichung, Fakt 2.27. Man beobachte, daß die Folge $Z_0, \ldots, Z_{s+1}$ wegen Fakt 2.26 ein Martingal ist.
Was passiert mit der Erwartungswert der Summe, wenn ein Pfad mehr fixiert wird? Der letzte Knoten des Pfades kann eine geringe Entropie erhalten, daher ist $X_i = 1$ anzunehmen. Andere Veränderungen in $E[X] = \sum E[X_i]$ können wie folgt beschränkt werden:

$$E[\sum_{j>i} X_j|\mathcal{F}_i] \leq \sum_{j>i} E[X_j|E_j]$$

für pfadfixierende Ereignisse E_j und so folgt

$$\begin{aligned}
&|Z_i - Z_{i-1}| \\
&\leq 1 + E[\sum_{j>i} X_j|\mathcal{F}_i] - E[\sum_{j>i} X_j|\mathcal{F}_{i-1}] \\
&\leq 1 + \sum_{j>i} E[X_j|E_j] \\
&\leq 1 + s(22(k+1)\sqrt{\epsilon} + (k+1)/n^s) \leq 24
\end{aligned}$$

mit Behauptung 4.1. Mit

$$Z_0 = E[X] = \sum E[X_i] \leq s \cdot (22(k+1)\sqrt{\epsilon} + (k+1)/n^s) \leq 23$$

erhalten wir über Fakt 2.27

$$\Pr(\sum X_i > s/2 + 23) \leq \Pr(|Z_s - Z_0| \geq s/2)$$
$$\leq 2e^{-\frac{s^2/4}{2s \cdot 576}} \leq 2 \cdot 2^{-s/3195}.$$

Nun können wir das Lemma zeigen: Gegeben sei ein nichtdeterministisches Protokoll, bei dem B startet, das k Runden hat und $s/3200$ Ratebits verwendet. Das Protokoll induziert ein deterministisches Protokoll, welches mindestens $2^{-s/3200}$ aller 1-Eingaben und daher $2^{-s/3200-1}$ aller Eingaben akzeptiert. Aber jedes derartige Protokoll mit der gegebenen Kommunikation hat die Eigenschaft, daß im zugehörigen Protokollbaum mit Wahrscheinlichkeit höchstens $2^{-s/3195+1}$ ein Blatt erreicht wird, in dem kein v^i_{k+2} die bedingte Entropie von mindestens $\log n - \delta$ hat. Ein akzeptierendes Blatt des Protokollbaums ist zu einer Matrix von Eingaben assoziiert, die 1-chromatisch ist, d.h. die Parität der $k$ten Knoten aller Pfade ist bekannt. Aber dann ist $H(F^{(k+1)}(v^i_1)|Y_i = y) \leq \log n - 1$ für alle i und y. Somit ist für kein i $H(F^{(k+1)}(v^i_1)|Y_i) \geq \log n - \delta$ und monochromatische Blätter sind für kein i sehr gut.
Daher kann kein Protokollbaum mit der erlaubten Kommunikation und Tiefe, bei dem B startet, einen Anteil von $2^{-s/3200}$ aller 1-Eingaben akzeptieren, aber keine 0-Eingabe. □
Bevor wir Theorem 4.2 herleiten, betrachten wir noch die nichtdeterministische Kommunikationskomplexität bei unbeschränktem Nichtdeterminismus.

Lemma 4.5 $N(f^s_k) = \Omega(sk \log(n/(sk)))$.

Beweis: Wir zeigen eine Reduktion vom folgenden Problem:

$$p(x_1, \ldots, x_s, y_1, \ldots, y_s) = \bigoplus_{i=1}^{s} (x_i = y_i)$$

auf Worten $x_i, y_i \in \{0,1\}^{\Theta(k \log(n/(sk)))}$. Dieses Problem hat eine nichtdeterministische Kommunikationskomplexität von $\Omega(sk \log(n/(sk)))$, was bewiesen werden kann mit der Methode der größten monochromatischen Teilmatrix, siehe [KN97].
Zuerst partitionieren wir die Knoten auf jeder Seite in s Mengen von je n/s Knoten. Ein *monotoner Pfad* in Segment i ist ein Pfad, der zwischen linken und rechten Knoten alterniert, die alle in der $i$ten Knotenmenge liegen, wobei der Pfad am ersten Knoten der $i$ten Menge links beginnt, und bei einem Knoten mit PARITÄT 1 endet. Zusätzlich muß die Folge der Knotennummern des Pfades monoton ansteigend sein.
Sei also P_i die Menge der monotonen Pfade der Länge $k+1$ in Segment i. Dann ist $|P_i| = \Omega(\binom{n/s}{k+1})$: man wählt $k+1$ Knoten, sortiert sie. Nun identifizieren wir die linken und rechten Seiten eines solchen monotonen Pfades miteinander und mit einem Wort aus $\{0,1\}^{\log |P_i|}$. Das ist sinnvoll, weil die linke

und rechte Seite zweier verschiedener monotoner Pfade nicht zusammenpassen: bis zu einem Punkt verlaufen die Pfade gleich, dann geht ein Pfad zu einem höher numerierten Knoten als der andere. Eine der links/rechts Kombinationen bricht hier ab, die andere Kombination muß irgendwann auch abbrechen, da zu wenig Kanten auf einer Seite übrig sind.
Wir betrachten die Menge von Eingaben, die zu allen s-Tupeln von monotonen Pfaden der Länge k korrespondiert, dazu nehmen wir alle anderen Kanten, welche zu einem Knoten 0 in Segment 0 zeigen. Wir können jeden Vektor von Pfaden mit einem Vektor von Zeichenketten identifizieren und erhalten die gewünschte Reduktion: jeder falsch zusammengesetzte Pfad trägt eine 0 bei, jeder korrekt zusammengesetzte Pfad trägt eine 1 bei. □
Lemma 4.4 und 4.5 implizieren die untere Schranke für die Funktion f_k^{3200s}. Die oberen Schranken sind offensichtlich. In der Formulierung des Theorems lassen wir die Konstante 3200 weg.

Theorem 4.2 *Für alle s, k und alle n gibt es eine Boolesche Funktion f_k^s mit n Eingaben und*

- $D^{(k)}(f_k^s) = O(sk \log n)$
- $N_s^{(B,k)}(f_k^s) = \Omega(n/(s^2k^2 \log n))$
- $N_{O(s \log n)}^{(B,k)}(f_k^s) = O(sk \log n)$.

4.3 Anwendungen

4.3.1 Nichtdeterministische Einweg Automaten

Die untere Schranke mittels Kommunikation

Ähnlich wie bei deterministischen und probabilistischen Einweg Automaten können untere Schranken für die Größe von nfa bewiesen werden, indem man uniforme nichtdeterministische Einweg Kommunikationsprotokolle betrachtet. Ein uniformes nichtdeterministisches Protokoll ist ein nichtdeterministisches Protokoll für das Kommunikationsproblem induziert von der uniformen Kommunikationsmatrix einer Sprache L wie in Abschnitt 3.3.1 definiert. Nun kann die untere Schranke von Fakt 3.7 einfach an den nichtdeterministischen Fall angepaßt werden [Hr97]. Wegen Fakt 2.1 sind nichtdeterministische Einweg Protokolle so mächtig wie generelle nichtdeterministische Protokolle.

Fakt 4.2 *Die Größe eines minimalen nfa für eine Sprache L ist von unten beschränkt durch die Anzahl der Nachrichten in einem optimalen uniformen nichtdeterministischen Protokoll für L.*

Nichtdeterministische Kommunikationskomplexität scheint die besten unteren Schranken für die Größe von nichtdeterministischen Automaten für L zu liefern. Alle anderen bekannten Techniken wie der Fooling Ansatz sind Spezialfälle davon, siehe auch [HKKSS00]. Es ist auch bekannt [DHS96], daß der Fooling Ansatz manchmal nur exponentiell schlechtere untere Schranken als der Kommunikationsansatz zeigen kann.
Unglücklicherweise zeigt aber das folgende Resultat, daß die Anzahl nichtdeterministische Nachrichten nicht benutzt werden kann, um die Größe optimaler nichtdeterministischer Einweg Automaten zu approximieren.

Lemma 4.6 *Es gibt eine reguläre Sprache L, so daß die Größe eines minimalen nfa für L mindestens $2^{\Omega(n)}$ ist, aber ein uniformes nichtdeterministisches Protokoll für L existiert, welches nur $O(n^2)$ Nachrichten verwendet.*

Beweis:
Sei $L = \{xyz : |x| = |y| = |z| = n, \text{ und } x \neq z \vee x = y\}$. Zuerst beschreiben wir ein nichtdeterministisches Protokoll für L mit nur $O(n^2)$ Nachrichten.
Spieler A und B berechnen die Längen l_I, l_{II} ihrer Eingaben. A sendet l_I und B verwirft, wenn $l_I + l_{II} \neq 3n$. Im folgenden nehmen wir also an, daß $l_I + l_{II} = 3n$.
Fall 1: $l_I \leq n$.
A wählt eine Position $1 \leq i \leq l_I$ und kommuniziert i, x_i, l_I. B akzeptiert, wenn $x_i \neq z_i$. Ansonsten akzeptiert B genau dann, wenn $y = z$.
Wenn nun $x \neq z$, dann gibt es eine akzeptierende Berechnung. Wenn aber $x = z$, dann akeptiert B genau dann, wenn $y = z$, d.h. wenn $x = y$.
Fall 2: $n < l_I \leq 2n$.
A wählt eine Position $1 \leq i \leq n$ und kommuniziert i, x_i, l_I. Weiterhin vergleicht A $x_1, \ldots, x_{l_I - n}$ mit $y_1, \ldots, y_{l_I - n}$ und sendet das Bit $b = 1$, wenn die Worte gleich sind. B akzeptiert, wenn $x_i \neq z_i$. Ansonsten vergleicht B $y_{l_I - n + 1}, \ldots, y_n$ mit $z_{l_I - n + 1}, \ldots, z_n$. Wann die zwei Worte gleich sind und $b = 1$, dann akzeptiert B, und verwirft ansonsten.
Ist es der Fall, daß $x \neq z$, so gibt es eine akzeptierende Berechnung. Wenn nicht, so akzeptiert B genau dann, wenn $x = y = z$.
Fall 3: $2n < l_I \leq 3n$.
A wählt eine Position $l_I - 2n < i \leq n$ und kommuniziert i, x_i, l_I. Weiterhin vergleicht A x mit y. Wenn $x = y$ oder $x_j \neq z_j$ für $1 \leq j \leq l_I - 2n$, dann akzeptiert A. Ansonsten akzeptiert B genau dann, wenn $x_i \neq z_i$.
Das Protokoll benutzt $O(n^2)$ Nachrichten.
Nun zur unteren Schranke für die Größe von nfa für L. Wir betrachten nur Eingaben der Form xyx. Wenn es einen nfa M mit s Zuständen gibt, so besteht nun seine Aufgabe darin, zu testen, ob $x = y$. M kann simuliert werden durch ein nichtdeterministisches (Zweiweg) Protokoll, in dem A x erhält und B y. A bestimmt den nach Lesen von x erreichten Zustand, sendet diesen an B, der auf y weiter simuliert, und den erreichten Zustand an A sendet. A

beendet die Simulation und entscheidet L. Die nichtdeterministische Kommunikation des Protokolls ist $2\lceil \log s \rceil$. Weil aber die nichtdeterministische Kommunikationskomplexität des Gleichheitsproblems n ist (siehe Fakt 2.8), muß $s \geq \Omega(2^{n/2})$ gelten. □
Wir haben also mit der Kommunikationstechnik keine generell gute untere Schranke für die Größe nichtdeterministischer Einweg Automaten erhalten.

Nichtdeterministische Einweg Automaten mit beschränktem Nichtdeterminismus

Einweg Automaten mit beschränktem Nichtdeterminismus werden u.a. in [GKW90] untersucht. Es wird dort gezeigt, daß die Kleene Hülle $(L\#)^*$ einer Sprache L von einem nfa mit konstantem Nichtdeterminismus nur genauso effizient erkannt werden kann wie von einem dfa. Weiterhin zeigen die Autoren, daß man eine Sprache durch einen nfa mit s Ratebits nur mit Größe $\Omega(M^{1/2^s})$ erkennen kann, wenn M die Größe des minimalen dfa für die Sprache ist. Anhand einer konkreten Sprache wird gezeigt, daß manchmal jedes weitere Ratebit die Größe reduzieren hilft, bis schließlich fast die optimale nfa Größe erreicht ist. Bei Kleeneschen Hüllen kann hingegen die Größe nur durch eine unbeschränkte Anzahl von Ratebits verringert werden. Im folgenden beschreiben wir eine Sprache mit einem anderen Verhalten: Obwohl mehr Nichtdeterminismus erlaubt ist, bleibt die Größe von nfa bis zu einem bestimmten Schwellenwert der Anzahl der Ratebits „nah“ an der deterministischen Größe, um dann sehr schnell exponentiell zu fallen. Der Schwellenwert ist polynomiell in der Größe optimaler nfa.

Theorem 4.3 *Es gibt eine Sprache $D_N \subseteq \{0,1\}^N$, so daß D_N von einem nfa mit $O(\sqrt{N \log N})$ Ratebits und $poly(N)$ Zuständen erkannt werden kann, während das Erkennen von D_N mindestens $2^{\Omega(\sqrt{N/\log N})}$ Zustände erfordert, wenn ein nfa nur $\epsilon\sqrt{N/\log N}$ Ratebits benutzt für eine hinreichend kleine Konstante $\epsilon > 0$. Der minimale dfa für D_N hat die Größe $2^{\Theta(\sqrt{N \log N})}$.*

Beweis: Sei

$$D_{64n^2 \log n} = \{x_1 y_1 x_2 y_2 \dots x_n y_n | \forall i : x_i, y_i \in \mathcal{P}(n^{32}, n) \wedge x_i \cap y_i \neq \emptyset\}.$$

Offenbar stimmt $D_{64n^2 \log n}$ mit der Sprache $D_{n,n}$ aus Definition 2.7 überein, wenn man die Worte geeignet permutiert. Die Wortlänge ist $64n^2 \log n$.
Angenommen, es gibt einen nfa mit Nichtdeterminismus s und M Zuständen. Der nfa kann von einem Protokoll simuliert werden, in dem A und B ihre Eingaben wie im Kommunikationsproblem aus Definition 2.7 erhalten, d.h. A erhält die x Eingaben und B die y Eingaben. Das Protokoll hat $2n-1$ Runden und simuliert den nfa, wobei jedesmal der Zustand kommuniziert wird, wenn ein x_i oder y_i verlassen wird. Daher löst das Protokoll das Problem

$D_{n,n}$ mit Kommunikation $O(n \log M)$. Somit gilt aber $M \geq 2^{\Omega(n)}$ nach Fakt 2.14, wenn $s \leq \epsilon n$ (für eine kleine Konstante $\epsilon > 0$).
Andererseits kann ein nfa für alle i je ein Element aus dem Schnitt von x_i und y_i raten, wobei insgesamt $32n \log n$ Bits geraten werden. In diesem Fall kann die Sprache mit $poly(n)$ Zuständen erkannt werden. □

4.3.2 Hierarchien für nichtdeterministische Zweiweg Automaten

Die Resultate des vorigen Abschnitts behandeln die spezielle Beschränkung von „Interaktion“ in Einweg Automaten. In diesem Abschnitt werden Hierarchien gezeigt, bei denen Sprachen effizienter zu erkennen sind, wenn eine bestimmte Länge von Kreuzungsfolgen bzw. eine bestimmte Anzahl von Reversal erlaubt ist. Das Hauptergebnis ist eine Anwendung der Rundenhierarchie aus Abschnitt 4.2.2 auf nichtdeterministische Zweiweg Automaten.
Doch zuerst wenden wir uns Zweiweg Automaten mit unbeschränktem Nichtdeterminismus zu. Auch hier gibt es eine Hierarchie über das Kreuzungsfolgen Maß und das Reversal Maß. Betrachten wir zunächst die maximalen Größenunterschiede, welche sich durch eine Beschränkung ergeben können. Folgende Aussage kann leicht aus Standard Konstruktionen zur Simulation von Zweiweg Automaten durch Einweg nfa abgeleitet werden, siehe [HU79].

Fakt 4.3 *Jede Sprache, die von einem nichtdeterministischen k-kreuzungsbeschränkten Automaten mit q Zuständen erkannt werden kann, besitzt auch einen nichtdeterministischen Einweg Automaten mit $O(q^k)$ Zuständen.*

Lemma 4.7 *Jede Sprache, die von einem nichtdeterministischen k-reversalbeschränkten Automaten mit q Zuständen erkannt werden kann, besitzt auch einen $(k-j)$-reversalbeschränkten Automaten mit nur $O(q^{j+2})$ Zuständen (wenn $j < k$).*

Beweis: Gegeben sei ein k-reversalbeschränkter nichtdeterministischer Automat A und eine k-reversalbeschränkte Berechnung des Automaten. Wir wissen, daß A jede solche Berechnung an einem Ende der Eingabe beendet. Ist k gerade, so am linken Ende, ist k ungerade, am rechten Ende.
A führt eine j-Exkursion zwischen i und l aus, wenn A auf i startet, sich beliebig zwischen i und l bewegt, und dabei genau j Reversals inklusive dem „Startreversal“ ausführt und l erreicht. Eine j-Exkursion kann mit Hilfe des Nichtdeterminismus' abgekürzt werden, indem man Kreuzungsfolgen rät. Offensichtlich ist die Kreuzungsfolgenlänge auf j-Exkursionen höchstens j.
Der simulierende Automat A' versucht, eine Berechnung von A und ihr Bewegungsmuster zu finden und Exkursionen abzukürzen.
So werden Exkursionen abgekürzt: Zu Beginn einer Exkursion stehen A und A' auf i. O.B.d.A. sei $i < l$. A' rät nun Kreuzungsfolgen. Der Zustand von A' kann Tupel von bis zu j Zuständen von A enthalten. Der Zustand von A'

wechselt dann in einem Schritt von einem solchen Tupel zu einem anderen. Zu Beginn enthält der Zustand von A' eine Folge, die nur aus dem Zustand von A besteht. Dann geht es so weiter: In einem Schritt liest A das Zeichen an der aktuellen Position. A rät eine (neue) Kreuzungsfolge der Länge höchstens j, wobei die alte Kreuzungsfolge im Zustand ersetzt wird. Es wird eine Kreuzungsfolge geraten, die gegeben den Buchstaben zu der vorigen Folge paßt. Zwei Kreuzungsfolgen passen zusammen, wenn sie in einem legalen Bewegungsmuster des Zweiweg Automaten A benachbarte Kreuzungsfolgen einer Teilberechnung sein können. Das ist nur von den Folgen und dem Buchstaben abhängig. Daher kann man einen solchen Zustandsübergang fest in den Automaten einbauen. Gibt es keine passende Kreuzungssequenz, so verwirft der Automat. Insgesamt werden nur höchstens j Zustände gespeichert. Der Automat A' läuft auf der Exkursion immer nur in eine Richtung.
Wird nach Ausführung der Exkursion l erreicht, angenommen alles Raten hat funktioniert, so ist die gesamte Exkursion durchgeführt und A' kennt den Endzustand der Berechnung auf der Exkursion. Insgesamt werden dabei $O(q^j)$ Zustände benutzt, und A' fährt einmal über den Bereich von i bis l. A' durchläuft also aufeinanderfolgende Exkursionen des Automaten A, bis dessen Berechnung endet. Wenn A' akzeptiert, so hat A' eine korrekte Berechnung von A simuliert und einen akzeptierenden Zustand gefunden. Ansonsten verwirft A'.
Ein beliebiges Bewegungsmuster von A soll also durch Abkürzung von $j+1$-Exkursionen so verändert werden, daß nur $k-j$ Reversals vorkommen. Da Anfang und Ende einer jeden solchen Exkursion ebenfalls geraten werden können, reicht es aus zu zeigen, daß jedes Bewegungsmuster so gekürzt werden kann. Wir nehmen an, daß j gerade ist, ansonsten wird j um 1 erhöht. Somit steigt die Zustandsanzahl auf höchstens $O(q^{j+2})$.
Der Beweis ist eine Induktion über $k-j$.
Betrachten wir zuerst den Fall $k-j=1$. Nun ist k ungerade und $j=k-1$. Die ganze Berechnung kann von einem Einweg nfa ausgeführt werden.
Wir kommen zum Induktionsschluß. Die Berechnung beginne o.B.d.A. mit einer Rechtsbewegung.
Betrachten wir den Fall $k-j$ gerade. Dann ist k gerade und $j \leq k-2$. Der Weg zum rechtesten Punkt der Berechnung ist eine l-Exkursion mit $l \leq k-1$ ungerade. Ist $1 \leq l \leq j+1$, so kann diese Exkursion in einem Durchlauf geraten werden. Dann hat der Rest der Berechnung ein Muster mit $k-l$ Reversals und es müssen noch $j-l+1$ Reversals gespart werden. Das geht per Induktion, weil $(k-l)-(j-l+1) \leq k-j-1$. Ist hingegen $k-1 \geq l > j+1$, so können per Induktion auf der l-Exkursion j Reversals gespart werden, denn $l-j \leq k-j-1$.
Betrachten wir nun den Fall k ungerade und $j < k-1$. Der Weg zum rechtesten Punkt der Berechnung, der vor dem Ende der Berechnung erreicht wird, ist eine l-Exkursion mit $l \leq k-2$ ungerade. Ist $1 \leq l \leq j+1$, so kann diese Exkursion in einem Durchlauf geraten werden. Dann hat der Rest der

Berechnung ein Muster mit $k-l$ Reversals und es müssen noch $j-l+1$ Reversals gespart werden. Das geht per Induktion, weil $(k-l)-(j-l+1) = k-j-1$.
Angenommen $k-2 \geq l > j+1$. Dann können per Induktion auf der l-Exkursion j Reversals gespart werden, denn $1 < l-j \leq k-j-2$. □
Wir kommen nun zu einer Sprache, wo fast maximale Einsparungen für jede hinreichend kleine erlaubte Reversalanzahl auftreten.

Theorem 4.4 *Es gibt eine von k unabhängige Sprache $L \subseteq \{0,1\}^n$, so daß jeder nichtdeterministische k-kreuzungsbeschränkte Automat für das Erkennen von L mindestens $\Omega(N^{1/k})$ Zustände benötigt. Ein deterministischer k-reversalbeschränkter Automat mit $O(N^{1/k}\log^2 N)$ Zuständen kann L erkennen. $N = \Theta(2^{n/2})$ ist die Größe des minimalen Einweg dfa für L.*

Beweis: Bei der Sprache handelt es sich um $L = \{xy | n/2 = |x| = |y| \wedge x = y\}$. Jeder k-kreuzungsbeschränkte Automat mit s Zuständen für L kann von einem k Runden Protokoll simuliert werden, in dem A x erhält und B y. Dabei findet in jeder Runde eine Kommunikation von $\lceil \log s \rceil$ statt. Da aber die nichtdeterministische Kommunikationskomplexität des Gleichheitsproblems für Worte der Länge $n/2$ auch $n/2$ ist, ist $\lceil \log s \rceil \cdot k \geq n/2$ und somit $s \geq \Omega(2^{n/(2k)})$.
Andererseits kann ein deterministischer Automat $n/(2k)$ Bits von x lesen, zur richtigen Position in y laufen, die gelesenen Bits vergleichen usw., bis alle Zeichen von x und y verglichen sind. Hierzu sind also $n/(2k)$ Bits „Speicher“ sowie zwei Zähler mit zusammen $2\log n$ Bits ausreichend. Daher kann man einen deterministischen k-reversalbeschränkten Automaten für L bauen, der $O(2^{n/(2k)}n^2)$ Zustände hat.
Die Größe eines minimalen dfa für L ist $N = \Theta(2^{n/2})$. □
Die Sprache L kann also von k-kreuzungsbeschränkten nichtdeterministischen Automaten effizienter erkannt werden als von $(k-1)$-kreuzungsbeschränkten nichtdeterministischen Automaten.
Diese Hierarchie gilt bis ungefähr $k = \Theta(\sqrt{\log N / \log\log N})$, allerdings sind alle solchen Unterschiede polynomiell. Das ändert sich im Fall von beschränktem Nichtdeterminismus.

Theorem 4.5 *Für alle s, k, n mit $n \geq s$ gibt es eine Sprache $L_{s,k} \subseteq \{0,1\}^n$, welche von einem deterministischen k-reversalbeschränkten Automaten mit $kn^{O(s)}$ Zuständen erkannt wird. Zum Erkennen von $L_{s,k}$ benötigt jeder nichtdeterministische $(k-1)$-kreuzungsbeschränkte Automat mit nur s nichtdeterministischen Ratebits $2^{\Omega(n/(s^2k^3\log n))}$ Zustände.*

Beweis: Wir verwenden f_k^{3200s} aus Definition 4.2. Die Eingaben werden als Worte geschrieben, bei denen in der ersten Hälfte die Pointer der linken Seite des Graphen stehen, und in der zweiten Hälfte die Pointer der rechten

Seite, die s Startknoten stehen ganz am Anfang der Eingabe. Ein deterministischer k-reversalbeschränkter Automat kann so arbeiten, daß je s Pointer der Pfade gelesen werden, dann zu den entsprechenden Positionen in der anderen Hälfte der Eingabe gelaufen wird, und dort die s Nachfolgepointer gelesen werden. k mal wird dabei die Bewegungsrichtung gewechselt. Es reichen $kn^{O(s)}$ Zustände.
Ein nichtdeterministischer $k-1$-kreuzungsbeschränkter Automat mit s Ratebits und $size$ Zuständen kann von einem $k-1$ Runden Protokoll mit derselben Anzahl nichtdeterministischer Bits simuliert werden. Die Kommunikation einer Runde ist $\lceil \log size \rceil$. Nach Theorem 4.2 ist die gesamte Kommunikation, die zur Berechung von f_k^{3200s} notwendig ist, mindestens $\Omega(n/(s^2k^2 \log n))$. Es gilt also $(k-1)\lceil \log size \rceil \geq \Omega(n/(s^2k^2 \log n))$ und somit $size \geq 2^c$ mit $c = \Omega(n/(s^2k^3 \log n))$. □
Sei z.B. N die Größe deterministischer k-reversalbeschränkter deterministischer Automaten für f_k^{3200s}, wobei $s = \sqrt{\log N}$ und $k = 2^{\epsilon\sqrt{\log N}}$ seien, und n so gewählt sei, daß die obere Schranke bei k Reversals genau N ist. Dann ist die untere Schranke für nichtdeterministische $(k-1)$-kreuzungsbeschränkte Automaten mit s Ratebits immer noch $2^{2^{\Omega(\sqrt{\log N})}}$. Man erhält also fast exponentielle Unterschiede für einen relativ großen Wertebereich.

4.3.3 Die Länge nichtdeterministischer Formeln

Zunächst sei bemerkt, daß ein nichtdeterministischer Schaltkreis sehr leicht in eine nichtdeterministische Formel umgewandelt werden kann, ohne daß dabei die Größe um mehr als einen konstanten Faktor ansteigt. Hierzu muß man lediglich für alle Gatter ihren Wert raten und durch eine Formel die Konsistenz des Geratenen prüfen. Somit sind untere Schranken für generelle nichtdeterministische Formeln sehr schwer zu zeigen, da sogar nichtlineare untere Schranken für die Größe deterministischer Schaltkreise für konkrete Funktionen bisher unbekannt sind. Formeln mit beschränktem Nichtdeterminismus sind aber zugänglicher.
Wir führen wieder eine Variation der Nečiporuk Methode ein, diesmal mittels nichtdeterministischer Kommunikation:

Definition 4.3 *Sei f eine Boolesche Funktion mit n Eingabevariablen und $y_1 \dots y_k$ eine disjunkte Partition der Eingaben in k Blöcke.*
Spieler B erhalte alle Eingaben in y_i und A alle anderen Eingaben. Die nichtdeterministische Einweg Kommunikationskomplexität mit s Ratebits von f unter dieser Eingabeaufteilung heiße $N_s^{(1)}(f_i)$. Definiere die s-nichtdeterministische Nečiporuk Funktion als $1/4 \sum_{i=1}^{k} N_s^{(1)}(f_i)$.

Lemma 4.8 *Die s-nichtdeterministische Nečiporuk Funktion ist eine untere Schranke für die Länge nichtdeterministischer Boolescher Formeln mit s Ratebits.*

BEWEIS: Für jede Partition $y_1, \ldots, y_k$ der Eingaben zeigen wir, wie eine nichtdeterministische Formel F mit s Ratebits von k Kommunikationsspielen simuliert werden kann. Sei F_i der Subbaum von F, der als Blätter die Variablen in y_i enthält und dazu alle Pfade von diesen Blättern bis zur Wurzel. Wir wollen die Formel in Spiel i so simulieren, daß die nichtdeterministische Einweg Kommunikation mit s Ratebits durch die Anzahl der Blätter in F_i beschränkt ist.
Gegeben ist eine nichtdeterministische Formel. Spieler A weiß alle Eingaben außer denen in y_i. Spieler A rät eine Belegung der nichtdeterministischen Eingaben. Die restliche Berechnung ist deterministisch.
V_i enthalte die Knoten in F_i, welche 2 Vorgänger in F_i haben und P_i alle Pfade, welche in V_i oder an einem Blatt anfangen, und in V_i oder an der Wurzel aufhören, aber keinen weiteren Knoten aus V_i enthalten. Es reicht aus, wenn A für jeden solchen Pfad 2 Bits sendet, die angeben, ob das letzte Gatter des Pfades $0, 1, g$, oder $\neg g$ berechnet, für die Funktion g des ersten Gatters des Pfades. Dann kann B die Formel alleine auswerten.
Es gibt aber höchstens $2|V_i| + 1$ solche Pfade, da der fan-in der Formel 2 ist. Damit ist die gesamte Kommunikation höchstens $4|V_i| + 2$. Die Menge der Blätter L_i mit Variablen aus y_i hat $|V_i| + 1$ Elemente, und daher ist $N_s^{(1)}(f_i) \leq 4|V_i| + 2 < 4|L_i|$ und $1/4 \sum_i N_s^{(1)}(f_i)$ ist eine untere Schranke für die Länge der nichtdeterministischen Formel F, die mindestens $\sum_i |L_i|$ beträgt. □

Definition 4.4 *$AD_{n,s}$ bezeichne die folgende Sprache (für $1 \leq s \leq n$):*

$$AD_{n,s} = \{(x_1, \ldots, x_{n+1}) | \forall i : x_i \in \mathcal{P}(n^3, s),$$
$$x_i \text{ ist in sortierter Reihenfolge geschrieben}$$
$$\wedge \exists i : |\{j | j \neq i; x_i \cap x_j \neq \emptyset\}| \geq s\}.$$

Theorem 4.6 *Jede nichtdeterministische Formel mit s Ratebits für $AD_{n,20s}$ hat eine Länge von mindestens $\Omega(n^2 s \log n)$.*
$AD_{n,s}$ kann von einer nichtdeterministischen Formel der Länge $O(ns^2 \log n)$ berechnet werden, welche $O(s \log n)$ Ratebits benutzt (wenn $s \geq \log n$).

BEWEIS: Für die untere Schranke verwenden wir den obigen Ansatz. Wir betrachten die $n + 1$ Aufteilungen der Eingabe, bei denen B die Menge x_i erhält und A alle anderen Mengen. Die nun zu berechnende Funktion ist die Funktion $OD_{n,s}$ aus Definition 4.1. Da die untere Schranke von Theorem 4.1 für das Problem $\Omega(ns \log n)$ ist, folgt eine untere Schranke für die Länge der Formel von $\Omega(n \cdot ns \log n)$ mit Lemma 4.8.
Für die obere Schranke betrachten wir folgenden Ansatz: die Formel rät (in Binärdarstellung) eine Zahl i mit $1 \leq i \leq n+1$ und Paare $(j_1, w_1), \ldots, (j_s, w_s)$, wobei $1 \leq j_k \leq n + 1$ und $1 \leq w_k \leq n^3$ für alle $k = 1, \ldots, s$. Das i indiziert eine Menge, und die Paare stellen Zeugen dar, daß Menge i und Menge j_k sich auf Element w_k schneiden.

Die Formel prüft nun folgendes nach. Zuerst wird getestet, ob alle Mengen in der Eingabe aus je s verschiedenen sortierten Elementen bestehen. Dazu reichen ns Vergleiche der Form $x_i^j < x_i^{j+1}$ aus, welche mit je $O(\log^2 n)$ Gattern zu realisieren sind. Wegen $s \geq \log n$ also mit insgesamt $O(ns^2 \log n)$ Gattern. Dann wird getestet, ob $j_1 < \cdots < j_s$. Das stellt sicher, daß wir Zeugen für s verschiedene Mengen haben. Außerdem wird $i \neq j_k$ für alle k geprüft. Dann wird getestet, ob für alle $1 \leq l \leq n+1$ gilt, daß wenn $l = i$ ist, alle geratenen Elemente in x_l liegen. Für alle $1 \leq l \leq n+1$ und $1 \leq k \leq s$ wird auch getestet, ob $l = j_k$ impliziert, daß $w_k \in x_l$. Alle diese Tests können simultan von einer Formel der Länge $O(ns^2 \log n)$ durchgeführt werden. □
Für $0 < \epsilon \leq 1/2$ sei $s = n^{\frac{\epsilon}{1-\epsilon}}$, dann ist die untere Schranke für beschränkt nichtdeterministische Formeln $\Omega(N^{2-\epsilon}/\log^{1-\epsilon} N)$ bei $N^\epsilon/\log^\epsilon N$ erlaubten Ratebits. $O(N^\epsilon \log^{1-\epsilon} N)$ Ratebits erlauben Formeln der Länge $O(N^{1+\epsilon}/\log^\epsilon N)$.

4.3.4 Monotone Schaltkreistiefe

Zuerst wenden wir uns Schaltkreisen mit unbeschränktem Nichtdeterminismus zu. Hier kann die Tiefe stets effizient reduziert werden.

Theorem 4.7 *Ein monotoner nichtdeterministischer Schaltkreis mit c Gattern kann in eine äquivalente monotone nichtdeterministische Formel mit Tiefe $\log c + O(1)$ und $O(c)$ Gattern umgewandelt werden. Bei unbeschränktem fan-in reicht Tiefe 2 bei Größe $O(c)$.*

Beweis: Gegeben sei ein monotoner nichtdeterministischer Schaltkreis C mit c Gattern. Es ist eine Formel zu konstruieren. Die Formel benutzt c Ratebits, welche entweder zu Gattern von C korrespondieren oder zu nichtdeterministischen Eingaben von C. Für jedes zu einem Gatter gehörige Ratebit führt die Formel einen Test aus, der prüft, ob das Gatter richtig geraten wurde bzw. ob der geratene Wert nicht unzulässigerweise zum Akzeptieren führt. Die ganze Formel ist als ein UND über solche Tests organisiert. Die Tests sind entweder ein ODER von Variablen, oder ein fan-in 2 UND von ODER Gattern.
Für ein ODER Gatter g_i in C mit Eingaben g_j und g_k gibt es also geratene Werte c_i und c_j und c_k (wobei c_j und c_k auch Eingaben von C seien können). Es soll getestet werden, ob $c_i \Rightarrow (c_j \vee c_k)$. Das ist äquivalent zu $\neg c_i \vee c_j \vee c_k$. Wir erinnern daran, daß negierte Ratebits erlaubt sind.
Für ein UND Gatter g_i in C mit Eingaben g_j und g_k gibt es geratene Werte c_i und c_j und c_k (wobei c_j und c_k auch Eingaben von C seien können). Es soll getestet werden, ob $c_i \Rightarrow (c_j \wedge c_k)$. Das ist äquivalent zu $\neg c_i \vee (c_j \wedge c_k)$ was wiederum äquivalent ist zu $(\neg c_i \vee c_j) \wedge (\neg c_i \vee c_k)$. Daher kann der Test als ein UND von ODER Gattern implementiert werden.
VerUNDet man alle Tests und die Variable für das Ausgabegatter von C, so erhält man ein UND von ODER Gattern von Eingaben und Ratebits.

Keine Eingabe ist negiert. Das UND hat fan-in höchstens $2c+1$, alle ODER Gatter haben fan-in ≤ 3. Ersetzt man das UND durch einen Baum von UND Gattern mit fan-in 2, so erhält man $O(c)$ Gatter und Tiefe $\log c + O(1)$.
Sei x eine Eingabe, die von C bei Ratebits a akzeptiert wird. Setzt die Formel die Ratebits von C wie a und rät zudem alle Gatter von C richtig, so akzeptiert sie, da alle Test bestanden sind. Wird ein x von C für kein Ratewort akzeptiert, so kann die Formel x nur akzeptieren, wenn ein Gatter falsch geraten wird. Dank der Tests kann ein falsch geratenes Gatter aber nur dazu führen, daß verworfen wird, nicht aber dazu, daß akzeptiert wird. Also wird x nie akzeptiert. □
Es ist leicht zu sehen, daß ein Schaltkreis mit s nichtdeterministischen Bits deterministisch gemacht werden kann, wobei die Tiefe additiv um s steigt. Dazu bildet man das ODER über alle 2^s Möglichkeiten. Hat man also starke untere Schranken für deterministische Schaltkreise, so erhält man auch untere Schranken für beschränkten Nichtdeterminismus. Aber was ist, wenn viel mehr nichtdeterministische Bits erlaubt sind als die Tiefe angibt?

Theorem 4.8 *Seien d, n so, daß $\sqrt{n} \geq d \geq \log n$. Dann gibt es eine explizite Boolesche Funktion g_n^d auf n Variablen sowie eine Konstante $\epsilon > 0$ mit den folgenden Eigenschaften: g_n^d kann von einer monotonen deterministischen Formel mit Tiefe $O(d)$ berechnet werden, und jeder nichtdeterministische monotone Schaltkreis mit $\epsilon n/d$ Ratebits braucht Tiefe $\Omega(d)$. g_n^d kann in monotoner Tiefe $O(\log n)$ mit n/d Ratebits berechnet werden.*

Beweis: Die Eingabe der Funktion g_n^d wird als wie folgt aufgebaut betrachtet: es gibt n/d^2 Blöcke, die je einer Adjazenzmatrix für einen ungerichteten bipartiten Graphen auf $2d$ Knoten entsprechen. Allerdings bedeutet eine 0, daß eine Kante vorhanden ist, und eine 1, daß eine Kante nicht vorhanden ist. Die Funktion entscheidet nun, ob für alle Graphen gilt, daß sie *kein* perfektes Matching enthalten. Das ist die direkte Summe des dualen Problems zum (perfekten) Matching Problem. Offensichtlich ist die Funktion monoton und kann von deterministischen monotonen Schaltkreisen in Tiefe $O(d)$ berechnet werden. Dazu kann man zuerst ein UND über die n/d^2 Instanzen verwenden, was zu Tiefe höchstens $\log n \leq d$ führt. Für jede Instanz muß das duale Problem für (perfektes) Maching gelöst werden, was genauso schwer ist wie (perfektes) Matching selbst. Es ist also zu zeigen daß, gegeben einen bipartiten Graphen mit d Knoten, in Tiefe $O(d)$ entschieden werden kann, ob der Graph ein perfektes Matching hat. Folgende Tatsache [H35] ist dazu nützlich.

Fakt 4.4 (Hall) *Sei $G = (U \cup V, E)$ ein bipartiter Graph. G hat ein perfektes Matching von U nach V dann und nur dann, wenn für alle Teilmengen $S \subseteq U$ gilt $|\Gamma(S)| \geq |S|$, wobei $\Gamma(S)$ die Menge der Nachbarn von Knoten in S bezeichnet.*

Es reicht also aus, zu testen, ob für alle Teilmengen S der linken Knoten die Nachbarschaft mindestens so groß wie S ist. Damit ergibt sich ein UND von 2^d Tests (Tiefe d). Für jeden Test muß die Größe der Nachbarschaft von einem festen S mit der festen Größe $t = |S|$ verglichen werden. Sei $(a^1, \ldots, a^d)$ der Inzidenzvektor von S ($a^i = 1$ bedeutet, daß der Knoten i in der Teilmenge liegt). $e_G^{i,j}$ seien Variablen für die Kanten von G ($e_G^{i,j} = 1$ bedeutet, daß die Kante im Graphen ist). Es wird getestet, ob

$$\sum_{j=1}^{d} \left(\bigvee_{i=1}^{d} a^i \wedge e_G^{i,j} \right) \geq t.$$

Die Threshold Funktion $\sum y_i \geq t$ kann effizient von monotonen Schaltkreisen berechnet werden, siehe [V84].

Fakt 4.5 *Für jedes t gibt es einen monotonen Schaltkreis der Tiefe $O(\log d)$, der eine Boolesche Eingabe $x_1, \ldots, x_d$ genau dann akzeptiert, wenn $\sum x_i \geq t$.*

Somit erhalten wir einen deterministischen monotonen Schaltkreis der Tiefe $O(d)$ für g_n^d.
Nun betrachten wir nichtdeterministische monotone Schaltkreise. Mit n/d Ratebits kann man n/d^2 Teilmengen raten, jede davon eine Teilmenge der linken Knoten eines der n/d^2 Eingabegraphen. Für jede Teilmenge ist zu prüfen, ob die Menge der Nachbarn kleiner als die geratene Menge ist. Ein solcher Test reicht wegen Halls Theorem aus. Um dies durch eine monotone Formeln zu implementieren tuen wir folgendes: Für jeden Eingabegraphen G sei $a_G = (a_G^1, \ldots, a_G^d)$ die geratene Teilmenge als Inzidenzvektor repräsentiert ($a_G^i = 1$ bedeutet, daß der Knoten i in der Teilmenge liegt). $e_G^{i,j}$ kodiert die Kanten von G ($e_G^{i,j} = 0$ bedeutet, daß die Kante im Graphen ist).

$$\bigvee_{k=0}^{d} \left[\sum_{i=1}^{d} \left(\bigwedge_{j=1}^{d} \neg a_G^j \vee e_G^{j,i} \right) \geq d - k + 1 \wedge k = \sum_{i=1}^{d} a_G^i \right]$$

testet nun, ob für die Teilmenge a_G von linken Knoten in G die Anzahl der Nachbarn kleiner ist als die Größe der Teilmenge selbst. Akzeptiert der Test, so bezeugt dies, daß G kein perfektes Matching hat. Die Tiefe der Tests ist $O(\log d)$ wegen der monotonen Formeln für Threshold Funktionen aus [V84]. Zusammen mit dem UND über alle Instanzen G erhalten wir Tiefe $\log n + O(\log d)$.
Nun zu der unteren Schranke für nichtdeterministische monotone Schaltkreise mit beschränktem Nichtdeterminismus. Für das bipartite perfekte Matching Problem wie auch für dessen duales Problem gilt eine untere Schranke für die Tiefe deterministischer monotone Schaltkreise von $\Omega(d)$ für Graphen mit $2d$ Knoten, siehe [RW92].

Fakt 4.6 *Jeder deterministische monotone Schaltkreis, welcher entscheidet, ob ein gegebener bipartiter Graph ein perfektes Matching hat, hat Tiefe* $\Omega(d)$.

Wir haben also zu zeigen, daß bei der Berechnung von g_n^d eine Anzahl von $s = \epsilon n/d$ Ratebits nicht wesentlich hilft, die Tiefe zu reduzieren (für eine hinreichend kleine Konstante $\epsilon > 0$).
Gegeben sei also ein monotoner Schaltkreis F der Tiefe t mit s Ratebits für g_n^d. Wir konstruieren einen monotonen deterministischen Schaltkreis mit Tiefe $t + \epsilon d + O(\log n)$ für das bipartite perfekte Matching Problem auf Graphen mit $2d$ Knoten. Das ergibt mit Fakt 4.6 eine untere Schranke von $t = \Omega(d)$, für ein hinreichend kleines konstantes $\epsilon > 0$.
Wir fixieren die Ratebits von F und erhalten 2^s deterministische Schaltkreise. Einer von diesen akzeptiert einen Bruchteil von $1/2^s$ aller 1-Eingaben gemessen gemäß der Gleichverteilung auf allen 1-Eingaben. Dann aber muß es in der direkten Summe der n/d^2 Instanzen eine Position i geben, an der für irgendwelche Belegungen der anderen Positionen insgesamt mindestens $1/2^{\epsilon d}$ aller bipartiten Graphen mit $2d$ Knoten ohne ein perfektes Matching vorkommen. D.h. für so viele der 1-Eingaben des perfekten Matching Problems gibt es $n/d^2 - 1$ andere solche Eingaben, so daß der gesamte Vektor akzeptiert wird. Ansonsten wäre es der Fall, daß weniger als $1/2^s$ aller 1-Eingaben von g_n^d akzeptiert würden. Weil der Schaltkreis aber monoton ist, können wir die $n/d^2 - 1$ Graphen durch leere Graphen ersetzen (nicht vorhandene Kanten sind durch 1 kodiert). Gibt es also für einen Graphen G an Position i eine Belegung der restlichen Eingaben, welche akzeptiert wird, so wird auch der Vektor aus G und leeren Graphen akzeptiert.
Fixiert man also $n/d^2 - 1$ Positionen zu leeren Graphen, so erhält man einen monotonen deterministischen Schaltkreis, der einen Anteil von $1/2^{\epsilon d}$ aller 1-Eingaben akzeptiert (für das duale von perfektem Matching).
Da ein nichtdeterministischer Schaltkreis auch auf einer beliebigen Teilmenge der Eingaben korrekt arbeitet, kann dieses Vorgehen iteriert werden auf der Menge der 1-Eingaben, die noch nicht akzeptiert sind. Genauer gesagt betrachten wir nun die Menge von 1-Eingaben, die aus n/d^2 langen Vektoren aus noch nicht bedeckten 1-Eingaben für Matching bestehen. Wieder wird ein großer Bruchteil akzeptiert, und wieder muß an einer Position ein großer Bruchteil der noch nicht überdeckten Graphen vorkommen.
Auf diese Art ist ein Überdeckungsproblem zu lösen. Das Universum ist die Menge aller 1-Eingaben des zum perfekten Matching dualen Problems und hat die Größe $O(2^{d^2})$. In jedem Schritt kann ein Anteil von $1/2^{\epsilon d}$ aller verbleibenden 1-Eingaben überdeckt werden. Daher ist nach $O(\ln(2^{d^2})2^{\epsilon d})$ Schritten eine ebensogroße Menge von deterministischen monotonen Schaltkreisen gefunden, die alle 1-Eingaben des dualen von perfektem Matching überdeckt, aber keine 0-Eingabe. Ein ODER Baum über alle diese Schaltkreise ergibt einen monotonen deterministischen Schaltkreis für das duale von perfektem Matching der Tiefe $t + \epsilon d + O(\log d)$. Um das perfekte Mat-

ching Problem selbst zu lösen tauscht man UND und ODER Gatter miteinander aus. □

Im s,t-connectivity Problem sind ein gerichteter Graph sowie zwei ausgezeichnete Knoten s,t gegeben. Es ist zu akzeptieren, wenn es im Graphen einen Weg von s nach t gibt.
Das folgende ist bekannt aus [KW90].

Fakt 4.7 *Jeder deterministische monotone Schaltkreis für s,t-connectivity auf Graphen mit n Knoten hat Tiefe $\Omega(\log^2 n)$.*

Theorem 4.9 *Sei d die Tiefe optimaler monotoner Schaltkreise für s,t-connectivity mit $(n/k)\log n$ nichtdeterministischen Ratebits.*
Dann ist $\Omega(\log^2 k + \log n) = d = O(\log n \log k)$.

Beweis: Für die untere Schranke betrachten wir einen beliebigen nichtdeterministischen monotonen Schaltkreis der Tiefe d, der das s,t-connectivity Problem löst und $(n/k)\log n$ Ratebits verwendet.
Die untere Schranke $\Omega(\log n)$ ist trivial, da das Problem von allen Eingaben abhängt. Im Fall $\log^2 k = O(\log n)$ folgt die untere Schranke also insgesamt. Ist ansonsten $\log^2 k \geq \omega(\log n)$, so dürfen wir annehmen, daß $r = \epsilon(n/k)\log^2 k$ Ratebits verwendet werden für ein beliebig kleines $\epsilon > 0$.
Wir teilen die n Knoten des Eingabegraphen in n/k disjunkte Mengen von je k Knoten auf. Dabei halten wir für jede solche Menge ein Paar s_i, t_i von Knoten fest und setzen die Kanten t_i, s_{i+1} für alle i ein, verbieten weitere Kanten zwischen den Mengen. So muß der Schaltkreis die direkte Summe von n/k connectivity Problemen auf Graphen mit je k Knoten lösen. Dabei ist $s = s_1$ und $t = t_{n/k}$, und es gibt einen Pfad von s nach t genau dann, wenn es für alle i einen Pfad von s_i nach t_i gibt.
Wieder erhalten wir durch Fixieren der Ratebits einen deterministischen monotonen Schaltkreis mit Tiefe d, der einen Bruchteil von $1/2^r$ mit $r = \epsilon(n/k)\log^2 k$ der 1-Eingaben erkennt. Auf einer der Positionen der direkten Summe müssen die akzeptierten Graphen über einen Bruchteil von $1/2^{\epsilon \log^2 k}$ der 1-Eingaben von s,t-connectivity auf Graphen mit k Knoten variieren, d.h. es gibt für jeden der Graphen Vektoren von $n/k-1$ anderen Graphen, so daß der gesamte Vektor akzeptiert wird. Ansonsten gäbe es weniger als $1/2^r$ akzeptierte 1-Eingaben insgesamt, ein Widerspruch.
Statt der unbekannten Graphen kann man vollständige Graphen an den $n/k-1$ Positionen einsetzen und erhält so einen deterministischen monotonen Schaltkreis, der $1/2^r$ aller 1-Eingaben für s,t-connectivity auf k Knoten akzeptiert.
Dasselbe Vorgehen kann wieder iteriert werden, indem man nur noch die Menge der Eingaben betrachtet, die aus n/k langen Vektoren aus bisher nicht überdeckten Graphen bestehen. Wieder gelingt es, eine Position mit $1/2^{\epsilon \log^2 k}$ akzeptierten Graphen zu finden und so kann das folgende Überdeckungsproblem gelöst werden: das Universum besteht aus allen 1-Eingaben

für s,t-connectivity auf k Knoten. In jedem Schritt werden $1/2^{\epsilon \log^2 k}$ der nicht überdeckten Graphen überdeckt. Damit erhält man eine Menge von $2^{\epsilon \log^2 k} poly(k)$ deterministischen monotonen Schaltkreisen, welche alle 1-Eingaben überdecken. Durch ein ODER über diese erhalten wir einen Schaltkreis der Tiefe $d + \epsilon \log^2 k + O(\log k)$ für s-t-connectivity und schließen, daß $d = \Omega(\log^2 k)$ (mit [KW90]).
Für die obere Schranke beobachten wir folgendes. In Tiefe $O(\log n \log k)$ kann man für jedes Paar von Knoten das Prädikat „haben Abstand höchstens k“ berechnen, indem man die $k$te Potenz von Adjazenzmatrix plus Einheitsmatrix bestimmt. Also reicht es aus, n/k Knoten zu raten, und für je zwei aufeinanderfolgende zu testen, ob sie einen Abstand von höchstens k haben. Weiterhin muß man testen, ob s und t in der richtigen Reihenfolge unter den geratenen Knoten sind. Gibt es einen Weg von s nach t, so wird ein solcher auch ein zum Akzeptieren führendes Ratewort ergeben, gibt es keinen Weg, kann kein Ratewort zum Akzeptieren führen. Die Tiefe ist $O(\log n \log k)$, der Schaltkreis ist monoton, benutzt $(n/k) \log n$ nichtdeterministische Ratebits.
□

Theorem 4.10 *Für alle $k \geq 3$ und $s \geq n$ gibt es eine Funktion q_k^s mit $N = \Theta(sn^{k-1})$ Eingaben, die von deterministischen monotonen Formeln mit fan-in $O(s)$, Tiefe k und Größe $O(N)$ berechnet werden kann. Jeder monotone Schaltkreis mit Tiefe $k-1$, unbeschränktem fan-in und s/k nichtdeterministischen Bits für q_k^s hat Größe $2^{\Omega((N/s)^{1/(k-1)}/k)}$.*
q_k^s kann auch berechnet werden von monotonen Schaltkreisen mit unbeschränktem fan-in, $O(s \log n)$ nichtdeterministischen Bits, Tiefe $k-1$ und Größe $O(N)$ (mit der Ausnahme von $k = 3$, wo die Größe $O(N \log N)$ ist).

Beweis: Die Funktion q_k^s ist definiert durch eine Formel, welche sie berechnet. Die Formel ist ein alternierender UND-ODER Baum der Tiefe k, bei der das oberste Gatter ein UND mit fan-in $(1/\epsilon)s$ ist und alle anderen Gatter fan-in n haben. Es gibt also $O(sn^{k-1})$ Boolesche Eingaben, ϵ wird später festgelegt. Man kann die Funktion betrachten als eine direkte Summe von $(1/\epsilon) \cdot s/n$ Funktionen q_k definiert durch einen alternierenden UND-ODER-Baum der Tiefe k, bei dem das oberste UND Gatter und alle anderen Gatter fan-in n haben.
Die obere Schranke für deterministische monotone Formeln der Tiefe k ist somit Teil der Definition.
Für die untere Schranke kann man argumentieren, daß ein monotoner Schaltkreis mit Tiefe $k-1$ und Größe t, der s/k Ratebits liest, zu einem monotonen deterministischen Schaltkreis mit Größe t, Tiefe $k-1$ führt, der einen Bruchteil von mindestens $1/2^{\epsilon n/k}$ aller 1-Eingaben von q_k unter einer beliebigen Verteilung akzeptiert, ansonstens hätte jeder solche Schaltkreis mit fixierten Ratebits weniger als $(1/2^{\epsilon n/k})^{(1/\epsilon)s/n}$ aller 1-Eingaben von q_k^s akzeptiert.

Durch Iteration wie oben erhält man ein ODER von $O(2^{\epsilon n/k} n^{k-1})$ solchen Schaltkreisen und damit einen deterministischen monotonen Tiefe k Schaltkreis für q_k mit einem ODER an der Spitze. Dies ergibt die untere Schranke mit folgendem Resultat aus [KPY84] oder [NW93].

Fakt 4.8 *Jeder monotone Schaltkreis mit unbeschränktem fan-in und Tiefe k mit einem ODER an der Spitze, welcher q_k berechnet, hat eine Größe von $2^{\Omega(n/k)}$.*

Wir haben einen deterministischen monotonen Schaltkreis der Tiefe k und Größe $O(2^{\epsilon n/k} n^{k-1} \cdot t)$ mit einem ODER an der Spitze. Somit ist $t = 2^{\Omega(n/k)}$, wenn ϵ als genügend kleine Konstante gewählt ist, und $n = (N/s)^{1/(k-1)}$.
Nun zur oberen Schranke für nichtdeterministische Schaltkreise. Wenn $k \geq 4$ rate man $a_1, \ldots, a_{s/\epsilon}$ aus jeweils $\log n$ Bits. $a_i = j$ bedeute dabei, daß das $j$te Kind $F_{i,j}$ von dem $i$ten Gatter unter den Kindern des obersten Gatters des q_k^s definieren Schaltkreises 1 ausgibt. So muß also geprüft werden, ob für alle i, j gilt, daß $a_i = j$ impliziert, daß auch der Subschaltkreis $F_{i,j}$ 1 ausgibt. D.h. $\bigwedge_{i,j} \neg(a_i = j) \vee F_{i,j}$. Damit erhält man einen Schaltkreis der Tiefe k. Um die Tiefe weiter zu reduzieren, kann der Term $\neg(a_i = j)$ (der ein ODER über Ratebits ist) zu jedem der ODER Kinder der Wurzel von $F_{i,j}$ addiert werden und von seiner ursprünglichen Position entfernt werden. Ist ein solcher Term wahr, dann ist der (neue) Subschaltkreis $F_{i,j}$ wahr, ist er falsch, so ist der neue Subschaltkreis genau dann wahr, wenn der (alte) $F_{i,j}$ wahr ist. Das Zufügen des Terms zu den Kindern erhöht die Größe asymptotisch nicht, da alle diese Kinder jeweils bereits N Kinder haben. Wenn man nun die obersten 3 UND Ebenen verschmilzt, so erhält man Tiefe $k-2$ und Größe $O(N)$.
Im Fall $k = 3$ wird genau dasselbe getan, aber wenn man den erwähnten Term zu den Kindern der Wurzeln der $F_{i,j}$ addiert, so muß man für jedes Kind (das eine Variable ist) ein ODER einsetzen, mit Eingaben aus dem Term und dem Kind. Somit wird die Tiefe nur auf $k-1$ reduziert, und und die Größe ist $O(N \log N)$. Eine Tiefenreduktion auf $k-2 = 1$ wäre in diesem Fall unmöglich. □
Das obige Resultat ist nach seiner Publikation noch verbessert worden (siehe [PRV99]). Die verbesserte untere Schranke enthält einen zusätzlichen Faktor von $\log^{(k)} n$ im Exponenten. Die Verbesserung ist eine direkte Konsequenz von Fakt 3.6, einer Reduktion in [NW93], welche Fakt 4.8 aus der deterministischen Rundenhierarchie für Pointer Jumping herleitet, und der obigen Konstruktion.

Kapitel 5

Quantenmechanische Kommunikation

5.1 Überblick

Das Forschungsgebiet der Quantenrechner befaßt sich mit Berechnungsmodellen, welche den Regeln der Quantenmechanik folgen. Klassische Berechnungsmodelle, wie Turingmaschinen oder Boolesche Schaltkreise scheinen nicht in der Lage zu sein, das Verhalten von quantenmechanischen Systemen effizient (d.h. mit polynomiellem Mehraufwand) zu simulieren. Deshalb haben Feynman [F82] und Deutsch [D85] vorgeschlagen, quantenmechanische Rechner zu konstruieren, Deutsch bereits mit der Absicht, bestimmte klassische Berechnungsprobleme eventuell effizienter als auf klassischen Rechnern lösen zu können. Die Untersuchung von Quantenrechnern hat in letzter Zeit viel Aufmerksamkeit auf sich gezogen, da wichtige Resultate erzielt worden sind. Ein Durchbruch in diesem Gebiet ist der Algorithmus von Shor [S97], der es erlaubt, natürliche Zahlen unter Verwendung eines Quantenrechners in polynomieller Zeit in ihrer Länge in Primfaktoren zu zerlegen. Die effiziente Lösung des Faktorisierungsproblems wird als ein Indiz gewertet, daß Quantenrechner nicht mit bloß polynomiellem Mehraufwand von klassischen Rechnern simuliert werden können.
Ein weiteres wichtiges Resultat des Gebietes ist Grovers Algorithmus, der es erlaubt, ein Element in einer unsortierten Datenbank mit n Elementen mit nur $O(\sqrt{n})$ Anfragen an die Datenbank zu finden [G96].
Die üblichen universellen Berechnungsmodelle für Quantenrechner sind die Quanten Turing Maschine und Quanten Schaltkreise (siehe [Y93] und [BV97]). Um besser zu verstehen, welche Möglichkeiten und Einschränkungen quantenmechanisches Rechnen hat, sind verschiedene klassische beschränkt mächtige Berechnungsmodelle in quantenmechanischen Varianten betrachtet worden, so endliche Quanten Automaten, Quanten Entscheidungsbäume und Quanten Kommunikationsprotokolle (siehe [KW97], [G96], [Y93]).

Die vielseitigen Anwendungen der Kommunikationskomplexität, vor allem in unteren Schranken, sind eine Hauptmotivation, das Modell auch im quantenmechanischen Kontext zu studieren. In einem Quantenprotokoll (wie in [Y93] definiert) tauschen die Spieler sogenannte Qubits aus. Ein Qubit entspricht einem Vektor der Norm 1 in $\mathbb{C}^2$, wobei die Basisvektoren des Raums den klassischen Booleschen Werten zugeordnet werden. Also erlaubt ein Qubit die Darstellung einer normierten Linearkombination von 0 und 1.
Ein geringfügig hiervon verschiedenes Szenario wird in [CB97] und [CDNT97] vorgeschlagen. Hier besitzen die Spieler Zugriff auf eingabeunabhängige Qubits, deren Zustand mit dem der Qubits des jeweils anderen Spielers „verschränkt“ (engl. entangled) sind. Aufgrund einer Technik mit dem Namen „superdense coding“, vorgeschlagen von [BW92] (siehe auch Abschnitt 5.2), ist es in diesem Modell möglich, 2 klassische Bits zu kommunizieren, indem nur ein Qubit ausgetauscht wird (allerdings wird dabei ein Paar von verschränkten Qubits „verbraucht“).
Das Hauptziel der Theorie der Quanten Kommunikationskomplexität ist es, zu ermitteln, eine wie große Effizienzsteigerung sich durch Quanten Protokolle im Vergleich zu klassischen Protokollen erzielen läßt. Hierbei unterscheidet man wieder die verschiedenen Akzeptanzmodi: Exakte Protokolle (ohne Fehler), Las Vegas Protokolle (ohne Fehler, aber nur erwartete Kommunikation wird gemessen), Protokolle mit beschränktem Fehler. Überblicke über Resultate zur Quanten Kommunikationskomplexität findet man in [T99] und [Kl00b].
In [BCW98] wird Grovers Suchalgorithmus auf das Disjunktheitsproblem angewendet und eine obere Schranke von $O(\sqrt{n}\log n)$ für die Quanten Kommunikationskomplexität mit beschränktem Fehler für dieses Problems gezeigt. Dies ergibt den größten bisher bekannten Unterschied zwischen quantenmechanischer und klassischer Kommunikationskomplexität für eine totale Funktion, die probabilistische Kommunikationskomplexität von Disjunktheit ist $\Omega(n)$ (siehe Fakt 2.9). Exponentielle Unterschiede zwischen der Quanten Kommunikation mit beschränktem Fehler und klassischer probabilistischer Kommunikation sind bisher nur für partielle Funktionen bekannt (siehe [R99]). Ein exponentieller Unterschied zwischen exakter Quanten und klassischer Las Vegas Kommunikation ist ebenfalls für eine partielle Funktion bekannt[BCW98].
Der Datenbanksuchalgorithmus von Grover entspricht im wesentlichen der Berechnung eines ODER über n Variablen, das mit $O(\sqrt{n})$ Anfragen an die Variablen gelöst wird. Eine Verallgemeinerung auf Prädikate, die durch konstant tiefe Formeln von UND und ODER Gattern auf den Black Box Variablen beschrieben werden, findet sich in [BCW98], auch hier ist die Beschleunigung fast quadratisch. In [BCWZ99] wird außerdem gezeigt, daß man mit ähnlichen Techniken einen fast quadratischen Komplexitätsunterschied zwischen Quanten Las Vegas und klassischen probabilistischen Black Box Problemen erhält. Durch eine Reduktion können bestimmte Kommu-

nikationsprobleme mittels der Black Box Algorithmen gelöst werden. Dadurch wird in [BCWZ99] ein polynomieller Unterschied zwischen Quanten Las Vegas Kommunikation und probabilistischer Kommunikation für eine bestimmte Relation gezeigt. Dieses Resultat wird in Abschnitt 5.4 insoweit verbessert, als ein polynomieller Unterschied zwischen denselben Modi, aber für eine totale Funktion gezeigt wird. Während die obere Schranke auf den Techniken von [BCWZ99] beruht, benutzt die untere Schranke die Argumente für Protokolle mit beschränktem Nichtdeterminismus aus [HrS96]. Es wird gezeigt, daß eine kleine, aber immer noch schwierige Teilmenge von $D_{n,s}$ (siehe Definition 2.7) effizient durch ein Quanten Las Vegas Protokoll entschieden werden kann. Die komplizierte Technik für die untere Schranke ist notwendig, da eine untere Schranke für probabilistische Kommunikation gezeigt werden soll, unter der Voraussetzung, daß die nichtdeterministische Komplexität sowohl der Funktion als auch ihres Komplements klein ist (wegen des Einsatzes von Grovers Suchalgorithmus). Außer in [HrS96] findet sich das einzige vergleichbare Resultat in [BL92], siehe Abschnitt 2.1.1. Desweiteren entwickeln wir eine untere Schranke für Quanten Las Vegas Kommunikationskomplexität in Abschnitt 5.4.1.
Die bisher entwickelten Quanten Protokolle für totale Funktionen, welche einen polynomiellen Speedup gegenüber klassischen Protokollen aufweisen, habe alle die Eigenschaft, viele Runden zu benötigen. Dies ist eine Folge des Einsatzes von Grovers Algorithmus. Wir untersuchen die Frage, wie effizient Protokolle sein können, wenn die Anzahl der Runden eingeschränkt wird. Das am stärksten beschränkte Modell ist Einweg Kommunikation. Wie wir bereits gesehen haben, gibt es im klassischen Fall wichtige Anwendungen der Einweg Kommunikationskomplexität für Boolesche Formeln und für Automaten.
Kremer [K95] hat Quanten Einweg Kommunikation untersucht und eine partielle Funktion definiert, die ein vollständiges Problem für die Klasse von Problemen mit Quanten Protokollen mit polylogarithmischer Kommunikation ist. Wir zeigen, daß die VC-Dimension eine untere Schranke für die Quanten Einweg Kommunikationskomplexität mit beschränktem Fehler ist. Hierbei erhalten wir eine gute untere Schranke unter Verwendung von Resultaten aus [ANTV99] und [Na99] über random access quantum codes. Weiterhin zeigen wir, daß für totale Funktionen und Einweg Protokolle die exakte Quanten Kommunikation genau so mächtig ist wie die deterministische Kommunikation und asymptotisch so effizient wie die Quanten Las Vegas Kommunikation. Ähnliche Resultate gelten auch, wenn verschränkte Qubits vorhanden sind.
Aus den obigen Resultaten schließen wir, daß Quanten Kommunikation von der erlaubten Interaktion abhängig ist. Beschleunigungen durch Quanten Kommunikation für totale Funktion scheinen Protokolle mit mehreren Runden zu erfordern, dies ist beweisbar für Las Vegas Protokolle und auch für Protokolle mit beschränktem Fehler für das Disjunktheitsproblem.

Dann wenden wir uns Anwendungen zu. Endliche Quanten Automaten (qfa) werden in [KW97] definiert. Folgendes ist über deses Modell bekannt, wobei die Automaten beschränkten Fehler haben: Für einige Sprachen können sie exponentiell kleiner als optimale probabilistische Automaten sein [AF98], sie können aber nur reguläre Sprachen erkennen [KW97]. Desweiteren gibt es endliche Sprachen, für die ein optimaler qfa exponentiell größer als der minimale dfa ist [ANTV99], [Na99]. Wie geben mit Hilfe der VC-Dimension ein kombinatorisches Werkzeug an, das untere Schranken für die Größe von qfa zu zeigen erlaubt. Ein weiteres Korollar ist, daß exakte qfa für eine Sprache niemals kleiner als der minimale dfa sein können, und daß Las Vegas qfa (qfa ohne Fehler, die aber mit Wahrscheinlichkeit ϵ „aufgeben“ dürfen) mindestens $D^{1-\epsilon}$ Zustände haben, wenn D die minimale dfa Größe für die erkannte Sprache ist. Diese unteren Schranken gelten auch für ein verallgemeinertes Modell von Quanten Automaten, das mindestens so mächtig wie probabilistische Automaten und wie Quanten Automaten ist.
Als eine zweite Anwendung betrachten wir die Nečiporuk Methode für Formellänge. Es wurde in [RV99] gezeigt, daß die (konventionelle) Nečiporuk Funktion bis auf einen logarithmischen Faktor auch eine untere Schranke für Quanten Formellänge ist. Das ist erstaunlich, weil diese Schranke für probabilistische Formeln nicht gilt, was wir in Theorem 3.7/3.8 gezeigt haben. Man kann daher folgern, daß das betrachtete Modell von Quanten Formeln [Y93] eingeschränkt ist. Wir schlagen ein verallgemeinertes Modell vor, in dem die Formel zusätzliche Zufallsvariablen lesen darf. So ist es möglich, klassische probabilistische Formeln zu simulieren. Für das verallgemeinerte Modell wird die untere Schranke mit der VC Nečiporuk Funktion als gültig nachgewiesen.
Auf dem Weg beweisen wir ein zu Fakt 2.32 analoges Theorem, welches die Spur Distanz zwischen Dichtematrizen mit der relativen von Neumann Entropie in Verbindung bringt. Weiterhin entwerfen wir ein programmierbares Quanten Gatter mit beliebig hoher Erfolgswahrscheinlichkeit (solche Gatter können nicht deterministisch sein, siehe [NC97]).
Die meisten Resultate dieses Kapitels sind bereits in [Kl00a] und [Kl00b] veröffentlicht.
Wir beginnen das Kapitel mit einer kurzen Einführung in die Quantenmechanik und die Quanten Informationstheorie.

5.2 Grundlagen aus der Quantenmechanik

Die Quantenmechanik ist die beste heute verfügbare Theorie für die Beschreibung physikalischer Ereignisse im mikroskopischen Bereich. Die Theorie ist in Begriffen von Zuständen und Transformationen von Zuständen formuliert. Der heute übliche mathematische Formalismus der Theorie wurde von John von Neumann eingeführt, siehe auch [vN32]. Wir konzentrieren uns

auf die mathematischen Definitionen und lassen alle physikalischen Erläuterungen weg. Wir verweisen auf [Pr98] und [Gr99] für sehr gute Einführungen in das Gebiet.

5.2.1 Quantenzustände und ihre Evolution

Pure Zustände und Hilberträume

Wir beginnen mit den Grundobjekten der Theorie, Zuständen und ihrer Evolution.
Für eine komplexe Zahl a sei a^* die konjugiert komplexe Zahl. Für eine Matrix M mit komplexen Einträgen sei $M^\dagger$ die Matrix, die sich aus M durch Transposition und Konjugation aller Einträge ergibt.
Zustände werden durch Vektoren in einem Hilbertraum formalisiert. Im folgenden wird die Bra-Ket Notation von Dirac benutzt (siehe [Di47]).

Definition 5.1 *Ein Hilbertraum H ist ein komplexer Vektorraum mit einem inneren Produkt, der vollständig unter der vom inneren Produkt induzierten Norm ist. Vektoren von H werden mit $|\psi\rangle$ bezeichnet, innere Produkte mit $\langle\psi|\phi\rangle$, und die Norm mit $|| \, |\psi\rangle \, || = \sqrt{\langle\psi|\psi\rangle}$.*
(Pure) Zustände sind Vektoren der Norm 1 im Hilbertraum.
Das äußere Produkt $|\psi\rangle\langle\phi|$ zweier Vektoren ist das übliche Produkt von Spaltenvektor und Zeilenvektor. Dabei ist $\langle\phi|$ als ein Zeilenvektor aufzufassen, der sich aus $|\phi\rangle$ durch Transposition und Konjugation der einzelnen Einträge ergibt.
$\langle\psi|A|\phi\rangle$ ist die Notation für $\langle\psi|A\phi\rangle$.

Wir benutzen ausschließlich Räume $\mathbb{C}^{2^n}$ und zeichnen eine Menge von Vektoren $\{|x\rangle | x \in \{0,1\}^n\}$ als eine orthonormale Basis aus, oder $\mathbb{C}^k$ mit $\{|i\rangle | i \in \{0, \ldots, k-1\}\}$ als orthonormaler Basis.
Alle puren Zustände sind Linearkombinationen von Basisvektoren mit Norm 1 und werden auch als Superpositionen bezeichnet. Die Koeffizienten einer solchen Linearkombination heißen auch Amplituden.

Definition 5.2 *Der Zustand eines Qubits ist ein Einheitsvektor in einem 2-dimensionalen Hilbertraum, d.h. eine Linearkombination $\alpha|0\rangle + \beta|1\rangle$ mit $|\alpha|^2 + |\beta|^2 = 1$.*
Entsprechend bezeichnet ein 2^n dimensionaler Hilbertraum ein n Qubit Register mit Zuständen $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$, so daß $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$.

Definition 5.3 *Das Tensorprodukt zweier Vektoren $(x_1, \ldots, x_n), (y_1, \ldots, y_m)$ ist $x \otimes y = (x_1y_1, \ldots, x_1y_m, \ldots, x_ny_m)$.*
Das Tensorprodukt zweier Matrizen

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

und B ist

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{n1}B & \cdots & a_{nn}B \end{pmatrix}.$$

Sind $B_1, \ldots, B_k$ Basen von Hilberträumen $H_1, \ldots, H_k$, dann wird der Raum $H_1 \otimes \cdots \otimes H_k$ von den Vektoren in $\{|x_1\rangle \otimes \cdots \otimes |x_k\rangle| \ |x_i\rangle \in B_i\}$ aufgespannt.

Im Raum $\mathbb{C}^4$ betrachten wir außer der Standard Basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ auch die sogenannte Bell Basis mit

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \ |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle),$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \ |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

Evolution von Quantenzuständen

Ein quantenmechanisches System evolviert im Laufe der Zeit, und ändert demgemäß seinen Zustand. Die Evolution wird durch eine unitäre Transformation beschrieben. Der Einfachheit halber nehmen wir an, daß die Zeit in diskreten Schritten abläuft. Dann ist der Zustand $|\phi(t+1)\rangle$ zur Zeit $t+1$ gegeben durch $U_t|\phi(t)\rangle$. Der Operator U_t ist dabei unitär und wird durch eine unitäre Matrix beschrieben.
Wir führen nun einige Beispiele von unitären Matrizen ein, die später benötigt werden. Ein grundlegendes Beispiel ist die Hadamard Transformation. Sei

$$H_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Dann ist $H_n = \underbrace{H_2 \otimes \cdots \otimes H_2}_{n}$, das $n$fache Tensorprodukt der 2 dimensionalen Hadamard Transformation. H_n operiert auf n Qubits. Man bemerke, daß

$$H_n|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y}|y\rangle,$$

wobei das Produkt $x \cdot y = \bigoplus_{i=1}^n x_i y_i$ verwendet wird.
Die Hadamard Transformation angewendet auf den Zustand $|0\rangle$ wird auch als ein fairer Münzwurf bezeichnet, da eine nachfolgende Messung in der Standardbasis jeden Basiszustand mit gleicher Wahrscheinlichkeit ergibt. Diese Eigenschaft wird im nächsten Abschnitt noch genauer betrachtet.
Die Pauli Matrizen sind

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Diese Transformationen sind für ein Qubit dimensioniert.
Die XOR Operation (exklusiv ODER) auf 2 Qubits, definiert durch $XOR : |x, y\rangle \to |x, x \oplus y\rangle$ auf Booleschen Werten x, y hat die Matrix

$$XOR = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Eine Anwendung der Hadamard Transformation auf ein Quantenregister mit n Qubits ergibt einen Zustand, der eine Superposition von exponentiell vielen Basiszuständen ist. Es ist daher verlockend, exponentiell viele Funktionswerte „parallel“ zu berechnen in der Hoffnung, nützliche Information daraus abzuleiten. Zum Beispiel könnte man einen nichtdeterministischen Schaltkreis parallel auf allen Belegungen der nichtdeterministischen Ratebits testen, und dann versuchen, Information über die Existenz eines erfüllenden Rateworts zu erhalten.
Man verwendet dazu die folgende Transformation. Sei $f : \{0,1\}^n \to \{0,1\}$ eine Boolesche Funktion. Dann ist die Abbildung $X_f : x \mapsto x, f(x)$ bijektiv und es gibt eine unitäre Transformation U_f, welche $|x, b\rangle$ auf $|x, b \oplus f(x)\rangle$ für alle Booleschen x und alle $b \in \{0,1\}$ abbildet. Wendet man U_f auf $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, 0\rangle$ an, so erhält man

$$\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, f(x)\rangle.$$

Ein weiterer Weg, alle Funktionswerte „parallel“ zu berechnen, ist durch den Operator $V_f|x\rangle = (-1)^{f(x)}|x\rangle$, der den Funktionswert durch einen Vorzeichenwechsel anzeigt. Durch folgendes Argument kann V_f von U_f abgeleitet werden.

$$\begin{aligned} U_f(|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)) &= \frac{1}{\sqrt{2}}(|x, f(x)\rangle - |x, 1 \oplus f(x)\rangle) \\ &= (-1)^{f(x)}|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned}$$

Observablen und Messungen

Abgesehen von der unitären Evolution gibt es mit der Messung einen weiteren Typ von Operationen. Wir beschreiben hier nur die einfachen von Neumann Projektionsmessungen, und lassen die komplizierteren „positive operator valued measurements (POVM)“ aus. Die letzteren können durch von Neumann Messungen auf Quantensystemen, die um einige Hilfsqubits erweitert worden sind, simuliert werden und werden implizit bei der Diskussion von Superoperatoren mitbehandelt.

Eine Messung ist die einzige Möglichkeit, klassische Information aus einem Quantenzustand zu erhalten. Um eine Messung anzuwenden, muß zunächst eine Observable definiert werden.

Definition 5.4 *Eine Observable für einen Hilbertraum ist durch einen selbstadjungierten Operator A gegeben. Der Operator hat eine spektrale Repräsentation $A = \sum_{i=1}^{k} \lambda_i P_i$, wobei die λ_i die paarweise verschiedenen Eigenwerte von A sind und die P_i die Projektionen auf die jeweils von den Eigenvektoren zu den λ_i aufgespannten Eigenräume. Es ist möglich, zu jedem der Eigenräume (und damit für den gesamten Hilbertraum) eine orthonormale Basis aus Eigenvektoren von A zu bestimmen.*
Eine Messung der Observablen A ergibt für einen Zustand $|\phi\rangle$ ein λ_i, also einen der Eigenwerte von A. Die Wahrscheinlichkeit von λ_i ist

$$\Pr(\lambda_i) = ||P_i|\psi\rangle||^2.$$

Wenn λ_i das Resultat der Messung ist, so verändert sich der gemessene Quantenzustand zu

$$\frac{P_i|\psi\rangle}{\langle\psi|P_i|\psi\rangle^{1/2}}.$$

Eine Messung verändert also anders als im klassischen Fall den gemessenen Zustand. Verschiedene Observablen müssen sich nicht kommutativ verhalten, d.h. es gibt „Eigenschaften“ von Quantensystemen, welche nicht gemessen werden können, ohne andere „Eigenschaften“ zu verändern.
Für eine gegebene orthonormale Basis eines Hilbertraums sagen wir, daß in dieser Basis gemessen wird, wenn eine Observable gemessen wird, welche aus Projektionen auf die einzelnen Basisvektoren besteht.
Weiterhin erlauben wir im allgemeinen auch, daß statt der Eigenwerte λ_i beliebige Boolesche Worte Ergebnis einer Messung sein können, wenn es eine einfache bijektive Abbildung von der Menge der Eigenwerte auf eine Menge solch alternativer Ergebnisse gibt.

Gemischte Zustände und Dichtematrizen

Eine Messung ist ein Prozeß, bei dem das Ergebnis probabilistisch bestimmt wird. Aber Probabilismus findet sich in der Quantenmechanik noch auf einem anderen, fundamentaleren Weg. Wenn ein Hilbertraum bipartit ist, d.h. das Tensorprodukt zweier Hilberträume ist, so kann man die Situation betrachten, wenn nur einer der Teilräume zugänglich ist. Sei es, daß der andere Raum ganz allgemein für die Umwelt eines Quantensystems steht, sei es, daß der Raum zu den Qubits eines zweiten Spielers korrespondiert, die uns nicht zugänglich sind. Als ein Beispiel betrachten wir den Zustand $|\Phi^+\rangle = 1/\sqrt{2}|11\rangle + 1/\sqrt{2}|00\rangle$, wobei das erste Qubit Spieler A gehöre, und das zweite Qubit Spieler B. Für beide Spieler ist der Zustand ihres Qubits

kein purer Zustand, sondern äquivalent zu einem probabilistischen Ensemble, bei dem Zustände $|1\rangle$ oder $|0\rangle$ mit Wahrscheinlichkeit $1/2$ auftreten, obwohl bei beiden Spielern immer derselbe Wert auftritt. Dieses Verhalten kann verglichen werden mit einem Paar von Würfeln, die beide für sich ganz zufällige Resultate liefern, aber die zusätzliche „globale“ Eigenschaft haben, daß beide stets den gleichen Wert zeigen. Besitzt man nur einen der Würfel, so ist es unmöglich, diesen von einem normalen Würfel zu unterscheiden, besitzt man jedoch beide, so kann man sie von einem gewöhnlichen Paar relativ problemlos unterscheiden. Dieses Phänomen heißt „Verschränkung“ bzw. entanglement im Englischen. Betrachtet man nur einen Teil eines Quantensystems, so erhält man keine vollständige deterministische Beschreibung des Zustands dieses Teils. Dem wird durch die Verwendung von gemischten Zuständen und deren Darstellung mit Dichtematrizen Rechnung getragen.

Definition 5.5 *Ein Ensemble von puren Zuständen wird durch eine Menge* $\{(p_i, |\phi_i\rangle)|1 \leq i \leq k\}$ *beschrieben. Dabei sind die* p_i *die Wahrscheinlichkeiten der puren Zustände* $|\phi_i\rangle$ *und* k *ist eine natürliche Zahl. Ein solches Ensemble wird ein gemischter Zustand genannt.*
Die Dichtematrix eines puren Zustands $|\phi\rangle$ *ist die Matrix* $|\phi\rangle\langle\phi|$*, die Dichtematrix eines gemischten Zustands* $\{(p_i, |\phi_i\rangle)|1 \leq i \leq k\}$ *ist*

$$\sum_{i=1}^{k} p_i |\phi_i\rangle\langle\phi_i|.$$

Man beachte, daß die obige Repräsentation eines gemischten Zustands von der gewählten Basis abhängt. Das Konzept der Dichtematrix kann auch zu einem Dichteoperator verallgemeinert werden, wir nehmen aber an, daß die Basis immer aus dem Kontext klar ist.
Eine Dichtematrix ist immer Hermitesch, positiv semidefinit, und hat die Spur 1. Daher hat die Dichtematrix nichtnegative Eigenwerte, die aufsummiert 1 ergeben. Alle möglichen Messungen eines gemischten Zustands sind durch die Dichtematrix bestimmt. Wenn also zwei gemischte Zustände dieselbe Dichtematrix haben, so können sie nicht auseinandergehalten werden. In der Quantenmechanik spielen Dichtematrizen dieselbe Rolle wie Dichtefunktionen bzw. Verteilungen von Zufallsvariablen in der klassischen Wahrscheinlichkeitstheorie. Ist ein gemischter Zustand eine Wahrscheinlichkeitsverteilung auf den paarweise orthogonalen Basisvektoren eines Hilbertraums, so enthält die Dichtematrix diese Verteilung auf ihrer Diagonalen.
Eine wichtige Anwendung von Dichtematrizen ist die Beschreibung von Teilsystemen eines Quantensystems. Ein purer Zustand in einem Hilbertraum $H = H_A \otimes H_B$ kann im allgemeinen nicht in das Tensorprodukt zweier purer Zustände der Teilsysteme zerlegt werden.

Definition 5.6 *Ein gemischter Zustand* $\{(p_i, |\phi_i\rangle)|1 \leq i \leq k\}$ *in einem Hilbertraum* $H_1 \otimes H_2$ *heißt separabel, wenn er dieselbe Dichtematrix hat wie*

ein gemischter Zustand $\{(q_i, |\psi_i^1\rangle \otimes |\psi_i^2\rangle)|i = 1, \ldots, k'\}$ *für pure Zustände* $|\psi_i^1\rangle$ *aus* H_1 *und* $|\psi_i^2\rangle$ *aus* H_2 *mit* $\sum_i q_i = 1$ *und* $q_i \geq 0$. *Ansonsten heißt der gemischte Zustand verschränkt.*

Separable Zustände sind klassische Wahrscheinlichkeitsverteilungen auf puren Zuständen, welche aus unabhängigen Komponenten in beiden Teilsystemen bestehen. Alle anderen Zustände sind verschränkt. Alle Korrelationen zwischen den Teilen eines separablen Zustands sind also klassischer Natur. Ist ein separabler Zustand pur, so ist er das Tensorprodukt aus zwei puren Zuständen. Wir betrachten z.B. den Zustand $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ in $\mathbb{C}^2 \otimes \mathbb{C}^2$. Der Zustand ist verschränkt und wird oft als ein EPR-Paar von Qubits bezeichnet. Diese Begriffsbildung bezieht sich auf Einstein, Podolsky und Rosen, die zuerst solche Zustände betrachtet haben [EPR35].
Sei ρ_{AB} eine Dichtematrix über $\mathbb{C}^m \otimes \mathbb{C}^n$. Das Reduzieren der Dichtematrix um den Raum $\mathbb{C}^n$ bedeutet die Ersetzung der Dichtematrix durch $\rho_A = Spur_B \rho_{AB}$, wobei man $Spur_B \rho_{AB}(i,j) = \sum_{k=1}^{2^n} \rho_{AB}(ik, jk)$ definiert. Das bedeutet intuitiv eine Mittelwertbildung über das zweite Quantensystem.
Jede Dichtematrix kann in einer Basis diagonalisiert werden. Dann kann eine Funktion auf die Einträge der Diagonalen angewendet werden, und die Basis rücktransformiert werden. In diesem Sinne wenden wir Funktionen auf Dichtematrizen an, wie $\log \rho$ und $\sqrt{\rho}$.
Wir betrachten wieder den Zustand $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Seine Dichtematrix ist

$$\begin{pmatrix} 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \end{pmatrix}.$$

Reduziert man auf 1 Qubit, so erhält die Dichtematrix $\begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$. Ist nur ein Qubit zugänglich, so kann dessen Zustand nicht von einem Ensemble unterschieden werden, in dem $|0\rangle$ und $|1\rangle$ mit Wahrscheinlichkeit $1/2$ gewählt werden. Der Zustand des einzelnen Qubits ist hier kein purer Zustand in $\mathbb{C}^2$. Mißt man hingegen den gesamten Zustand in der Bell Basis, so ist es einfach, den Zustand $|\Phi^+\rangle$ von dem Tensor Produkt zweier Zufallsbits zu unterscheiden. Es gibt also einen Unterschied zwischen einem verschränkten Zustand und zwei separaten zufälligen Zuständen, aber dieser Unterschied ist nur feststellbar, wenn man den gesamten Zustand betrachtet.

Superoperatoren

Transformationen auf Dichtematrizen müssen linear sein, und Dichtematrizen auf Dichtematrizen abbilden. Solche Transformationen heißen Superoperatoren. Aber nicht alle Superoperatoren sind physikalisch erlaubt. Beispiele

erlaubter Superoperatoren sind unitäre Transformationen auf den zugrundeliegenden Zuständen, Messungen von Observablen, das Reduzieren um Teilsysteme, und das Zufügen einiger „leerer“ Qubits (als Tensorprodukt mit dem Ausgangszustand).

Definition 5.7 *Ein Superoperator T heißt positiv, wenn er positiv semidefinite Hermitesche Matrizen auf positiv semidefinite Hermitesche Matrizen abbildet. Ein Superoperator heißt spurbewahrend, wenn er Matrizen mit Spur 1 auf Matrizen mit Spur 1 abbildet.*
Ein Superoperator T heißt vollständig positiv, wenn jeder Superoperator $T \otimes I_F$ positiv ist, wobei I_F der identisch abbildende Superoperator auf einer endlich dimensionalen Erweiterung F des zugrundeliegenen Hilbertraums ist.
Ein Superoperator ist genau dann physikalisch erlaubt, wenn er vollständig positiv und spurbewahrend ist.

Sei U eine unitäre Transformation. Durch ihre Anwendung auf die Zustände im zur Dichtematrix gehörigen Ensemble wird die Dichtematrix ρ zu $U\rho U^\dagger$ transformiert. Das ist ein Beispiel eines erlaubten Superoperators. Weitere Beispiele sind das Reduzieren um ein Teilsystem, das Messen, und das Zufügen einiger „leerer“ Qubits. Ein Beispiel eines positiven spurbewahrenden Superoperators, der nicht vollständig positiv ist, ist die Transposition einer Dichtematrix.
Das folgende wichtige Theorem (auch Kraussches Repräsentationstheorem genannt) zeigt, daß jeder erlaubte Superoperator durch das Zufügen einiger Qubits, die Anwendung einer unitären Transformation, und das Reduzieren um ein Teilsystem simuliert werden kann (siehe [Pr98]).

Fakt 5.1 *Folgende Aussagen sind äquivalent:*

1. *Ein Superoperator T, der Dichtematrizen von H_1 auf Dichtematrizen von H_2 abbildet, ist spurbewahrend und vollständig positiv.*

2. *Es gibt einen Hilbertraum H_3 mit $dim(H_3) \leq dim(H_1)$ und eine unitäre Abbildung U, so daß für alle Dichtematrizen ρ über H_1 gilt:*

$$T\rho = Spur_{H_1 \otimes H_3}[U(\rho \otimes |0_{H_3 \otimes H_2}\rangle\langle 0_{H_3 \otimes H_2}|)U^\dagger].$$

Superdense Coding

Um diesen Abschnitt abzuschließen, beschreiben wir einen Prozeß namens *superdense coding*, der von Bennett und Wiesner [BW92] erfunden wurde. A und B besitzen je eines der Qubits eines EPR Paars. Weiterhin sind A und B durch einen Quantenkommunikationskanal verbunden, das heißt A kann B ein Qubit senden. Das Ziel der beiden ist es, zwei klassische Bits von A zu B zu kommunizieren, indem nur ein Qubit von A zu B gesendet wird,

wobei allerdings das EPR Paar „verbraucht“ wird. Das Protokoll verläuft wie folgt:
A und B besitzen je eines der Qubits von $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. A kennt Bits b_1, b_2 und möchte diese B mitteilen. A wendet abhängig von den zwei Bits eine der unitären Transformationen aus der folgenden Tabelle auf ihr Qubit an und sendet B das berechnete Qubit. B wendet die XOR Transformation an. Zum Schluß mißt B seine Qubits. Das von A empfangene Qubit wird in der Basis $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ und $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ gemessen, das andere Qubit in der Basis $|0\rangle$ und $|1\rangle$. Das gibt B zwei Bits, aus denen der Wert von b_1, b_2 rekonstruiert werden kann.

A Bits	A Transformation	Neuer Zustand der Qubits
00	I	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$
01	σ_x	$\frac{1}{\sqrt{2}}(\|01\rangle + \|10\rangle)$
10	$i \cdot \sigma_y$	$\frac{1}{\sqrt{2}}(\|01\rangle - \|10\rangle)$
11	σ_z	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$

A Bits	Nach XOR	Nach Messung	B Ausgabe
00	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)\|0\rangle$	00	00
01	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)\|1\rangle$	01	01
10	$\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)\|1\rangle$	11	10
11	$\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)\|0\rangle$	10	11

Es ist bemerkenswert, daß die von A gesendete „Nachricht“ einfach ein Qubit mit der Dichtematrix $\frac{1}{2}I$ ist (I sei die Einheitsmatrix), also für jemanden ohne das zweite Qubit keinerlei Information enthält.

5.2.2 Quanten Informationstheorie

In diesem Abschnitt beschreiben wir Verallgemeinerungen der Begriffe der klassischen Informationstheorie auf die Quantenwelt und deren Eigenschaften.

Definition 5.8 *Die von Neumann Entropie einer Dichtematrix ρ_X ist definiert durch $S(X) = S(\rho_X) = -Spur(\rho_X \log \rho_X)$.*
Die bedingte von Neumann Entropie $S(X|Y)$ eines bipartiten Systems mit Dichtematrix ρ_{XY} ist definiert als $S(XY) - S(Y)$, wobei der Zustand ρ_Y des Y Teilsystems das Resultat des Reduzierens des gemeinsamen Zustands ρ_{XY} um X ist (siehe [CA96] für eine andere, aber äquivalente Definition).
Die von Neumann Information zwischen zwei Teilen eines bipartiten Systems im Zustand ρ_{XY} ist $S(X : Y) = S(X) + S(Y) - S(XY)$ (ρ_X und ρ_Y ergeben sich wieder durch Reduzieren um das jeweils andere Teilsystem).

Die bedingte von Neumann Information bei einem dreiteiligen System im Zustand ρ_{XYZ} ist $S(X:Y|Z) = S(XZ) + S(YZ) - S(Z) - S(XYZ)$.
Die relative von Neumann Entropie zwischen zwei Dichtematrizen ρ_X, ρ_Y der gleichen Dimension ist $R(X|Y) = R(\rho_X|\rho_Y) = Spur(\rho_X[\log \rho_X - \log \rho_Y])$. Dieser Wert kann unendlich sein.
Sei $\mathcal{E} = \{(p_i, \rho_i)|i = 1, \ldots, k\}$ ein Ensemble von Dichtematrizen. Die Holevo Information des Ensembles ist $\chi(\mathcal{E}) = S(\sum_i p_i \rho_i) - \sum_i p_i S(\rho_i)$.

Da die von Neumann Entropie nur von den Eigenwerten einer Dichtematrix abhängt, ist sie invariant unter unitären Transformationen. Hat ein Hilbertraum die Dimension d so ist die von Neumann Entropie einer Dichtematrix auf dem Raum durch $\log d$ beschränkt. Ein fundamentales Ergebnis ist die sogenannte Holevo Schranke [H73], die angibt, wieviel klassische Information in einem Quantenzustand kodiert werden kann.

Fakt 5.2 *Sei X eine klassische Zufallsvariable mit $\Pr(X = x) = p_x$. Angenommen, für jedes x werde ein Quantenzustand mit Dichtematrix ρ_x präpariert, d.h. es gibt ein Ensemble $\mathcal{E} = \{(p_x, \rho_x)|x = 0, \ldots, k\}$. Sei $\rho_{XZ} = \sum_{x=0}^{k} p_x |x\rangle\langle x| \otimes \rho_x$. Sei Y eine (klassische) Zufallsvariable, die das Ergebnis einer Messung des Quantenzustands mit Dichtematrix $\rho_Z = \sum_x p_x \rho_x$ enthält. Dann ist*

$$H(X:Y) \leq \chi(\mathcal{E}) = S(X:Z).$$

In Abschnitt 5.4.3. benötigen wir das folgende Lemma (eine „Las Vegas Version“ eines Lemmas aus [Na99]).

Lemma 5.1 *Sei $\mathcal{E} = \{(p_x, \sigma_x)|x = 0, \ldots, k\}$ ein Ensemble von Dichtematrizen und sei $\sigma = \sum_x p_x \sigma_x$ die Dichtematrix des durch das Ensemble beschriebenen gemischten Zustands. Gibt es eine Observable mit möglichen Resultaten x und ?, so daß für alle x bei einer Messung der Observablen auf σ_x das Ergebnis x Wahrscheinlichkeit mindestens $1-\epsilon$ hat und das Ergebnis ? Wahrscheinlichkeit höchstens ϵ hat, dann gilt*

$$S(\sigma) \geq \sum_x p_x S(\sigma_x) + H(X)(1-\epsilon), \text{ d.h. } \chi(\mathcal{E}) \geq (1-\epsilon)H(X).$$

Beweis: Es werden klassische Zustände x einer Zufallsvariablen X als Quantenzustände σ_x kodiert, wobei x und σ_x Wahrscheinlichkeit p_x haben. Die Dichtematrix des gemischten Zustandes ist σ und hat eine von Neumann Entropie $S(\sigma)$. σ entspricht der „Kodierung“ eines zufälligen x.
Nach Holevos Theorem ist die Information, welche man über X durch eine Messung von σ mit Resultat Y erfahren kann, durch $H(X:Y) \leq S(\sigma) - \sum_x p_x S(\sigma_x)$ beschränkt. Aber es gibt eine Messung wie in der Behauptung beschrieben, und nach Lemma 2.3 ist $H(X:Y) \geq (1-\epsilon)H(X)$. Somit folgt das Lemma. □

Nicht alle Beziehungen, die in der klassischen Informationstheorie gelten, sind auch für die von Neumann Entropie wahr. Die folgende Tatsache beschreibt die sogenannte Araki-Lieb Ungleichung und eine ihrer Konsequenzen. Hier wird ein wesentlicher Unterschied zur klassischen Entropie sichtbar (siehe [Pr98]).

Fakt 5.3 $S(XY) \geq |S(X) - S(Y)|$.
$S(X:Y) \leq 2S(X)$.

Die in der klassischen Informationstheorie gültigen Ungleichungen $H(XY) \geq H(X)$ und $H(X:Y) \leq H(X)$ sind für die von Neumann Entropie nicht korrekt, was sich im Superdense Coding ausnutzen läßt. Der Grund für dieses Verhalten ist die Verschränkung.

Lemma 5.2 *Ist σ_{XY} separabel, dann gilt $S(XY) \geq S(X)$,*
$S(X:Y) \leq S(X)$.

Beweis: In [CA96] wird gezeigt, daß für einen separablen bipartiten Zustand $S(X|Y) \geq 0$ gilt. Daher ist $S(XY) = S(X) + S(Y|X) \geq S(X)$ und $S(X:Y) = S(X) - S(X|Y) \leq S(X)$. □
Die nächste Eigenschaft heißt auch Lindblad Uhlmann-Monotonie der von Neumann Entropie (siehe [Pr98]).

Fakt 5.4 *Sei T ein vollständig positiver, spurbewahrender Superoperator. Dann gilt für alle Dichtematrizen ρ_X, ρ_Y:*

$$R(X|Y) = R(\rho_X|\rho_Y) \geq R(T(\rho_X)|T(\rho_Y)).$$

Die folgende Norm auf linearen Operatoren wird in [AKN98] betrachtet.

Definition 5.9 *Sei ρ die Matrix eines linearen Operators. Dann ist die Spur Norm von ρ, geschrieben $||\rho||_1$, die Summe der Absolutwerte der Elemente der Multimenge der Eigenwerte von ρ. Es ist $||\rho||_1 = Spur(\sqrt{\rho^\dagger \cdot \rho})$.*

Die Distanz $||\rho - \sigma||_1$ ist reell für Hermitesche Matrizen ρ, σ. Die Spur Norm hat eine enge Beziehung zu der maximalen meßbaren Distanz zwischen zwei Zuständen, wie in [AKN98] bewiesen.

Fakt 5.5 *Für eine Observable O und eine Dichtematrix ρ bezeichne p_ρ^O die durch O und ρ induzierte Verteilung auf den Meßresultaten. Dann gilt*

$$||\rho - \sigma||_1 = \max_O \{||p_\rho^O - p_\sigma^O||\}.$$

Wenn also zwei Dichtematrizen in der Spur Distanz nah zueinander sind, so kann keine Messung sie mit hoher Wahrscheinlichkeit auseinanderhalten. Haben zwei Zustände die gleiche Dichtematrix, so sind sie prinzipiell ununterscheidbar. Das nächste, ähnliche Lemma folgt direkt aus dem Krausschen Repräsentationstheorem (Fakt 5.1), welches ermöglicht, physikalisch erlaubte Superoperatoren auf Dichtematrizen durch unitäre Transformationen auf Dichtematrizen über einem erweiterten Hilbertraum und Reduktion um einen Teilraum zu simulieren.

Lemma 5.3 *Für jede Hermitesche Matrix ρ und jeden spurbewahrenden, vollständig positiven Superoperator F gilt*

$$||\rho||_1 \geq ||F(\rho)||_1.$$

Ist die übliche Distanz in einem Hilbertraum zwischen zwei puren Zuständen klein, so ist die Spur Distanz zwischen ihren Dichtematrizen klein[AKN98].

Fakt 5.6 *Für alle puren Zustände $|\phi\rangle, |\psi\rangle$:*

$$||\,|\phi\rangle\langle\phi| - |\psi\rangle\langle\psi|\,||_1 \leq 2||\,|\phi\rangle - |\psi\rangle||.$$

Das folgende Theorem erlaubt es, die Spur Distanz über die relative Entropie zu beschränken. Eine klassische Variante des Theorems findet sich in [Bl87] und wurde z.B. in [R98] und in Abschnitt 3.2.2 benutzt.

Theorem 5.1 *Für Dichtematrizen ρ, σ der gleichen Größe gilt:*

$$R(\rho|\sigma) \geq \frac{1}{2\ln 2}||\rho - \sigma||_1^2.$$

Beweis: Weil sowohl die Spur Norm als auch die relative von Neumann Entropie invariant unter unitären Transformationen auf den zugrundeliegenden Zuständen sind, nehmen wir an, daß die Basis der Dichtematrizen die Matrix $\rho - \sigma$ diagonalisiert. Im allgemeinen sind allerdings weder ρ noch σ diagonal in der so gewählten Basis. Sei S die Multimenge der nichtnegativen Eigenwerte von $\rho - \sigma$ und R die Multimenge der negativen Eigenwerte. Alle Eigenwerte sind reell, weil $\rho - \sigma$ Hermitesch ist. Hat nun der Raum H_S, der von den Eigenvektoren zu S aufgespannt wird, Dimensions k, und der Raum H_R, der von den Eigenvektoren zu R aufgespannt wird, Dimensions $n - k$, so vergrößern wir den zugrundeliegenden Hilbertraum, bis beide Räume dieselbe Dimension haben. Die Dichtematrizen haben an den entsprechenden Stellen Nullen als Einträge. Nun betrachten wir die Dichtematrizen als definiert über dem Produktraum $H_2 \otimes H_{n'}$, wobei H_2 den Raum H_S oder H_R „angibt“ und $n' = \max\{k, n-k\}$.
Wir reduzieren nun $\rho, \sigma, \rho - \sigma$ um den Raum $H_{n'}$ und erhalten 2×2 Matrizen $\widetilde{\rho}, \widetilde{\sigma}, \widetilde{\rho - \sigma}$. Die Matrix $\widetilde{\rho - \sigma}$ ist diagonal und enthält die Summe aller

nichtnegativen Eigenwerte und die Summe aller negative Eigenwerte auf der Diagonalen. Weiterhin ist $\widetilde{\rho - \sigma} = \widetilde{\rho} - \widetilde{\sigma}$.
Wegen der Lindblad-Uhlmann-Monotonie (siehe Fakt 5.4) der relative Entropie gilt $R(\rho|\sigma) \geq R(\widetilde{\rho}|\widetilde{\sigma})$. Wie beschränken letzteres durch

$$1/(2\ln 2)||\widetilde{\rho} - \widetilde{\sigma}||_1^2 = 1/(2\ln 2)||\widetilde{\rho - \sigma}||_1^2,$$

und schließen dann auf das Theorem, weil die Spur Norm von $\widetilde{\rho - \sigma}$ die Summe der Absolutwerte der Eigenwerte ist, welche nach Konstruktionn die Summe der Absolutwerte der Eigenwerte von $\rho - \sigma$ ist, d.h. $||\widetilde{\rho} - \widetilde{\sigma}||_1 = ||\rho - \sigma||_1$.
Es reicht also, das Theorem für 2×2 Dichtematrizen zu zeigen. Angenommen, die Basis sei so gewählt, daß σ diagonal ist. Dann ist

$$\rho = \begin{pmatrix} a & b \\ b^* & 1-a \end{pmatrix} \text{ and } \sigma = \begin{pmatrix} c & 0 \\ 0 & 1-c \end{pmatrix}.$$

Die relative von Neumann Entropie ist $R(\rho|\sigma) = -S(\rho) - trace[\rho \log \sigma]$. Der zweite Term ist $-a\log c - (1-a)\log(1-c)$.
Der erste Term ist die mit -1 multiplizierte Entropie der Verteilung, die durch die Eigenwerte von ρ gegeben ist. Wir berechnen die Eigenwerte.
Die Eigenwerte von ρ sind die Nullstellen des charakteristischen Polynoms $t^2 - t + a(1-a) - bb^*$. Diese sind $1/2 \pm \sqrt{1/4 - a(1-a) + bb^*}$. Sei $x = 1/2 + \sqrt{1/4 - a(1-a) + bb^*}$. Dann ist $S(\rho) = H(x)$.
Die quadrierte Norm von $\rho - \sigma$ ist die quadrierte Summe der Absolutwerte der Eigenwerte von $\rho - \sigma$. Diese Matrix hat als charakteristisches Polynom $t^2 - (-a(1-a) + a(1-c) + (1-a)c - c(1-c) + bb^*)$. Daher sind die Eigenwerte $\pm\sqrt{-a(1-a) + a(1-c) + (1-a)c - c(1-c) + bb^*}$. Die quadrierte Norm ist die quadrierte Summe der Absolutwerte der Eigenwerte, und daher

$$4(a^2 + c^2 - 2ac + bb^*).$$

Zuerst betrachten wir den Fall $a = c$. Um diesen Fall zu beweisen haben wir zu zeigen, daß $H(a) - H(x) \geq 2\log(e)bb^* = 2\log(e)[(x-1/2)^2 - 1/4 + a(1-a)] = 2\log(e)[(x^2 - x) - (a^2 - a)]$.
Betrachtet man die Funktion $H(y)/\log(e) + 2y^2 - 2y$, so ist leicht zu sehen, daß sie nichtnegativ und monoton fallen ist für alle y zwischen $1/2$ und 1. Daher gilt die Ungleichung, wenn $1/2 \leq a$ und $a \leq x$. Die erste Bedingung gilt o.B.d.A., die zweite Bedingung folgt, da $x \geq 1/2$ der größte Eigenwert ist, und $a \geq 1/2$ ein Element der Diagonale.
Nun betrachten wir den Fall $c \geq a$. Gilt $c < a$, so können wir stattdessen dasselbe Argument für $1-c$ und $1-a$ durchführen. Wir wollen zeigen, daß

$$f(c) = R(\rho|\sigma)/\log(e) - \frac{1}{2}||\rho - \sigma||_1^2 \geq 0.$$

Dies ist wahr für $a = c$, somit reicht es zu zeigen, daß eine Vergrößerung von c diese Differenz nicht verkleinern kann. Dies ist wahr, weil für die Ableitung nach c gilt:

$$\begin{aligned} f'(c) &= -a/c + (1-a)/(1-c) - 2(2c-2a) \\ &= \frac{(1-a)c - a(1-c)}{c(1-c)} - 4(c-a) \\ &\geq 4(c-a) - 4(c-a) \geq 0. \end{aligned}$$

Somit erhalten wir das Theorem für den 2×2 Fall und daher allgemein. □

5.3 Modelle von Quantenrechnern

5.3.1 Das Kommunikationsmodell

Nun definieren wir Quanten Kommunikationsprotokolle. Allgemeine Information über Quantenrechner findet man in [Gr99] und [Pr98].

Definition 5.10 *In einem zwei Spieler Quanten Protokoll erhalten die Spieler A und B jeweils eine private Menge von Qubits. Einige Qubits sind jeweils initialisiert zu einer Booleschen Eingabe, alle anderen Qubits sind im Zustand* $|0\rangle$.
In einer Kommunikationsrunde kann einer der Spieler verschiedene Aktionen ausführen. Der Spieler darf seine Qubits mit einer von seiner klassischen Eingabe abhängigen Observable messen. Weiterhin wendet der Spieler eine unitäre Transformation auf seine Qubits an. Formal bedeutet beides, daß das Tensorprodukt der Operation (auf den Qubits des Spielers) mit der identischen Abbildung (auf den Qubits des anderen Spielers) auf den Gesamtzustand angewendet wird. Dann werden einige Qubits zum anderen Spieler gesendet. Letztere Aktion verändert nicht den globalen Zustand, aber den Besitz einzelner Qubits. Die Wahl der Transformation und der zu sendenden Qubits darf von der Eingabe und von dem Ergebnis der Messung abhängen. Nach dem Ende des Protokolls wird der Zustand einiger Qubits mit einer bestimmten Observablen gemessen und das Resultat bestimmt. Die Komplexität eines Protokolls ist die maximale Anzahl von Qubits, die von dem Protokoll verschickt wird.
Bei einem exakten Quanten Protokoll muß das Resultat mit Wahrscheinlichkeit 1 korrekt sein. Die exakte Quanten Kommunikationskomplexität einer Funktion, $Q_E(f)$, *ist die minimale Komplexität eines exakten Quanten Protokolls für* f.
Bei einem Quanten Protokoll mit beschränktem Fehler muß die korrekte Ausgabe mit Wahrscheinlichkeit $1 - \epsilon$ *(für* $1/2 > \epsilon > 0$*) gegeben werden. Die Quanten Kommunikationskomplexität (mit beschränktem Fehler) einer Funktion* f *ist* $Q_\epsilon(f)$ *bzw.* $Q(f) = Q_{1/3}(f)$, *die minimale Komplexität eines Quanten Protokolls mit beschränktem Fehler für* f.

Quanten Las Vegas Protokolle sind ebenfalls von der Akzeptanz her wie ihr klassisches Pendant definiert, die Notationen sind $Q_0(f)$ und $Q_{0,\epsilon}(f)$.
In einem nichtdeterministischen Quanten Protokoll werden Eingaben genau dann akzeptiert, wenn das Protokoll mit positiver Wahrscheinlichkeit akzeptiert. $NQ(f)$ bezeichnet die nichtdeterministische Quanten Kommunikationskomplexität.
Beschränkungen der Rundenzahl werden wie zuvor definiert.
[CB97] und [CDNT97] haben ein anderes Modell der Quanten Kommunikation vorgeschlagen: A und B besitzen eine Menge von Qubits in einem beliebigen eingabeunabhängigen Zustand, d.h. die Qubits sind eventuell miteinander verschränkt. Dann wird wie oben definiert ein Kommunikationsprotokoll benutzt. Wir bezeichnen Komplexitätsmaße in diesem Modell durch den Superscript pub.

Man kann das Modell mit verschränkten Qubits simulieren, indem man zuerst eine eingabeunabhängige Kommunikation beliebiger Länge erlaubt, und dann ein normales Protokoll benutzt.
Es ist möglich, durch das Messen verteilter EPR Paare öffentlichen Zufall zu simulieren, wenn verschränkte Qubits vorhanden sind.
Die Technik des superdense coding von [BW92] erlaubt es im Modell mit Verschränkung, n Bits klassische Information mit $\lceil n/2 \rceil$ Qubits zu kommunizieren, siehe Abschnitt 5.2.1.

5.3.2 Quanten Schaltkreise

Abgesehen von Quanten Turingmaschinen sind Quanten Schaltkreise [D89] ein universelles Modell von Quantenberechnungen, siehe [Y93], und im allgemeinen besser als Turingmaschinen geeignet, um Quantenalgorithmen zu beschreiben. Ein allgemeineres Modell von Quanten Schaltkreisen, in dem auf gemischten Zuständen gearbeitet wird, ist in [AKN98] definiert. Wir beginnen mit dem grundlegenden Modell.

Definition 5.11 *Ein unitäres Quanten Gatter mit k Eingängen und k Ausgängen ist durch einen unitären Operator $U : \mathbb{C}^{2^k} \to \mathbb{C}^{2^k}$ spezifiziert, und wird durch eine unitäre Matrix repräsentiert.*
Ein Quanten Schaltkreis besteht aus Quantengattern mit jeweils höchstens $O(1)$ Eingängen sowie ausEingaben, die insgesamt zu einem azyklischen gerichteten Graphen verbunden sind, bei dem die Eingaben den Quellen zugeordnet sind. Die Quellen sind entweder mit einer Booleschen Konstante oder mit einer Eingabevariablen markiert. Ein ausgezeichnetes Gatter mit einem Ausgang ist die Ausgabe. Kanten entsprechen Qubits, auf denen gerechnet werden, es wird auf so vielen Qubits gerechnet, wie Quellen vorhanden sind. Die Größe eines Schaltkreises ist die Anzahl der Gatter, die Tiefe die Länge eines längsten Pfades von einer Eingabe oder einer Konstanten zur Ausgabe.

Ein Quanten Schaltkreis berechnet eine Boolesche Funktion folgendermaßen. Zu Beginn wird den Quellen des Schaltkreises je ein Boolescher Wert zugewiesen. Der Boolesche Wert 0 korrespondiert dann zu $|0\rangle$*, der Wert 1 zu* $|1\rangle$*. Dann werden die Gatter in einer beliebigen topologischen Reihenfolge auf die ihren Eingängen zugehörigen Qubits angewendet. Zum Schluß wird der Zustand des Augabequbits in der Basis* $\{|0\rangle, |1\rangle\}$ *gemessen. Das Resultat wird als Boolesche Ausgabe interpretiert.*
Ein Quanten Schaltkreis berechnet eine Funktion mit beschränktem Fehler, wenn für jede Eingabe die korrekte Ausgabe mit Wahrscheinlichkeit mindestens 2/3 produziert wird.
Ein Paar von Quanten Schaltkreisen berechnet eine Funktion im Las Vegas Sinn, wenn für jede Eingabe mit Wahrscheinlichkeit 1/2 der erste Schaltkreis akzeptiert, und wenn, falls der erste Schaltkreis akzeptiert, der zweite mit Sicherheit das korrekte Ergebnis berechnet.
Ein Quanten Schaltkreis berechnet eine Funktion exakt, wenn für jede Eingabe die korrekte Ausgabe mit Wahrscheinlichkeit 1 produziert wird.

Wir sind insbesondere an eingeschränkten Typen von Schaltkreisen, nämlich Quanten Formeln interessiert [Y93].

Definition 5.12 *Eine Quanten Formel ist ein Quanten Schaltkreis mit der zusätzlichen Eigenschaft, daß es für jede Eingabe höchstens einen Pfad gibt, der sie mit der Ausgabe verbindet. Die Länge oder Größe einer Quanten Formel ist die Anzahl ihrer Eingaben. Akzeptanzmodi sind wie für allgemeine Schaltkreise definiert.*

Abgesehen von den Booleschen Eingabevariablen darf eine Quanten Formel nur Boolesche Konstanten lesen. Es gibt nur eine abschließende Messung. Wie nennen das Modell aus [Y93] auch *pure* Quanten Formeln.
In [AKN98] wird ein allgemeineres Modell von Quanten Schaltkreisen eingeführt, das auf gemischten Zuständen arbeitet, wobei Dichtematrizen von Superoperator Gattern transformiert werden.

Definition 5.13 *Ein Superoperator Gatter* g *der Ordnung* (k, l) *ist eine spurbewahrende, vollständig positive, lineare Abbildung von den Dichtematrizen mit* k *Qubits auf die Dichtematrizen mit* l *Qubits.*
Ein Quanten Superoperator Schaltkreis ist ein gerichter azyklischer Graph, bei dem die inneren Knoten mit Superoperator Gattern markiert sind, die zu fan-in und fan-out der Knoten passen. Die Quellen sind mit Eingabevariablen oder Booleschen Konstanten markiert. Ein Gatter ist als Ausgabe markiert.
Eine Funktion wird wie folgt berechnet. Zu Beginn wird den Quellen je ein Boolescher Wert zugewiesen. Der Boolesche Wert 0 korrespondiert dann zu $|0\rangle\langle 0|$*, der Wert 1 zu* $|1\rangle\langle 1|$*. Der Gesamtzustand ist dann das Tensorprodukt der Zustände aller einzelnen Qubits.*

Dann werden die Gatter in einer beliebigen topologischen Reihenfolge auf die ihren Eingängen zugehörigen Dichtematrizen angewendet. Das bedeutet, daß derjenige Superoperator auf den Gesamtzustand angewendet wird, der aus dem Tensorprodukt von dem Gatter (auf den ihm zugeordneten Qubits) und dem identisch abbildenden Operator (auf den restlichen Qubits) besteht. Zum Schluß muß der Zustand des Ausgabe Qubits eine klassische Wahrscheinlichkeitsverteilung auf $|0\rangle$ und $|1\rangle$ sein und wird ausgegeben.

Der folgende Fakt aus [AKN98] rechtfertigt die Anwendung von Gattern in einer beliebigen topologischen Ordnung.

Fakt 5.7 *Sei C ein Quanten Superoperator Schaltkreis. C_1 und C_2 seien Mengen von Gattern, welche auf disjunkten Mengen von Qubits arbeiten. Dann gilt für alle Dichtematrizen ρ der Qubits im Schaltkreis, daß das Resultat von C_1 angewendet auf das Resultat von C_2 angewendet auf ρ das gleiche ist, wie das Resultat von C_2 angewendet auf das Resultat von C_1 angewendet auf ρ.*
Es seien zwei beliebige topologische Sortierungen der Gatter eines Quanten Superoperator Schaltkreises gegeben. Das Resultat der Anwendung der Gatter in der einen Reihenfolge ist identisch zu dem Resultat der Anwendung der Gatter in der anderen Reihenfolge.

Es ist noch ein weiterer Aspekt bei der Definition von Quanten Formeln zu beachten: um Zufallsvariablen, welche mehrmals gelesen werden dürfen, simulieren zu können, erlauben wir verallgemeinerten Quanten Formeln auch, Zufallseingabevariablen zu lesen. Wir beschränken uns auf fan-in 2, die Menge der Quantengatter mit fan-in 2 ist universal [BBC$^+$95].

Definition 5.14 *Eine verallgemeinerte Quanten Formel ist ein Quanten Superoperator Schaltkreis mit fan-out 1 und fan-in 1 oder fan-in 2 Gattern, welcher zusätzlich zu den Eingabevariablen der zu berechnenden Funktion auch noch Variablen aus einer Menge von Zufallseingabevariablen lesen darf. Jede dieser Zufallseingabevariablen entspricht einem Ensemble von puren Zuständen eines Qubits. Die Zustände der einzelnen Zufallsvariablen sind dabei nicht miteinander verschränkt. Eine Zufallsvariable darf mehrmals gelesen werden, dabei wird für alle Vorkommen der Variable immer derselbe pure Zustand aus dem Ensemble gewählt.*

Wie in [AKN98] gezeigt, impliziert das Kraussche Repräsentationstheorem, daß Quanten Superoperator Schaltkreise mit konstantem fan-in asymptotisch genauso effizient sind wie normale Quanten Schaltkreise mit konstanten fan-in. Das gleiche gilt auch für Formeln. Der wesentliche Unterschied zwischen puren und verallgemeinerten Quanten Formeln besteht also in der Verfügbarkeit von mehrfach lesbaren Zufallsvariablen.

Ein programmierbares Quanten Superoperator Gatter

In Simulationen von quantenmechanischen Berechnungsmodellen durch Quanten Kommunikationsprotokolle werden wir ein programmierbares Quanten Gatter benötigen. Das erlaubt uns, eine erlaubte Transformation als Programm zu kommunizieren (kodiert in einigen Qubits) und dann diese Operation auf einige andere Qubits anzuwenden.
Formal suchen wir nach einem unitären Operator G mit

$$G(|d\rangle \otimes |P_U\rangle) = U(|d\rangle) \otimes |P'_U\rangle.$$

Dabei ist $|P_U\rangle$ der „code“ des unitären Operators U, und $|P'_U\rangle$ ein Überbleibsel des Codes.
Die schlechte Nachricht ist, daß ein solches programmierbares Quanten Gatter nicht existiert, wie in [NC97] bewiesen. Man bemerke, daß solche Gatter im klassischen Fall trivialerweise existieren.

Fakt 5.8 *Wenn N verschiedene unitäre Operatoren (verschieden um mehr als eine globale Phase $e^{i\phi}$) durch ein programmierbares Quanten Gatter implementiert werden, so hat das Gatter mindestens* $\log N$ *Qubits Eingabe für das Programm.*

Weil es aber unendlich viele unitäre Operatoren auf einem Qubit gibt, gibt es kein programmierbares Quanten Gatter mit endlicher Programmlänge. Wir haben allerdings vorausgesetzt, daß das Gatter deterministisch arbeitet. Und tatsächlich ist es möglich, eine probabilistische Lösung des Problems zu finden. Wir beschreiben die Konstruktion von Nielsen und Chuang [NC97] und zeigen dann, wie man die Konstruktion so modifizieren kann, daß man ein Ergebnis mit beliebig kleiner Fehlerwahrscheinlichkeit erhält.
Der Einfachheit halber beschreiben wir die Konstruktion zuerst für unitäre Operationen auf einem Qubit.
Das Programm eines unitären Operators U ist

$$|P_U\rangle = \frac{1}{\sqrt{2}}(|0\rangle U|0\rangle + |1\rangle U|1\rangle).$$

Das Gatter hat als Eingabe $|d\rangle \otimes |P_U\rangle$. Das Gatter führt nun einfach eine Messung des ersten und zweiten Qubits in der Basis $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$ aus. Nachher wird das dritte Qubit ausgegeben.
Für einen Zustand $|d\rangle = a|0\rangle + b|1\rangle$ ist die Gesamteingabe des Gatters

$$\begin{aligned} & [a|0\rangle + b|1\rangle]\frac{|0\rangle U|0\rangle + |1\rangle U|1\rangle}{\sqrt{2}} \\ = \; & \frac{1}{2}\left[|\Phi^+\rangle(aU|0\rangle + bU|1\rangle) + |\Phi^-\rangle(aU|0\rangle - bU|1\rangle)\right. \\ & \left. +|\Psi^+\rangle(aU|1\rangle + bU|0\rangle) + |\Psi^-\rangle(aU|1\rangle - bU|0\rangle)\right]. \end{aligned}$$

Also produziert die Messung den korrekten Zustand auf dem dritten Qubit mit Wahrscheinlichkeit 1/4 und überdies gibt das Resultat der Messung die Korrektheit an.

Fakt 5.9 *Es gibt ein probabilistisches programmierbares Quanten Gatter mit m Eingabe Qubits für einen Zustand und $2m$ Eingabe Qubits für das Programm, das jede unitäre Operation auf m Qubits auszuführen versucht, und mit Wahrscheinlichkeit $1/2^{2m}$ Erfolg hat. Das Resultat einer internen Messung des Gatters zeigt an, ob das Ergebnis korrekt ist.*

Beweis: Für das Programm werden m EPR Paare $|\Phi^+\rangle$ verwendet. Die unitäre Transformation U wird auf die m Qubits angewendet, die aus einem Qubit aus jedem EPR Paar bestehen. Formal

$$|P_U\rangle = (I_m \otimes U) \bigotimes_{i=1}^{m} |\Phi^+_{i,i+m}\rangle,$$

wobei I_m der Identitäts Operator auf den ersten m Qubits ist, und $|\Phi^+_{i,i+m}\rangle$ ein EPR Zustand auf den Qubits i und $i+m$.
Das Gatter erhält als Eingabe das Programm sowie einen beliebigen Zustand $|d\rangle$ von m Qubits. Um die Transformation auszuführen, wird eine Messung in der Bell Basis auf alle m Paare bestehend aus dem $i$ten Qubit des Programms und dem $i$ten Qubit von $|d\rangle$ angewendet. Das Resultat ist eine Folge von m klassischen Werten, welche den gewählten Basisvektor anzeigen, wie auch der Zustand der restlichen m Qubits. Der Zustand der gemessenen Qubits ist nun wertlos, und wir betrachten nur noch die restlichen m Qubits.
Die Reihenfolge der Anwendung der unitären Transformation und der Messung ist gleichgültig, da sie auf verschiedenen Qubits arbeiten. Also betrachten wir den Zustand nach der Messung (und vor der unitären Transformation). Der Zustand, den wir erhalten, ist der folgende:

1. Ergibt die $i$te Messung Φ^+, so bleibt das $i$te Qubit unverändert.

2. Ergibt die $i$te Messung Φ^-, so wechselt die Amplitude von Basisvektoren mit einer 1 an Position i das Vorzeichen.

3. Ergibt die $i$te Messung Ψ^+, so tauschen die Basisvektoren $|x\rangle$ mit einer 1 an Position i den Amplitudenwert mit den Basisvektoren mit 0 an der Position i und ansonsten dem gleichen Wort wie x.

4. Ergibt die $i$te Messung Ψ^-, so treten 2 und 3 auf.

Erhalten wir also immer Φ^+, so ist der Zustand korrekt auf die m Ausgabe Qubits transportiert, und die Anwendung der unitären Transformation führt zum gewünschten Zustand. Dies geschieht mit Wahrscheinlichkeit 4^{-m}. Weiterhin kann abgelesen werden, wann dies der Fall ist. □

Die Erfolgswahrscheinlichkeit der obigen Konstruktion ist nicht gut, und ein großes Problem ist, daß der Zustand $|d\rangle$ nach Anwendung des Gatters zerstört ist, auch wenn es keinen Erfolg gab. Man kann also nicht durch Wiederholung die Erfolgswahrscheinlichkeit verbessern. Im folgenden zeigen wir, wie es möglich ist, den Eingabezustand zurückzuerhalten, und durch Wiederholung die Erfolgswahrscheinlichkeit beliebig groß zu machen.

Theorem 5.2 *Es gibt ein probabilistisches programmierbares Quanten Gatter, welches erlaubt, einen beliebigen erlaubten Superoperator von m auf n Qubits zu implementieren, so daß die Erfolgswahrscheinlichkeit $1 - 1/k$ ist, und das Programm $O((m+n)2^{4m+2n} \log k)$ Qubits enthält. Scheitert das Gatter, so kann man dies an dem Resultat einer internen Messung ablesen.*

Beweis: Wegen Fakt 5.1 kann ein m Qubits nach n Qubits abbildender Superoperator durch ein unitäres Gatter simuliert werden, das auf $2m+n$ Qubits angewendet wird, deren Zustand die Dichtematrix des Eingabezustands zusammen mit $m+n$ Hilfsqubits im puren Zustand $|0\rangle$ ist. Danach wird auf n Qubits reduziert. Weil letzteres bedeutet, einfach einige Qubits „fallenzulassen“, reicht es zu zeigen, wie ein unitäres Gatter auf $2m+n$ Qubits implementiert werden kann.
Für eine unitäre Transformation U sei $|P_U\rangle$ das Programm aus der Konstruktion zu Fakt 5.9. Das Programm der neuen Konstruktion ist dann $\bigotimes_{i=1}^{l} |P_U\rangle$, wobei $l = O(2^{4m+2n} \log k)$.
Jede ungerade numerierte Kopie von $|P_U\rangle$ wird für einen Versuch gebraucht, die Transformation U auszuführen. Jede gerade Kopie wird verwendet, um die Transformation bei einem Scheitern wieder umzukehren. Nach höchstens $O(2^{4m+2n} \log k)$ unabhängigen Versuchen ist die Erfolgswahrscheinlichkeit dann $1 - 1/k$ aufgrund der Chernov Ungleichung.
Das Berechnen der unitären Transformation U geht wie zuvor vonstatten: Ist das Ergebnis korrekt, so wird der Prozeß gestoppt. Ansonsten wird der Zustand $|d\rangle$ zurückberechnet. Dazu wenden wir die inverse Transformation von U an, das Programm für U^{-1} ist durch dasselbe Programm wie U gegeben, wenn man die ersten $2m+n$ Qubits mit den letzten $2m+n$ Qubits von $|P_U\rangle$ vertauscht. Dies gilt, weil $U|x\rangle = |x'\rangle$ eine Basis ergibt, und wir das Programm für $U^{-1}|x'\rangle$ zur Hand haben. Nach dieser Transformation haben wir einen Zustand, sowie 2 Folgen von je $2m+n$ Meßergebnissen, welche anzeigen, wie der Zustand zu $|d\rangle$ transformiert werden kann.
Betrachten wir eines der $2m+n$ Qubits. Wir haben zwei Meßresultate aus $\{\Phi^+, \Phi^-, \Psi^+, \Psi^-\}$ für diese Position. Ein Resultat Φ^+ bedeutet, daß nichts getan werden muß. Φ^- zeigt einen Vorzeichenwechsel an, der durch σ_z ausgeführt wird. Ψ^+ zeigt einen Tausch von 0 und 1 an, der durch σ_x kompensiert wird. Für Ψ^- hilft eine Anwendung von $i \cdot \sigma_y$. Führt man die entsprechenden Transformationen für beide Folgen und alle Positionen aus, so erhält man den ursprünglichen Zustand zurück.

Also kann man $|d\rangle$ zurückgewinnen und noch einen Versuch machen, bis ein Erfolg erzielt wurde. Mit Wahrscheinlichkeit $1-1/k$ tritt ein Erfolg während der l Versuche ein. Ansonsten gibt das Gatter ohne Ergebnis auf. □

5.3.3 Black Box Berechnungen

Wir betrachten Black-box Quanten Berechnungen wie z.B. in [G96], [BCW98], [BCWZ99] und beschreiben, wie obere Schranken für die Kommunikationskomplexität bestimmter Probleme aus Black Box Algorithms mittels eines Lemmas aus [BCW98] folgen.

Definition 5.15 *Es sei eine Eingabefunktion $a : \{0,1\}^l \to \{0,1\}$ gegeben. Ein a-Gatter ist die wie folgt definierte unitäre Abbildung:*

$$U_a : |i\rangle|b\rangle \mapsto |i\rangle|a(i) \oplus b\rangle,$$

wobei $i \in \{0,1\}^l$ und $b \in \{0,1\}$.
Ein Quanten Schaltkreis mit einer Black Box Eingabe arbeitet auf einer Menge von Qubits, die konstant als $|0,\ldots,0\rangle$ initiiert sind. Nun werden beliebige unitäre Transformationen wie auch a-Gatter sowie Messungen in einer festen Reihenfolge angewendet. Zum Schluß wird der Zustand eines Ausgabe Qubits gemessen. Der Schaltkreis berechnet eine Funktion $f(a)$ wie ein normaler Quanten Schaltkreis.
Die Kosten, oder die Anzahl der Fragen, eines Black Box Algorithmus sind gegeben durch die maximale Anzahl von a-Gattern, welche für eine Eingabe angewendet werden.

Mit einem gegebenen Algorithmus für ein Black Box Problem G kann man ein dazu verwandtes Kommunikationsproblem mit folgendem Resultat lösen [BCW98].

Fakt 5.10 *Sei G ein Black Box Problem (mit einer Eingabefunktion $a : \{0,1\}^{\log l} \to \{0,1\}$), das von einem Quanten Algorithmus mit höchstens t Fragen an ein a-Gatter gelöst wird. Sei L eine Funktion $K \times K \to \{0,1\}$. Dann gibt es ein Kommunikationsprotokoll, das folgendes Problem löst. A erhält $x \in K^l$, B erhält $y \in K^l$ und es wird $G(L(x_1,y_1),\ldots,L(x_l,y_l))$ berechnet. Das Protokoll kommuniziert höchstens $O(t(\log l + \log K))$ Qubits.*

Grovers Suchalgorithmus

Wir beschreiben nun den berühmtesten Quanten Black Box Algorithmus, Grovers Algorithmus zur Datenbanksuche. Die hier beschriebene Variante des Verfahrens stammt aus [BBHT96].
Das zu lösende Problem ist das folgende: Es gibt eine Liste von N Elementen, und eine Eigenschaft, die man für ein gegebenes Element leicht verifizieren

kann (z.B. durch einen klassischen Schaltkreis). Gesucht ist ein Element mit der Eigenschaft. Die Eingabe ist als eine Black Box f gegeben, welche durch die Nummer des Elements adressiert wird. Die Black Box antwortet dann mit ja oder nein, je nachdem, ob das Element die Eigenschaft erfüllt oder nicht. Jeder klassische (probabilistische) Black Box Algorithmus für das Problem benötigt $\Theta(N)$ Fragen.
Für den ersten Quanten Algorithmus nehmen wir an, daß im voraus die Anzahl t der Elemente mit der Eigenschaft bekannt ist. Es sei $n = \log N$, und N sei eine Potenz von 2.

SCHRITT 1: Wende die Hadamard Transformation auf ein Register im Zustand $|0^n\rangle$ an mit dem Ergebnis

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$

SCHRITT 2: Wende $\lceil \frac{\pi}{4}\sqrt{\frac{N}{t}} \rceil$ mal die unitäre Transformation der sogenannten Grover Iteration $G = -H_n R_n^{(1)} H_n V_f$ an.
SCHRITT 3: Messe das x-Register in der Standard Basis mit Resultat x_0.
SCHRITT 4: Teste, ob $f(x_0) = 1$.
Die Grover Iteration $G = -H_n R_n^{(1)} H_n V_f$ besteht aus der Komposition der Hadamard Transformation H_n mit dem Operator V_f, der f evaluiert, wobei f die Black Box ist, sowie mit der Rotationsmatrix $R_n^{(1)}$. Letztere ist wie folgt definiert: $R_n^{(1)}[i,j] = 0$ wenn $i \neq j$, $R_n^{(1)}[1,1] = -1$, und $R_n^{(1)}[i,i] = 1$ für alle $1 < i \leq 2^n$.
$D_n = -H_n R_n^{(1)} H_n$ wird oft als „inversion about the average“ bezeichnet, denn D_n bildet wie folgt ab:

$$D_n : \sum_{i=0}^{2^n-1} a_i|i\rangle \mapsto \sum_{i=0}^{2^n-1} (2E - a_i)|i\rangle,$$

wobei $E = \sum_{i=0}^{2^n-1} a_i/2^n$ der Durchschnitt der a_i ist.

Fakt 5.11 *Mit Wahrscheinlichkeit 1/2 findet obiger Algorithmus ein Element mit der gesuchten Eigenschaft, ansonsten gibt der Algorithmus auf. Die Kosten des Algorithmus sind $O(\sqrt{N})$.*

Nun betrachten wir den Fall, wenn die Anzahl t der Lösungen unbekannt ist.

SCHRITT 1: Wähle eine konstante Anzahl von Elementen zufällig und teste sie. Ist eine Lösung gefunden, so halte. Die Anzahl sei so groß, daß ansonsten mit hoher Wahrscheinlichkeit weniger als $3/4 \cdot N$ Elemente die gesuchte Eigenschaft haben.

SCHRITT 2: Setze $m = 1$, $\lambda = 6/5$.
SCHRITT 3: Wähle $j_0 \in \{1, \ldots, m\}$ zufällig.
SCHRITT 4: Wende j_0 mal Grovers Iteration auf $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ an.
SCHRITT 5: Messe x in der Standard Basis.
SCHRITT 6: Teste, ob x eine Lösung ist.
SCHRITT 7: Wähle $\min\{\lambda m, \sqrt{N}\}$ als neues m. Gehe zu 3.

Fakt 5.12 *[BBHT96] Obiger Algorithmus findet ein Element mit der gesuchten Eigenschaft nach $O(\sqrt{N/t})$ Fragen mit Wahrscheinlichkeit 1/2, wenn es $t > 0$ solche Elemente gibt, und hält sonst nach $O(\sqrt{N})$ Fragen.*

Wollen wir herausfinden, ob es eine Eingabe $x \in \{0,1\}^{\log N}$ gibt, die $f(x) \neq 0$ erfüllt, und haben wir f als Black Box gegebenen, so benötigt dieses ODER Problem nur $O(\sqrt{N})$ Fragen, während klassische Black Box Algorithmen (bzw. Entscheidungsbäume) mindestens $\Omega(N)$ Fragen benötigen, selbst bei Probabilismus mit beschränktem Fehler. Diese Beschleunigung durch einen Quantenalgorithmus ist optimal für das ODER Problem, siehe [G96].
Der Algorithmus für das ODER Problem ist ein Monte Carlo Algorithmus: Wenn es eine 1-Eingabe gibt, so wird sie mit hoher Wahrscheinlichkeit gefunden, wenn es keine gibt, so wird niemals eine gefunden. Wir erhalten keinen Las Vegas Algorithmus, und es ist bekannt, daß kein Quanten Las Vegas Algorithmus für ODER gegenüber einem deterministischen Algorithms eine Ersparnis bringen kann[BCMW98]. Weiterhin ist bekannt, daß kein Quanten Black Box Algorithmus mehr als polynomiell schneller sein kann als ein deterministischer Black Box Algorithmus, solange eine totale Funktion berechnet wird[BCMW98].
Grovers Algorithmus kann durch Wiederholung in seiner Erfolgswahrscheinlichkeit verbessert werden, so daß sich eine Erfolgswahrscheinlichkeit von $1 - \epsilon$ bei Verlust eines $O(\log(1/\epsilon))$ Faktors in der Anzahl der Fragen ergibt. Weiterhin ist bekannt, daß ein Suchalgorithmus mit Erfolgswahrscheinlichkeit $1 - \epsilon$ sogar nur $O(\sqrt{n \log(1/\epsilon)})$ Fragen braucht, siehe [BCWZ99].
Die Kosten der Algorithmus sind also $O(\sqrt{n \log s})$, um eine 1 in einem fan-in n ODER von Black Box Variablen zu finden (falls eine solche existiert) mit Erfolgswahrscheinlichkeit $1 - 1/s$.
Der obige Algorithmus kann auch auf die Berechnung von Prädikaten, die UND-ODER Formeln konstanter Tiefe auf Black Box Variablen sind, verallgemeinert werden [BCW98].

Fakt 5.13 *Sei F eine monotone Boolesche Formel mit unbeschränktem fan-in, Tiefe d und n Variablen, bei der der fan-in aller Gatter auf einer Ebene jeweils gleich ist. Dann kann F auf einer Black Box Eingabe ausgewertet werden durch einen Quanten Black Box Algorithmus, der $O(\sqrt{n} \log^{d-1})$ Fragen stellt, und sich mit Wahrscheinlichkeit höchstens 1/2 irrt. Weiterhin, wenn das oberste Gatter der Formel ein UND ist, und der Algorithmus 0*

ausgibt, so findet der Algorithmus ein Kind des UND Gatters, welches nicht erfüllt ist.

In [BCWZ99] wird ein Quanten Las Vegas Black Box Algorithmus für die Auswertung von UND-ODER Formeln beschrieben. Wir betrachten hier nur den Fall von Tiefe 2 Formeln mit einem UND von fan-in s an der Spitze und ODER Gattern vom fan-in r. Der Algorithmus setzt sich zusammen aus zwei parallel laufenden Teilalgorithmen, welche folgendes Verhalten haben. Der erste Algorithmus akzeptiert die Einsen der Funktion (oder gibt auf), der zweite Algorithmus verwirft die Nullen der Funktion (oder gibt auf). Beide Algorithmen bestimmen einen Zeugen für ihre Antwort, d.h. der erste Algorithmus findet eine 1-Eingabe für jedes ODER, der zweite ein ODER, dessen Eingaben alle 0 sind.
Beide Algorithmen verwenden Grovers Algorithmus als Unterroutine, wobei die Erfolgswahrscheinlichkeit auf $1-\epsilon/s$ verbessert ist (für konstantes $\epsilon > 0$), und die Anzahl der Lösungen unbekannt ist.
Der erste Algorithmus arbeitet wie folgt: für alle der s ODER Gatter wird Grovers Suchalgorithmus durchgeführt (mit Erfolgswahrscheinlichkeit $1 - \epsilon/s$). Das ergibt ein Zertifikat für jedes der ODER mit $O(s\sqrt{r \log s})$ Fragen und Wahrscheinlichkeit $1 - \epsilon$, wenn solche Zertifikate existieren. Ist kein Zertifikat gefunden, so gibt der Algorithmus auf.
Der zweite Algorithmus arbeitet wie folgt: zuerst wird multi-level Groversuche wie in Fakt 5.13 durchgeführt. Es wird ein nicht erfülltes ODER gefunden mit $O(\sqrt{sr}\log(sr))$ Fragen bei konstanter Wahrscheinlichkeit, falls eines existiert. Auf dem nicht erfüllten ODER werden dann alle r Eingaben abgefragt. Wieder gibt der Algorithmus auf, wenn kein Zertifikat gefunden ist.
Wenn man beide Algorithmen parallel ausführt, so wird die korrekte Antwort nach $O(s\sqrt{r}\log(sr) + r)$ Fragen mit Wahrscheinlichkeit 1/2 gefunden.

Fakt 5.14 *Ein fan-in s UND von fan-in r ODERs von Black Box Variablen kann von einem Quanten Las Vegas Algorithmus mit $O(s\sqrt{r}\log(sr) + r)$ Fragen und Erfolgswahrscheinlichkeit 1/2 ausgewertet werden.*

5.3.4 Quanten Einweg Automaten

Quantenmechanische endliche Automaten sind in [KW97], [MC97] und [Na99] definiert worden. Das Modell aus [MC97] erlaubt nur eine Messung am Schluß der Berechnung, während die anderen Modelle Messungen „unterwegs“ erlauben.
Unsere unteren Schranken gelten für alle diese Modelle, und sogar in einem stärkeren Modell von Quanten Einweg Automaten, das aus einem klassischen Automaten „mit eingebautem Quantenregister“ besteht.

Wir betrachten außer endlichen Quanten Automaten (qfa) mit beschränktem Fehler auch exakte qfa (welche ohne Fehler arbeiten) und Las Vegas qfa (welche mit Wahrscheinlichkeit ϵ aufgeben, sich aber niemals irren).
Im folgenden geben wir die Definitionen von Quanten Einweg Automaten. Zweiweg Automaten betrachten wir nicht. Zuerst geben wir die Standard Definition aus [KW97].

Definition 5.16 *Ein (endlicher) Quanten Einweg Automat ist ein Tupel* $(Q, \Sigma, q_0, F_a, F_r, \delta)$*, wobei* Q *eine endliche Menge von (Basis) Zuständen ist,* Σ *ein endliches Alphabet,* $\Gamma = \Sigma \cup \{\$\}$ *das Bandalphabet zusammen mit einem Symbol für das Ende der Eingabe,* q_0 *der Startzustand,* $F_a \subseteq Q$ *die Menge der akzeptierenden Zustände,* $F_r \subseteq Q$ *die Menge der verwerfenden Zustände und* $Q - F_a - F_r = F_n$ *die Menge der nichthaltenden Zustände.* $\delta : Q \times \Gamma \times Q \to \mathbb{C}$ *ist die Übergangsfunktion.*
Die puren Zustände des Automaten sind Norm 1 Vektoren im Hilbertraum $\mathbb{C}^{|Q|}$ *mit der Basis* $\{|q\rangle | q \in Q\}$*. Die Berechnung beginnt im Zustand* $|q_0\rangle$*. Der Automat liest einen Buchstaben* a*, ändert seinen Zustand gemäß einer durch*

$$V_a|q\rangle = \sum_{q' \in Q} \delta(q, a, q')|q'\rangle,$$

definierten und als unitär geforderten Abbildung.
Nach der unitären Transformation wird die Observable gemessen, die auf folgende Teilräume projiziert: den Raum der akzeptierenden Zustände, den Raum der verwerfenden Zustände, und den Raum der nichthaltenden Zustände. Nach jeder unitären Transformation wird also der Zustand gemessen, und entweder wird gehalten (und akzeptiert oder verworfen), oder nicht gehalten (und weitergearbeitet in einer Superposition nichthaltender Zustände). Wird weitergearbeitet, dann wird das nächste Zeichen gelesen. Wird das Ende der Eingabe erreicht, und wurde noch nicht gehalten, so wird der Automat mit zufälligem Ergebnis gestoppt.
Akzeptanz in den Modi beschränkter Fehler, Las Vegas und exakt ist wie üblich definiert.

Die obige Definition ergibt ein Modell mit einigen merkwürdigen Eigenschaften. Während von einem qfa nur eine reguläre Sprache erkannt werden kann, gibt es einige reguläre Sprachen, die nicht von einem qfa erkannt werden können [KW97]. Weiterhin gilt, daß ein qfa exponentiell kleiner als ein optimaler pfa für dieselbe Sprache sein kann [AF98], es können aber andererseits sogar deterministische Automaten exponentiell kleiner als jeder qfa für dieselbe endliche Sprache sein [Na99]. Auch ist es nicht möglich, die Akzeptanzwahrscheinlichkeit eines qfa beliebig zu verbessern [AF98].
Nayak [Na99] hat vorgeschlagen, ein Modell mit beliebigen Messungen zu definieren. Wir geben hier eine Definition, in der diese Messungen von einem klassischen Automaten gesteuert werden.

Definition 5.17 *Ein Quanten/Klassischer (endlicher) Automat, kurz qcfa, ist ein Tupel* $(Q, \Sigma, F_a, F_r, q_0, k, S, \delta)$*, wobei* Q *eine endliche Menge klassischer Zustände ist,* Σ *ein Alphabet,* $F_a \subseteq Q$ *die Menge der akzeptierenden Zustände,* $F_r \subseteq Q$ *die Menge der verwerfenden Zustände,* $F_a \cap F_r = \emptyset$*,* q_0 *der Startzustand,* k *die Dimension eines Hilbertraums* $\mathbb{C}^k$ *(welcher den Quantenspeicher des Automaten beschreibt). Die Menge* S *besteht aus Paaren von einer unitären Transformationen und einer Observablen für den Hilbertraum* $\mathbb{C}^k$*,* $\delta : Q \times \{1, \ldots, k\} \times \Sigma \to Q \times S$ *ist die Übergangsfunktion.*
Eine Berechnung beginnt in q_0*. Zu Beginn setzen wir* $m = 1$ *und der anfängliche Quantenzustand ist* $|0\rangle$*. Wird nun ein Buchstabe* $a \in \Sigma$ *gelesen, so wird die Übergangsfunktion auf den Zustand,* m *und* a *angewendet, was einen neuen Zustand und einen Superoperator (aus einer unitären Transformation und einer Messung) ergibt. Der Superoperator wird nun auf den Quantenzustand angewendet und ergibt einen neuen Quantenzustand sowie ein Meßergebnis, welches in* m *gespeichert wird (es gibt höchstens* k *verschiedene Resultate). Die Berechnung endet, nachdem die Eingabe gelesen ist. Der Automat akzeptiert, wenn der (klassische) Zustand in* F_a *liegt, der Automaten verwirft, wenn der Zustand in* F_r *liegt, sonst gibt der Automat auf.*
Die Größe des Automaten ist $k + |Q|$*.*
Akzeptanzmodi sind wie für klassische Automaten definiert.

Offensichtlich kann ein qcfa einen pfa effizient simulieren: Ein Zufallsbit wird durch eine Messung eingabeunabhängiger Quantenzustände bereitgestellt. Auch kann ein qfa direkt simuliert werden. Aufgrund der Resultate von [KW97], [ANTV99], [Na99] kann dieses Modell einige Sprachen erkennen, die qfa nicht erkennen können, und kann für einige Sprachen exponentiell kleiner sein als sowohl pfa wie auch qfa.

5.4 Quanten Kommunikationskomplexität

5.4.1 Überblick

Eine Übersicht bereits bekannter Resultate

Die erste Frage, die sich stellt, ist, ob die Quanten Kommunikationskomplexität einer Funktion kleiner sein kann als die probabilistische Kommunikationskomplexität. Ein Beispiel ist das Disjunktheitsproblem [BCW98], [KS92].

Fakt 5.15 $Q(DISJ_n) = O(\sqrt{n} \log n)$.
$R(DISJ_n) = \Omega(n)$.

Das Protokoll ist eine Anwendung von Grovers Quantensuchalgorithmus [G96] durch die generelle Simulation von Quanten Black Box Algorith-

men mittels Protokollen, siehe Abschnitt 5.3.3 und [BCW98]. Das Protokoll benötigt $O(\sqrt{n})$ Runden. Die untere Schranke ist aus [KS92], siehe Fakt 2.9. Dies ist der maximale bisher bekannte Unterschied zwischen Quantenkommunikation mit beschränktem Fehler und probabilistischer Kommunikation für eine totale Funktion.
Für Quanten Las Vegas Protokolle, die eine bestimmte Relation berechnen (also für das Paar von Eingaben eine der „passenden“ Ausgaben bestimmen müssen), ist folgendes bekannt [BCWZ99].

Fakt 5.16 *Es gibt eine Relation r so daß*
$Q_0(r) = O(n^{2/3} \log n)$.
$R(r) = \Omega(n)$.

Ronald de Wolf hat gezeigt [W00], daß nichtdeterministische Quanten Kommunikation wie folgt charakterisiert werden kann: Sei $M(f)$ die Kommunikationsmatrix von f. Der nichtdeterministische Rang von f ist $nrank(f)$, der minimale Rang einer Matrix, die man aus $M(f)$ erhält, wenn man 1 Einträge durch beliebige reelle Zahlen ungleich 0 ersetzt.

Fakt 5.17 $NQ(f) = \Theta(\log nrank(f))$.

Die wichtigste offene Frage der Quanten Kommunikationskomplexität ist daher, ob es einen superpolynomiellen Unterschied zwischen der Quanten Kommunikation mit beschränktem Fehler und der probabilistischen Kommunikation mit beschränktem Fehler für eine totale Funktion gibt.
Für partielle Funktionen ist die Situation die folgende. Das nächste Resultat ist von Raz [R99] bewiesen worden.

Fakt 5.18 *Es gibt eine partielle Funktion $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ so daß*
$Q(f) = O(\log n)$,
$R(f) = \Omega(n^{1/4}/\log n)$.

Das nächste Resultat ist aus [BCW98].

Fakt 5.19 *Es gibt eine partielle Funktion $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ so daß*
$Q_E^{(1)}(f) = O(\log n)$,
$R_0(f) = \Omega(n)$,
$R^{(1)}(f) = O(\log n)$.

Überblick über die Resultate dieser Arbeit

In der klassischen Kommunikationskomplexität gilt, daß es einen quadratischen Unterschied zwischen Las Vegas und deterministischer Kommunikationskomplexität gibt, und daß ein solcher Unterschied für totale Funktionen maximal ist. Dies folgt aus der Relation $D(f) = O(N(f) \cdot N(\neg f)) = O(R_0^2(f))$, Fakt 2.10.
Mittels Fakt 5.17 und Fakt 2.12 (siehe [L90]) [welcher aussagt, daß $D(f) = O(N(\neg f) \cdot \log trank(f))$, wobei $trank(f)$ die maximale Anzahl von Zeilen einer Teilmatrix der Kommunikationsmatrix angibt, welche nur Nullen über einer Diagonalen mit nur Einsen enthält] erhalten wir:

Theorem 5.3 *$D(f) = O(N(f) \cdot NQ(\neg f)) = O(N(f) \cdot Q_0(f))$ für alle totalen Funktionen f.*
$Q_0(f) = \Omega(D(f)/N(f))$ für alle totalen Funktionen f.

Der Grund ist, daß der nichtdeterministische Rang stets größer gleich $trank$ ist. Will man also einen superpolynomiellen Unterschied zwischen $Q_0(f)$ und $R_0(f)$ zeigen, so müssen $N(f)$ und $N(\neg f)$ groß sein. Wir erhalten die unteren Schranken

Theorem 5.4 *$Q_0(EQ_n) = \Omega(n/\log n)$ und*
$Q_0(DISJ_n) = \Omega(n/\log n)$.

Im nächsten Abschnitt beweisen wir

Theorem 5.5 *Es gibt eine totale Boolesche Funktion f mit der oberen Schranke $Q_0(f) = O(N^{10/11+\epsilon})$ (für alle Konstanten $\epsilon > 0$), während $R(f) = \Omega(N/\log N)$.*

In Abschnitt 5.4.3 untersuchen wir Quanten Einweg Kommunikation. Die Resultate sind wie folgt.

Theorem 5.7 *Für alle Funktionen f : $Q_\epsilon^{(1)}(f) \geq (1 - H(\epsilon))VC(f)$ und*
$Q_\epsilon^{(1,pub)}(f) \geq (1 - H(\epsilon))VC(f)/2$.

Es gilt also dieselbe untere Schranke, die in Fakt 3.2 für probabilistische Einweg-Kommunikation beschrieben wurde. Durch eine Analyse der Index Funktion folgt insbesondere ein exponentieller Unterschied zwischen deterministischer Zweiweg Kommunikation $D(IX_n) \leq \log n$ und der Quanten Einweg Kommunikation mit beschränktem Fehler $Q^{(1)}(IX_n) = \Omega(n)$.
Eine weitere Anwendung ist:

Korollar 5.1 *$Q_\epsilon^{(1)}(DISJ_n) \geq (1 - H(\epsilon))n$.*
$Q_\epsilon^{(1,pub)}(DISJ_n) \geq (1 - H(\epsilon))n/2$.

Teil eins des obigen Korollars ist unabhängig in [BW99] bewiesen worden. Auch Fakt 3.3 verallgemeinert sich auf den Fall quantenmechanischer Kommunikation.

Theorem 5.8 *Für alle totalen Funktionen f:*
$Q_E^{(1)}(f) = D^{(1)}(f)$,
$Q_{0,\epsilon}^{(1)}(f) \geq (1-\epsilon)D^{(1)}(f)$.

Theorem 5.9 *Für alle totalen Funktionen f:*
$Q_E^{(1,pub)}(f) = \lceil D^{(1)}(f)/2 \rceil$,
$Q_{0,\epsilon}^{(1,pub)}(f) \geq D^{(1)}(f)(1-\epsilon)/2$

Damit ist Interaktion notwendig, um Speedups wie in Theorem 5.5 zu erhalten.
Weiterhin geben wir noch eine Methode für untere Schranken an, welche es uns erlaubt, eine $\Omega(n/\log n)$ Schranke für die Einweg Quanten Kommunikationskomplexität bei beschränktem Fehler für die Größer-Gleich Funktion zu zeigen.

Korollar 5.4 $Q^{(1)}(GT_n) = \Omega(n/\log n)$.

Dies impliziert, daß es Funktionen gibt, für die die VC Schranke keine gute untere Schranke für die Quanten Einweg Kommunikationskomplexität ist.

5.4.2 Quanten Las Vegas Kommunikation

In diesem Abschnitt beweisen wir einen polynomiellen Unterschied zwischen der Quanten Las Vegas Kommunikationskomplexität und der probabilistischen Kommunikationskomplexität mit beschränktem Fehler für eine totale Boolesche Funktion. Diese Funktion beschreibt eine Teilmenge der Sprache zu $D_{n,s}$ aus Definition 2.7.
Eine untere Schranke werden wir mit Fakt 2.14 und Lemma 2.1 herleiten.
Zuerst jedoch wenden wir uns dem Quanten Protokoll zu. Wie betrachten Black Box Quanten Berechnungen und leiten eine Kommunikationssschranke mit Fakt 5.10 her.
Es geht uns darum, die Funktion $D_{n,s}$ zu berechnen, die wie folgt definiert ist

$$D_{n,s}(x_1x_2\ldots x_s, y_1y_2\ldots y_s) = 1 \iff \forall i : x_i, y_i \in \mathcal{P}(n^{32}, n) \text{ und } x_i \cap y_i \neq \emptyset.$$

Allerdings werden wir nur eine Teilmenge der Einsen der Funktion erkennen. Um einen Black Box Algorithmus wie in Fakt 5.10 anwenden zu können, brauchen wir eine Funktion L, welche die linken und rechten Eingaben „verbindet“. Die Funktion L ist in unserem Fall der Vergleich zweier Zahlen aus $K = \{1, \ldots, n^{32}\}$. Sei $m = n^{32}$.

Wir wollen das folgenden Black Box Problem auswerten: ein UND mit fan-in s von ODER Gattern mit fan-in r von Black Box Variablen. Wie in Abschnitt 5.3.3 beschrieben, gibt es einen solchen Algorithmus aus [BCWZ99] mit $O(s\sqrt{r}\log(sr)+r)$ Fragen und ohne Fehler, der mit Wahrscheinlichkeit $1/2$ aufgibt, siehe Fakt 5.14.
Nach Fakt 5.10 erhalten wir hiermit ein Quanten Protokoll für folgendes Problem: A und B erhalten jeder einen Vektor von Mengen (wir nehmen an, daß die Elemente jeder Menge aufsteigend sortiert sind). Das Protokoll akzeptiert, wenn für alle Paare x_i, y_i von Mengen r bestimmte Vergleiche mindestens einen Erfolg haben. Anders gesagt, es wird eine Teilmenge der Einsen von $D_{n,s}$ akzeptiert.
Es ist unser Ziel, eine große solche Teilmenge zu akzeptieren. Eine erste Idee ist es, das vollständige Problem zu lösen. Dazu vergleicht man alle Paare von Elementen für alle s Positionen. Man setzt also $r = n^2$. Ein solcher Ansatz ist aber nicht effizient genug.
Intuitiv gesehen sollte es für zwei zufällige Mengen jedoch ausreichen, weniger Vergleiche zu machen: das i größte Element einer zufälligen Menge liegt irgendwo in der Nähe von $i \cdot m/n$, wenn die Eingabe uniform zufällig gezogen ist. Für jedes i und eine zufällige Teilmenge der Größe n aus einem Universum der Größe m gilt folgendes.

$$\begin{aligned}&\Pr(\text{Es gibt nicht } i \pm c \cdot n^{2/3} \text{ Elemente}\\ &\text{in dem Intervall } [1,\ldots,im/n]) = O(n)/(c^2 n^{4/3}).\end{aligned}$$

Dies gilt nach der Chebyshef Ungleichung, denn die Varianz der Zufallsvariablen, welche die Anzahl der kleinen Elemente zählt, ist durch $O(i) = O(n)$ beschränkt: ihre Verteilung ist hypergeometrisch.

$$\begin{aligned}&\Pr(\forall i \in [1,\ldots,n] \text{ gibt es}\\ &i \pm c \cdot n^{2/3} \text{ Elemente in } [1,\ldots,im/n])\\ &\geq \Pr(\forall i \in [1, n^{2/3}, 2n^{2/3}, \ldots, n] \text{ gibt es}\\ &i \pm (c+1) \cdot n^{2/3} \text{ Elemente in}[1,\ldots,im/n])\\ &\geq (1 - \tfrac{O(1)}{c^2 n^{1/3}})^{n^{1/3}} \geq 1 - \epsilon\end{aligned}$$

für jedes konstante ϵ $(1 > \epsilon > 0)$ und für eine genügend große Konstante c. Dabei gilt die erste Ungleichung, weil die Bedingung für i und $i + n^{2/3}$ die Bedingung für alle dazwischenliegenden j mit größerer Konstante impliziert. Die zweite Ungleichung gilt, weil die Wahrscheinlichkeit, daß die Bedingung für i gilt, höchstens so groß ist wie die Wahrscheinlichkeit, daß die Bedingung für i gilt unter der Voraussetzung, daß sie bereits für alle $j < i$ gilt. Es ist nun weiterhin wichtig, daß die Ungleichung auch gilt, wenn wir irgendein Element der Menge fixieren und nur $n-1$ Elemente zufällig wählen.

Für einen zufälligen Vektor von s Mengen nehmen wir also im folgenden an, daß mit Wahrscheinlichkeit $1/2^{\epsilon' s}$ alle Mengen die Eigenschaft haben, daß für alle i gilt, daß zwischen dem $i$ten Element in der (sortierten) Mengen und im/n nur $O(n^{2/3})$ andere Elemente liegen. Dies gilt auch für beide in einem Paar solcher Vektoren, welche unter der Einschränkung gezogen sind, eine Eins von $D_{n,s}$ zu sein, denn wir können für alle Positionen ein Element fixieren, in dem sich beide Mengen der Position schneiden, und alle anderen Elemente zufällig wählen.
Angenommen, zwei Mengen x, y schneiden sich auf einem Element a, welches x^i und y^j ist (das $i$te bzw. $j$te Element der Mengen). Angenommen $j > i+8cn^{2/3}$. Um dies zum Widerspruch zu führen wählen wir $\lceil i+(j-i)/2 \rceil = k$. Wenn nun $a \leq km/n$, dann gibt es höchstens $cn^{2/3}$ Elemente y^l zwischen a und jm/n und daher höchstens $cn^{2/3}$ Elemente y^l zwischen km/m und jm/n, obwohl $j - k > 4cn^{2/3}$ und wir erhalten den Widerspruch, daß es bis jm/n höchstens $k+2cn^{2/3} \leq j-2cn^{2/3}$ von den y^l gibt. Einen symmetrischen Widerspruchsbeweis gibt es für $a \geq km/n$. Also gilt $|j - i| = O(n^{2/3})$.
Wenn wir also für alle s Paare $t = 1, \ldots, s$ von Mengen x_t, y_t das i größte Element x_t^i jeweils mit den Elementen y_t^k von $i - cn^{2/3}$ bis $i + cn^{2/3}$ vergleichen, dann können wir eine Teilmenge der Größe $1/2^{\epsilon' s}$ aller Einsen von $D_{n,s}$ akzeptieren, ohne je eine 0 zu akzeptieren. Nach Fakt 5.10 haben wir ein Quanten Las Vegas Protokoll für diese Teilmenge der Einsen von $D_{n,s}$ gefunden, und die Teilmenge ist groß genug, so daß die untere Schranke von Lemma 2.1 von $\Omega(ns)$ gilt. Auf diese Weise ist durch den Vergleichsbaum die Funktion f der Behauptung definiert.
Was ist die Komplexität des Quanten Protokolls? Es wird ein Baum aus einem fan-in s UND von fan-in r ODERs ausgewertet. r ist $n \cdot O(n^{2/3})$. Daher brauchen wir die Kommunikation $O(s\sqrt{r}\log(sr)\log n + r\log n) = O(sn^{5/6}\log(sr)\log n + n^{5/3}\log n)$. Wählt man $s = n^{5/6}$, so ist die Eingabelänge $N = \Theta(n^{11/6}\log n)$, und wir erhalten folgendes Theorem.

Theorem 5.5 *Es gibt eine totale Boolesche Funktion f mit der oberen Schranke $Q_0(f) = O(N^{10/11+\epsilon})$ (für alle Konstanten $\epsilon > 0$), während $R(f) = \Omega(N/\log N)$.*

5.4.3 Quanten Einweg Kommunikation

Wir wollen nachweisen, daß die VC-Dimension auch eine untere Schranke für Quanten Einweg Protokolle mit beschränktem Fehler ist. Um dies zu zeigen, geben wir eine Reduktion von der Index Funktion auf eine jede Funktion mit hoher VC-Dimension an. Es ist leicht zu sehen, daß $VC(IX_n) = n$, und deshalb ist die probabilistische Einweg Kommunikation mit beschränktem Fehler für die Index Funktion hoch.
Das Problem des *random access quantum coding* wurde in [ANTV99] und [Na99] analysiert. In einem n, m, ϵ-random access quantum code müssen

alle Booleschen n-Bit Worte x auf je m Qubits abgebildet werden, so daß es für $i = 1, \ldots, n$ je eine Observable gibt, deren Messung das Bit x_i mit Wahrscheinlichkeit $1 - \epsilon$ ergibt. Dabei darf ein Code ein gemischter Zustand sein. Nayak [Na99] hat gezeigt

Fakt 5.20 *Für jedes n, m, ϵ-random access quantum coding gilt $m \geq (1 - H(\epsilon))n$.*

Es ist nun leicht zu beobachten, daß das Problem des random access quantum coding äquivalent dazu ist, ein Quanten Einweg Protokoll für die Index Funktion zu finden. Wenn es ein solches Protokoll gibt, kann man die Nachrichten als (gemischte Zustände) von Codes nehmen, und wenn es einen Code gibt, so kann man jedes Codewort als Nachricht wählen. Wir können also eine untere Schranke für IX_n im Einweg Modell ohne Verschränkung folgern. Wir fügen aber noch einen weiteren Beweis auf, aus dem auch eine untere Schranke im Modell mit Verschränkung folgt.

Theorem 5.6 $Q_\epsilon^{(1)}(IX_n) \geq (1 - H(\epsilon))n$.
$Q_\epsilon^{(1,pub)}(IX_n) \geq (1 - H(\epsilon))n/2$.

Beweis: Die Dichtematrix des Zustands der Nachrichten M und der uniform zufälligen Eingaben X sei σ_{XM}, der Zustand einer zufälligen Nachricht ist dann σ_M. Nun ist jedes Bit mit Wahrscheinlichkeit $1 - \epsilon$ dekodierbar und daher folgt $S(X_i : M) \geq 1 - H(\epsilon)$ für alle i. Das gilt, weil $S(X_i : M)$ der Holevo Information des folgenden Ensembles entspricht:

$$\sigma_{i,0} = \sum_{x:x_i=0} \frac{1}{2^{n-1}} \sigma_M^x$$

mit Wahrscheinlichkeit $1/2$ und

$$\sigma_{i,1} = \sum_{x:x_i=1} \frac{1}{2^{n-1}} \sigma_M^x$$

mit Wahrscheinlichkeit $1/2$, wobei σ_M^x die Dichtematrix der Nachrichten auf Eingabe x sei. Die durch eine Messung von σ_M erreichbare Information über x_i ist aber mindestens $1 - H(\epsilon)$ wegen Fanos Ungleichung Fakt 2.30, und daher ist die Holevo Information des Ensembles mindestens $1 - H(\epsilon)$ und somit $S(X_i : M) \geq 1 - H(\epsilon)$.
Aber dann gilt $S(X : M) \geq (1 - H(\epsilon)n$ (weil alle X_i voneinander unabhängig sind). $S(X : M) \leq S(M)$ wegen Lemma 5.2, da X und M nicht miteinander verschränkt sind. Also ist die Anzahl der Qubits in M mindestens $(1 - H(\epsilon))n$.
Nun analysieren wir die Komplexität von IX_n im Kommunikationsmodell mit Verschränkung.

Die Dichtematrix des Zustands von einer gleichverteilten Eingabe X, der Nachricht M und der Qubits E_A und E_B, welche miteinander verschränkt und im Besitz von A und B sind, ist $\sigma_{XME_AE_B}$. Wenn wir den Zustand um X sowie um den Teil von A an den verschränkten Qubits reduzieren, so erhalten wir σ_{ME_B}, den Zustand der Qubits, die B zugänglich sind. Nun ist jedes Bit von X mit Wahrscheinlichkeit $1-\epsilon$ dekodierbar, also gilt $S(X_i : ME_B) \geq 1-H(\epsilon)$ für alle i wie oben. Aber dann gilt auch $S(X : ME_B) \geq (1-H(\epsilon)n$, weil alle X_i voneinander unabhängig sind.
$S(X : ME_B) = S(X : E_B) + S(X : M|E_B) \leq 2S(M)$ unter Verwendung der Araki-Lieb Ungleichung. Es gilt $S(X : E_B) = 0$. So muß die Anzahl der Qubits in M mindestens $(1-H(\epsilon))n/2$ sein. □

Theorem 5.7 *Für alle Funktionen* f : $Q_\epsilon^{(1)}(f) \geq (1-H(\epsilon))VC(f)$ *und* $Q_\epsilon^{(1,pub)}(f) \geq (1-H(\epsilon))VC(f)/2$.

Beweis: Wir beschreiben eine Reduktion von der Index Funktion. Es gelte $VC(f) = d$, es gibt also eine Menge $S = \{s_1, \ldots, s_d\}$ von Eingaben für B, welche von der Menge der Funktionen $f(x,.)$ zerschmettert wird. Wir reduzieren IX_d auf f. Für jedes $R \subseteq S$ sei c_R der Inzidenzvektor von R (mit Länge d). c_R wird als Eingabe für A im Index Problem IX_d verwendet. Für jedes R wählen wir ein x_R, welches diese Teilmenge von dem Rest von S trennt, d.h. $f(x_R,.)$ trenne R von $S-R$. Die Reduktion bildet c_R auf x_R ab. Bs Eingaben i zum Index Problem IX_d werden auf die s_i abgebildet. So ist $f(x_R, s_i) = 1 \iff s_i \in R \iff c_R(i) = 1$.
Auf diese Weise muß ein Quanten Protokoll für f implizit IX_d lösen. Nach Theorem 5.6 folgen beide untere Schranken. □

Korollar 5.1 $Q_\epsilon^{(1)}(DISJ_n) \geq (1-H(\epsilon))n$.
$Q_\epsilon^{(1,pub)}(DISJ_n) \geq (1-H(\epsilon))n/2$.

Das erste Resultat wurde unabhängig auch in [BW99] gezeigt.
Nun wenden wir uns der exakten und Quanten Las Vegas Einweg Kommunikationskomplexität von totalen Funktionen zu. Für klassische Einweg Protokolle und totale Funktionen ist bekannt [DHRS97], daß Las Vegas Kommunikation höchstens den Faktor 1/2 effizienter ist als deterministische Kommunikation.

Theorem 5.8 *Für alle totalen Funktionen* f *gilt:*
$Q_E^{(1)}(f) = D^{(1)}(f)$,
$Q_{0,\epsilon}^{(1)}(f) \geq (1-\epsilon)D^{(1)}(f)$.

Beweis: Sei $row(f)$ die Anzahl der verschiedenen Zeilen in der Kommunikationsmatrix für $f(x,y)$. Nach Fakt 3.1 gilt $D^{(1)}(f) = \lceil \log row(f) \rceil$. Wir

nehmen im folgenden an, daß die Kommunikationsmatrix nur paarweise verschiedene Zeilen besitzt.
Wir zeigen, daß ein Las Vegas Protokoll mit Aufgabewahrscheinlichkeit $\epsilon \geq 0$ für ein f mit $row(f) = R$ Nachrichten mit einer von Neumann Entropie von mindestens $(1-\epsilon)\log R$ benutzt, wenn es auf einer uniform zufälligen Eingabe gestartet wird. Eingaben von A werden mit Zeilen der Kommunikationsmatrix identifiziert. Wir schließen dann, daß der Hilbertraum der Nachrichten eine Dimension von mindestens $R^{1-\epsilon}$ haben muß und daher mindestens $(1-\epsilon)\log R$ Qubits gesendet werden müssen. Das ergibt die zweite untere Schranke. Die obere Schranke der ersten Behauptung ist trivial, die untere Schranke der ersten Behauptung folgt aus der zweiten Behauptung für $\epsilon = 0$.
Wir beschreiben zuerst einen Zufallsprozeß, in dem Zeilen der Kommunikationsmatrix Bit für Bit festgelegt werden. Sei p die Wahrscheinlichkeit einer 0 in Spalte 1 (das heißt die relative Häufigkeit der Nullen in Spalte 1). Dann wird eine 0 mit Wahrscheinlichkeit p gezogen, eine 1 mit Wahrscheinlichkeit $1-p$. Nachher wird die Menge der Zeilen partitioniert in die Menge I_0 der Zeilen, die mit einer 0 beginnen und die Menge I_1 der Zeilen, die mit einer 1 beginnen. Ist $x_1 = b$ gezogen, fährt man mit I_b und der nächsten Spalte fort.
Sei ρ_y die Dichtematrix des folgenden gemischten Zustandes: Es wird gleichverteilt die Nachricht zu einer Zeile, die mit y beginnt, gewählt. Jede Dichtematrix einer Nachricht zu einer solchen Zeile wird also mit derselben Wahrscheinlichkeit gewählt und ρ_y ist die Dichtematrix einer zufälligen solchen Nachricht.
Die Wahrscheinlichkeit, daß eine 0 nach y gewählt wird heiße p_y, und die Anzahl der verschiedenen Zeilen, welche mit y beginnen, sei row_y.
Wir wollen per Induktion zeigen, daß $S(\rho_y) \geq (1-\epsilon)\log row_y$. Sicher gilt $S(\rho_y) \geq 0$ für jedes y. Damit gilt der Induktionsanfang.
Mit Lemma 5.1 erhalten wir $S(\rho_y) \geq p_y S(\rho_{y0}) + (1-p_y)S(\rho_{y1}) + (1-\epsilon)H(p_y)$.

$$\begin{aligned} S(\rho_y) &\geq p_y((1-\epsilon)\log row_{y0}) \\ &\quad +(1-p_y)((1-\epsilon)\log row_{y1}) + (1-\epsilon)H(p_y) \\ &= (1-\epsilon)[p_y \log(p_y row_y) \\ &\quad +(1-p_y)\log((1-p_y)row_y) + H(p_y)] \\ &= (1-\epsilon)\log row_y. \end{aligned}$$

Wir schließen, daß $S(\rho) \geq (1-\epsilon)\log row(f)$. □
Wir betrachten nun wieder das Modell mit Verschränkung.

Theorem 5.9 *Für alle totalen Funktionen f:*
$Q_E^{(1,pub)}(f) = \lceil D^{(1)}(f)/2 \rceil$,
$Q_{0,\epsilon}^{(1,pub)}(f) \geq D^{(1)}(f)(1-\epsilon)/2$.

Die obere Schranke gilt wegen des superdense coding. Statt der unteren Schranken des obigen Theorems beweisen wir eine stärkere Aussage. Wir betrachten ein erweitertes Modell von Einweg Kommunikation, das später (bei den Quanten Formeln) nützlich sein wird. In einem Nichtstandard Einweg Quantenprotokoll dürfen A und B in beliebig vielen Runden miteinander kommunizieren. B darf allerdings keine Nachricht schicken, so daß die von Neumann Information der Eingabe von A und der A zugänglichen Qubits über Bs Eingabe größer als 0 ist. Als Komplexität wird die Länge der Nachrichten von A zu B gewählt. Das Modell ist mindestens so mächtig wie das Modell mit verschränkten Qubits, da B zuerst z.B. beliebig viele Qubits aus EPR-Paaren senden kann, worauf A eine Nachricht schickt.

Lemma 5.4 *Für alle Funktionen f muß ein Nichtstandard Quanten Einweg Protokoll bei beschränktem Fehler mindestens $(1-H(\epsilon))VC(f)/2$ Qubits von A zu B kommunizieren.*
Für alle totalen Funktionen f muß ein Nichtstandard Quanten Einweg Protokoll

- *ohne Fehler mindestens $\lceil D^{(1)}(f)/2 \rceil$ Qubits von A zu B kommunizieren,*
- *im Las Vegas Modus mit Erfolgswahrscheinlichkeit $1-\epsilon$ mindestens $D^{(1)}(f)(1-\epsilon)/2$ Qubits von A zu B kommunizieren.*

Beweis: Für die erste Aussage reicht es wieder, die Komplexität des Index Problems zu untersuchen. Sei σ_{XYPQ} der Zustand von zufälligen Eingaben X, Y für A und B, sowie der Qubits P und Q im Besitz von A und B. Zum Schluß, wenn B das Ergebnis bestimmt, muß gelten $S(X_Y : YQ) \geq 1-H(\epsilon)$, da man aus den Qubits bei B und der rechten Eingabe den Wert X_Y mit Wahrscheinlichkeit $1-\epsilon$ bestimmen kann. Es gilt immer im Protokoll, daß $S(XP : Y) = 0$. Sei $\rho_P^{X=x,Y=y}$ die Dichtematrix von P für festgelegte $X = x$ und $Y = y$. $\rho_P^{X=x,Y=y}$ entsteht durch Fixierung von $X = x$ und $Y = y$ und Reduzierung des Zustands der benutzten Qubits um Q. Da $S(XP : Y) = 0$ gilt für alle x, y, y', daß $\rho_P^{X=x,Y=y} = \rho_P^{X=x,Y=y'}$. Das Hinzufügen des Zustands von Q bewirkt dann eine „Purifikation“ von $\rho_P^{X=x,Y=y}$. Nach folgendem Fakt aus [M97] und [LC98] sind alle y und Q Zustände „gleich“ aus der Sicht von A.

Fakt 5.21 *Angenommen $|\phi_1\rangle$ und $|\phi_2\rangle$ seien zwei pure Zustände in einem Hilbertraum $H \otimes K$, so daß $Tr_K|\phi_1\rangle\langle\phi_1| = Tr_K|\phi_2\rangle\langle\phi_2|$.*
Dann gibt es eine unitäre Transformation U auf K, welche $I \otimes U|\phi_1\rangle = |\phi_2\rangle$ erfüllt (für den Identitätsoperator I auf H).

Es gibt also eine unitäre Transformation, die B lokal auf seine Qubits anwenden kann, so daß $\rho_{PQ}^{X=x,Y=y}$ einfach zu $\rho_{PQ}^{X=x,Y=y'}$ gewechselt werden

kann. Damit muß aber für alle i gelten $S(QY : X_i) \geq 1 - H(\epsilon)$, und daher $S(X : QY) \geq (1 - H(\epsilon))n$.
Zu Beginn ist $S(X : QY) = 0$. Dann geht das Protokoll o.B.d.A. so vor, daß jeweils einer der Spieler eine (von seiner Eingabe abhängige) unitäre Transformation auf seine Qubits anwendet und dann ein Qubit zum anderen Spieler sendet. Da die Information durch lokale unitäre Transformationen nicht ansteigen kann, reicht es, sich den Einfluß des Verschickens eines Qubits anzusehen. Sendet B zu A ein Qubit, so steigt die Information $S(X : QY)$ nicht an. Sendet A zu B ein Qubit, so wird Q um ein Qubit M erweitert und $S(X : QMY) \leq S(X : QY) + S(XQY : M) \leq S(X : QY) + 2S(M) \leq S(X : QY) + 2$ wegen Fakt 5.3. Daher kann die Information nur dann steigen, wenn A ein Qubit an B sendet, und nur jeweils um 2. Also folgt die Behauptung.
Nun zum zweiten Teil. Wir betrachten dieselbe Situation wie in Theorem 5.8. Es sei σ_P^{rc} die Dichtematrix der Qubits P im Besitz von A unter der Voraussetzung, daß die Zeile r und die Spalte c ist. Das Hinzufügen des Zustands der Qubits Q von B purifiziert dann σ_P^{rc}. Wieder ist $\sigma_P^{rc} = \sigma_P^{rc'}$ für alle r, c, c', und nach Fakt 5.21 gilt für alle c und alle zugehörigen Zustände von Q, daß B lokal zwischen ihnen wechseln kann. Es ist B also möglich, für eine beliebige Spalte das Ergebnis zu erhalten.
Die Wahrscheinlichkeit, daß eine 0 nach y gewählt wird, heiße wieder p_y, und die Anzahl der verschiedenen Zeilen, welche mit y beginnen, sei row_y. ρ_y enthalte den Zustand von Bs Qubits am Ende des Protokolls. Sicher gilt $S(\rho_x) \geq 0$ für jedes x. Da B durch eine unitäre Transformation seine Spalte (und den Zustand von Q) einfach wechseln kann, ohne daß A dies bemerkt, kann B nach Ablauf des Protokolls eine beliebige Spalte messen, immer mit der Erfolgsgarantie des Protokolls. Mit Lemma 5.1 erhalten wir $S(\rho_y) \geq p_y S(\rho_{y0}) + (1 - p_y)S(\rho_{y1}) + (1 - \epsilon)H(p_y)$ und wieder $S(\rho_y) \geq (1 - \epsilon) \log row_y$. Damit ist die Holevo Information des Ensembles, in dem ρ_x mit Wahrscheinlichkeit $1/row(f)$ gewählt wird mindestens $(1-\epsilon) \log row(f)$. Sei σ_{XPQ} die Dichtematrix von Zeilen, Qubits bei A und B. Es folgt $S(Q : X) \geq (1-\epsilon) \log row(f)$ und wie oben müssen mindestens halb so viele Qubits von A zu B gesendet werden. □

Eine weitere untere Schranke

In diesem Abschnitt beschreiben wir eine weitere untere Schranke für Quanten Einweg Kommunikationskomplexität bei beschränktem Fehler. Diese untere Schranke ist in einigen Fällen wesentlich besser als die VC-Dimension, aber niemals viel schlechter.
Sei M die Kommunikationsmatrix von f, $row(M)$ sei die Anzahl der verschiedenen Zeilen in M. Wir sagen, daß sich eine Menge R' von Zeilen aus einer Zeilenmenge R durch das Fixieren von Spalten ergibt, wenn es Spalten $c_1, \ldots, c_s$ und Bits $b_1, \ldots, b_s$ für ein $s \in \mathbb{N}$ gibt, und R' genau die Zeilen aus R enthält, die an Spalte c_i den Wert b_i haben, für alle $i = 1, \ldots, s$.

Die Methode arbeitet wie folgt:

- Wähle eine Teilmenge M' der Zeilen in M, so daß die konstruierte Schranke maximiert wird.
- Sei $h \in (0,1)$ minimal mit der folgenden Eigenschaft:

  Für alle Teilmengen M'' der Zeilen von M' mit $|M''| \geq 2$, welche man durch das Fixieren von Spalten erhält, gibt es eine Spalte c so daß:

  Wenn U bzw. V die Anzahl der Einsen bzw. Nullen in c eingeschränkt auf die Zeilen in M'' bezeichnet, gilt $\max\{U/(U+V), V/(U+V)\} \leq h$.
- Setze $bound(f) = \log_2 row(M')/\log_2 \log_{1/h} row(M')$.

Theorem 5.10 *Für alle totalen f: $Q^{(1)}(f) = \Omega(bound(f))$.*

Beweis: Angenommen P ist ein Quanten Einweg Protokoll für f mit Fehler $1/3$ und C Qubits Kommunikation. Die Schranke *bound* werde erreicht mit den Parametern M', h. Dann kann die Fehlerwahrscheinlichkeit auf $\epsilon \leq 1/(64 \log^2_{1/h} row(M'))$ verringert werden durch eine parallele Mehrfachausführung des Protokolls. Das erhöht die Kommunikation auf höchstens $O(C \cdot \log_2 \log_{1/h} row(M'))$. Nun reicht es zu zeigen, daß $\Omega(\log_2 row(M'))$ Qubits Kommunikation für f bei diesem Fehler notwendig sind.
Wir argumentieren, daß das Protokoll mit dem kleinen Fehler benutzt werden kann, um eine beliebige Zeile von M' zu B zu kommunizieren bei einer hohen Erfolgswahrscheinlichkeit über die Resultate von Messungen von B. Da es $row(M')$ verschiedene Worte gibt, die so kommuniziert werden, muß nach Holevos Theorem Fakt 5.2 die Kommunikation für das modifizierte Protokoll $\Omega(\log_2 row(M'))$ sein und für das ursprüngliche Protokoll ist damit die Kommunikation $C = \Omega(\log_2 row(M')/\log_2 \log_{1/h} row(M'))$.
Es muß also gezeigt werden, daß B, gegeben eine Nachricht von A, die Zeile von A mit hoher Wahrscheinlichkeit rekonstruieren kann. B geht wie folgt vor:
B wertet die Nachricht an einer Spalte aus. Zu jeder Spalte ist im Protokoll eine Observable assoziiert, deren Anwendung auf eine Nachricht den Funktionswert mit hoher Wahrscheinlichkeit ergibt. B mißt die Nachricht in der Absicht, so viele Zeilen wie möglich aus der Menge möglicher Zeilen zu entfernen. Dazu wählt B die Observable einer Spalte, welche $\max\{U/(U+V), V/(U+V)\}$ minimiert. Dieser Wert ist jedoch immer höchstens h. Ist das Ergebnis 1, so werden V Zeilen entfernt, ansonsten U Zeilen, immer jedoch bleibt ein Bruchteil von höchstens h der Zeilen übrig. Somit reichen $\log_{1/h} row(M')$ Fragen, wenn alle korrekt beantwortet werden, um eine Zeile eindeutig zu bestimmen.
Eine Analyse wie in [ANTV99] zeigt, daß die Wahrscheinlichkeit, daß ein Fehler irgendwann gemacht wird, höchstens $4\sqrt{\epsilon}$ ist (dies ist ähnlich zu dem

Fall des seriellen random access quantum codings, weil die Fragen adaptiv sind).
Wir nehmen an, daß die Nachrichten von A pure Zustände sind. A wendet einen von ihrer Eingabe abhängigen Superoperator auf einen Nullzustand an, um eine Nachricht zu erzeugen. Nach Fakt 5.1 kann sie auch eine unitäre Transformation verwenden, auf 3 mal mehr Qubits als die ursprüngliche Nachrichtenlänge. B kann dann durch Reduzieren um einige Qubits die ursprüngliche Nachricht erzeugen.
Bs Messungen (welche von seiner Booleschen Eingabe abhängen) teilen den Raum $\mathbb{C}^{2^k}$ von Nachrichten und einigen leeren Qubits in zwei Teilräume W_i^0 und W_i^1 auf für Bs $i$te Eingabe. Sei $|\phi\rangle$ der (von der Zeile abhängige) Zustand der Nachricht. Jeder von B angewendete Superoperator kann simuliert werden, indem man einige leere Qubits hinzufügt, eine unitäre Transformation ausführt, und den Zustand um einige Qubits reduziert. Da das Ergebnis eine klassische Verteilung auf einem Bit ist, reicht es aus, wenn B eine orthogonale Messung auf der um einige leere Qubits erweiterten Nachricht $|\phi\rangle|0^l\rangle$ durchführt. Dieser Vektor wird zerlegt in $|\phi_i^0\rangle + |\phi_i^1\rangle$, die nichtnormalisierten Projektionen auf W_i^0 und W_i^1.
Angenommen eine Folge von Messungen für Spalten $i_1, \ldots, i_d$ mit einem Wert $d \leq \log_{1/h} row(M')$ und mit immer korrektem Ausgang bestimmt eine Zeile. Zu jeder Zeile gibt es eine solche (leicht eindeutig zu machende) Folge von Spalten, welche unabhängig von den Messungsergebnissen ist.
Anstelle des Effektes der Messungen analysieren wir den Effekt von unitären Transformationen, welche so arbeiten: Sei $U_i|\phi_i^0, a\rangle = |\phi_i^0, a\rangle$ und $U_i|\phi_i^1, a\rangle = |\phi_i^1, a \oplus e_i\rangle$ für den Vektor e_i mit genau einer 1 an Position i und Nullen sonst und für alle Booleschen a. Weiterhin betrachten wir „ideale“ Transformationen $U'_{i,x}$, welche $|\phi, a\rangle$ auf $|\phi, a \oplus e_i \cdot x_i\rangle$ abbilden (aber von der unbekannten Zeile x abhängen).
Während nun die unitären Transformationen U_i kein Ergebnis ausgeben, erhalten wir doch folgendes: wenn der durch die unitären Transformationen eingeführte Fehler klein ist, so auch der Fehler durch die Messungen, und mit kleiner Wahrscheinlichkeit versagt eine Messung. Wie schließen also, daß mit hoher Wahrscheinlichkeit alle adaptiven Messungen Erfolg haben und die richtige Zeile bestimmt wird.
Wir beschränken nun den durch Anwendung von U_i eingeführten Fehler. O.B.d.A. sei $x_i = 1$. Dann ist $U_i|\phi_i^1, a\rangle = U'_{i,x}|\phi_i^1, a\rangle$. Wir bemerken folgendes.

$$\begin{aligned} ||U_i|\phi, 0^l, a\rangle - U'_{i,x}|\phi, 0^l, a\rangle|| &\leq ||U_i(|\phi_i^0, a\rangle + |\phi_i^1, a\rangle) \\ &- U'_{i,x}(|\phi_i^0, a\rangle + |\phi_i^1, a\rangle)|| \\ &= ||U_i(|\phi_i^0, a\rangle) - U'_{i,x}(|\phi_i^0, a\rangle)|| \\ &\leq ||U_i|\phi_i^0, a\rangle|| + ||U'_{i,x}|\phi_i^0, a\rangle|| \\ &\leq 2\sqrt{\epsilon}, \end{aligned}$$

weil $|| \, |\phi_i^0\rangle||^2 \leq \epsilon$ die Wahrscheinlichkeit eines Fehlers ist.
Nun analysieren wir den Fehler einer Folge von unitären Transformationen.

$$\begin{aligned}
& ||U_1 \cdots U_d|\phi, 0^l, 0^d\rangle - U'_{1,x} \cdots U'_{d,x}|\phi, 0^l, 0^d\rangle|| \\
\leq \; & ||U_1 \cdots U_{d-1}U_d|\phi, 0^l, 0^d\rangle - U_1 \cdots U_{d-1}U'_{d,x}|\phi, 0^l, 0^d\rangle|| \\
+ \; & ||U_1 \cdots U_{d-1}U'_{d,x}|\phi, 0^l, 0^d\rangle - U_1 \cdots U'_{d-1,x}U'_{d,x}|\phi, 0^l, 0^d\rangle|| \\
+ \; & \cdots \\
+ \; & ||U_1U'_{2,x} \cdots U'_{d-1,x}U'_{d,x}|\phi, 0^l, 0^d\rangle - U'_{1,x} \cdots U'_{d-1,x}U'_{d,x}|\phi, 0^l, 0^d\rangle||
\end{aligned}$$

aufgrund der Dreiecksungleichung. Aber da für jeden Term

$$||U_1 \cdots U_tU'_{t+1,x} \cdots U'_{d,x}|\phi, 0^l, 0^d\rangle - U_1 \cdots U'_{t,x}U'_{t+1,x} \cdots U'_{d,x}|\phi, 0^l, 0^d\rangle||$$

die unitären Transformationen $U_1 \cdots U_{t-1}$ fallengelassen werden können und die Transformation $U'_{t+1,x} \cdots U'_{d,x}$ das $|\phi, 0^l, 0^d\rangle$ auf $|\phi, 0^l, a\rangle$ für ein a abbildet, schließen wir von obiger Beobachtung, daß

$$||U_1 \cdots U_d|\phi, 0^l, 0^d\rangle - U'_{1,x} \cdots U'_{d,x}|\phi, 0^l, 0^d\rangle|| \leq d \cdot 2\sqrt{\epsilon}.$$

Sei ρ_1 die Dichtematrix des puren Zustands $|\psi_1\rangle = U_1 \cdots U_d|\phi, 0^l, 0^d\rangle$ und ρ_2 die Dichtematrix des Zustands $|\psi_2\rangle = U'_{1,x} \cdots U'_{d,x}|\phi, 0^l, 0^d\rangle$. Fakt 5.6 ergibt $||\rho_1 - \rho_2||_1 \leq 2|||\psi_1\rangle - |\psi_2\rangle||$ und nach Fakt 5.5 ist die meßbare Distanz zwischen den Zuständen höchstens $4d\sqrt{\epsilon}$. Aufgrund der Wahl von ϵ ist die Wahrscheinlichkeit eines Fehlers $< 1/2$. □
Zuerst wenden wir die neue Methode auf die Index Funktion an.

Korollar 5.2 *$bound(IX_n) = \Omega(n/\log n)$.*

Beweis: Wir nehmen alle Zeilen der Kommunikationsmatrix von IX_n in die Menge M' auf. Wenn wir die Werte einiger Spalten fixieren, wird ein Teil des Wortes x der Zeile fixiert, alle anderen Positionen sind frei. So gilt $h = 1/2$ und die Schranke folgt. □
Wir vergleichen unsere Methoden für untere Schranken.

Korollar 5.3 *Für alle f: $bound(f) = \Omega(VC(f)/\log VC(f))$.*

Beweis: Wie wir gesehen haben, ist die VC-Dimension von f nichts anderes als die Größe der größten Instanz der Index Funktion, deren Matrix in die Kommunikationsmatrix von f eingebettet werden kann. Wenn wir uns auf diese Submatrix einschränken, gibt das vorige Korollar die Schranke. □.
Wie bereits gesagt, ist die VC-Dimension der GT_n Funktion nur 1, aber die neue Methode gibt ein besseres Ergebnis.

Korollar 5.4 *$Q^{(1)}(GT_n) = \Omega(n/\log n)$.*

Beweis: Für M' nehmen wir alle Zeilen der Kommunikationsmatrix von GT_n. Wenn einige Spalten fixiert sind, enthält die restliche Matrix immer noch eine obere Dreiecksmatrix, was zu $h = 1/2$ führt. □

5.5 Anwendungen

5.5.1 Überblick

In diesem Abschnitt untersuchen wir Anwendungen der Quanten Kommunikationskomplexität auf Quanten Automaten und Quanten Formeln. Die unteren Schranken aus der Kommunikationstheorie, welche hier angewendet werden, sind die Schranken für Einweg Kommunikation aus dem vorigen Abschnitt.

5.5.2 Quanten Einweg Automaten

Theorem 5.11 *Für alle Sprachen L über Σ^* mit einem minimalen dfa der Größe D gilt:*
Jeder exakte qcfa für L hat mindestens Größe D.
Jeder qcfa mit Fehler ϵ für L hat mindestens Größe $2^{(1-H(\epsilon))VC(L)}$.
Jeder Las Vegas qcfa mit Erfolgswahrscheinlichkeit $1-\epsilon$ für L hat eine Größe von mindestens $D^{1-\epsilon}$.

Beweis: Wie die Argumente im Beweis zu Theorem 5.8 zeigen, ist die von Neumann Entropie der Nachrichten in einem Protokoll mit einer Kommunikationsmatrix mit R verschiedenen Zeilen und bei Erfolgswahrscheinlichkeit $1-\epsilon$ mindestens $(1-\epsilon)\log R$ und der Hilbertraum der Nachrichten hat eine Dimension von mindestens $R^{1-\epsilon}$. Da ein Quanten Protokoll für das uniforme Kommunikationsproblem (siehe Abschnitt 3.3.1) einen Automaten der Größe q mittels Nachrichten in einem q-dimensionalen Hilbertraum simulieren kann, gilt $q > R^{1-\epsilon}$. R ist gleich dem Nerodeindex und wir erhalten die Resultate für exakte und Las Vegas Automaten.
Im Falle des beschränkten Fehlers simuliert wieder ein Protokoll für das uniforme Kommunikationsproblem einen Automaten mit q Zuständen mit q-dimensionalen Nachrichten. Die von Neumann Entropie des Nachrichtenraums muß mindestens $(1-H(\epsilon))VC(L)$ sein, ist aber höchstens $\log q$ und die untere Schranke folgt. □

Korollar 5.5 *Die Größe eines qcfa für IX_n (als Sprache) ist $2^{\Omega(n)}$.*
Die Größe eines qcfa für $DISJ_n$ (als Sprache) ist $2^{\Omega(n)}$.
Die Größe eines qcfa für GT_n (als Sprache) ist $2^{\Omega(n/\log n)}$.

5.5.3 Quanten Formeln

In der Arbeit [RV99] werden pure Quanten Formeln betrachtet (also Formeln, welche nur Eingaben und Boolesche Konstanten lesen dürfen, und auch nur ein Ausgabe Qubit messen). Folgendes ist das Resultat.

Fakt 5.22 *Jede pure Quanten Formel, welche eine Funktion f mit beschränktem Fehler berechnet, hat die Länge*

$$\Omega\left(\sum_i D^{(1)}(f_i)/\log D^{(1)}(f_i)\right),$$

für die Nečiporuk Funktion $\sum_i D^{(1)}(f_i)$, siehe Definition 3.2.

Nun wissen wir aus Abschnitt 3.3.3, daß es eine Boolesche Funktion MP mit $O(n^2)$ Eingaben gibt (die Matrix Produkt Funktion), so daß probabilistische Formeln für MP lineare Größe haben können, während die Nečiporuk Schranke groß ist, d.h. es gibt eine Partition der Eingaben, so daß die Nečiporuk Funktion für die Partition und die MP Funktion $\Omega(n^3)$ ist (Theoreme 3.7 und 3.8). Es ergibt sich folgendes.

Korollar 5.6 *Es gibt eine Boolesche Funktion mit N Eingaben, die von fairen probabilistischen Formeln der Länge $O(N)$ berechnet werden kann, für deren Berechnung pure Quanten Formeln aber eine Länge von $\Omega(N^{3/2}/\log N)$ benötigen.*

Eine jede faire probabilistische Formel kann direkt von einer verallgemeinerten Quanten Formel simuliert werden. Also benötigt man eine andere untere Schranke für verallgemeinerte Quanten Formeln. Wir beschreiben zuerst eine untere Schranke über Quanten Einweg Kommunikation und zeigen dann mittels Lemma 5.4 die Gültigkeit der VC-Nečiporuk Funktion als untere Schranke. Somit erhalten wir für Quanten Formeln und für probabilistische Formeln dieselbe kombinatorische Technik für untere Schranken. Es gilt also nach Theorem 3.6, daß der maximale Unterschied zwischen den Längen von deterministischen Formeln und Quanten Formeln mit beschränktem Fehler, der mit der Nečiporuk Methode beweisbar ist, höchstens $O(\sqrt{n})$ ist.
Wir halten zuerst noch folgendes Korollar fest, das folgt, weil pure Quanten Formeln probabilistische Formeln, die ihre Zufallseingaben je nur einmal lesen, effizient simulieren können.

Korollar 5.7 *Die (konventionelle) Nečiporuk Funktion geteilt durch $\log n$ ist eine asymptotische untere Schranke für die Länge fairer probabilistischer Formeln, welche jede Zufallseingabe nur einmal lesen.*

Beweis: Wir müssen nur zeigen, daß pure Quanten Formeln die speziellen probabilistischen Formeln effizient simulieren können. Wir betrachten fan-in 2. Für jede Zufallseingabe verwenden wir zwei Qubits im Zustand $|00\rangle$. Diese werden durch eine Hadamard Transformation in den Zustand $|\Phi^+\rangle$ versetzt. Eines der Qubits wird nie wieder benutzt, das andere Qubit hat die Dichtematrix der Verteilung eines Zufallsbits. Dann kann die probabilistische Formel simuliert werden. Bei der Simulation eines Gatters werden

unitäre Transformationen auf drei Qubits verwendet. Diese erhalten die beiden Eingaben eines Gatters sowie ein leeres Qubit und geben auf einem Qubit die Ausgabe des Gatters, auf den anderen die beiden Eingabebits aus. So sind sie unitär. Nach [BBC$^+$95] kann jede solche Transformation aus $O(1)$ unitären Transformationen auf 2 Qubits zusammengesetzt werden. □
Wir brauchen noch folgendes Resultat [AKN98].

Fakt 5.23 *Wenn die Dichtematrix zweier Qubits in einem Schaltkreis nicht das Tensorprodukt der einzelnen Dichtematrizen ist, dann gibt es ein Gatter, so daß beide Qubits auf einem Pfad von dem Gatter erreicht werden.*

Also sind in einer Formel die Eingaben eines Gatters niemals verschränkt.
Die erste untere Schranke wird mit Kommunikation formuliert.

Theorem 5.12 *Sei f eine Boolesche Funktion auf n Eingaben und $y_1 \dots y_k$ eine disjunkte Partition der Eingabevariablen in k Blöcke. Spieler B kenne die Eingaben in y_i und Spieler A alle anderen Eingaben. Die Quanten Einweg Kommunikationskomplexität von f (mit beschränktem Fehler) unter dieser Aufteilung der Eingaben heiße $Q^{(1)}(f_i)$.*
Jede verallgemeinerte Quanten Formel für f mit beschränktem Fehler hat eine Länge von

$$\Omega\left(\sum_i \frac{Q^{(1)}(f_i)}{\log Q^{(1)}(f_i)}\right).$$

Beweis: Für eine gegebene Partition der Eingaben zeigen wir, wie eine verallgemeinerte Quanten Formel F in den k Kommunikationsspielen simuliert werden kann, so daß die Quanten Einweg Kommunikation in Spiel i beschränkt ist durch eine Funktion in der Anzahl der Blätter in dem Subbaum F_i von F, der genau die Variablen, die B gehören, als Blätter hat, und die Wurzel der Formel als Wurzel und alle Pfade von solchen Blättern zur Wurzel enthält. F ist ein Baum von fan-in 2 fan-out 1 Superoperator Gattern (Superoperatoren sind nicht unbedingt reversibel).
Als erstes legt A für alle zusätzlichen Eingaben einen puren Zustand fest. Nun können alle Vorkommen dieser Variablen als einzelne Qubits ohne Zusammenhang behandelt werden.
In allen Kommunikationsspielen versucht Spieler B, die Formel so weit wie möglich auszuwerten ohne die Hilfe von A. Durch ein Argument wie in anderen Nečiporuk Methoden ([BS90],[RV99]) reichen wenige Qubits Kommunikation aus, um einen maximalen Pfad in der Subformel auszuwerten, der folgende Eigenschaft hat: alle Gatter auf dem Pfad haben eine Eingabe von A und eine Eingabe vom Vorgänger auf dem Pfad, außer dem ersten Gatter, daß eine Eingabe bei A und eine bei B hat (welche bereits bekannt ist). Mit Standard Argumenten ist die Anzahl solcher Pfade eine untere Schranke der Anzahl der Blätter aus der Subformel, siehe Abschnitt 3.3.3.

Wir betrachten also einen Pfad $g_1, \ldots, g_m$ in F, wobei g_1 eine Eingabe/Gatter von A und eine Eingabe/Gatter von B als Vorgänger hat und alle Gatter $g_2, \ldots, g_m$ das jeweils vorige Gatter und eine Eingabe/Gatter von A als Vorgänger haben. Die Dichtematrix von Bs Eingabe zu g_1 sei ρ und die Dichtematrix der anderen m Eingaben zum Pfad sei σ. Der Schaltkreis, der σ erzeugt, arbeitet auf anderen Qubits als derjenige, der ρ erzeugt (alle Eingaben der Formel sind nach der Initialisierung als unabhängig zu betrachten, da es keine Korrelationen zwischen ihnen gibt, sie sind fix).
Somit ist die Dichtematrix aller Eingaben zu dem Pfad ein Tensorprodukt $\rho \otimes \sigma$, siehe Fakt 5.23. Der Pfad bildet $\rho \otimes \sigma$ mit einem Superoperator T auf eine Dichtematrix μ auf einem Qubit ab. Nun kann A bestimmen, welcher Superoperator dies für ein fixiertes σ ist (für jede Boolesche Eingabe für A gibt es ein solches σ und einen dazugehörigen Superoperator).
A kann σ allein berechnen. B kennt bereits ρ. B möchte den Zustand μ wissen, der durch die Anwendung eines A bekannten Superoperators bestimmt ist. A präpariert also das Programm für ein programmierbares Quanten Superoperator Gatter wie in Abschnitt 5.3.2 beschrieben. Da σ ein gemischter Zustand ist, bestimmt A tatsächlich sogar einen gemischten Zustand von Programmen, aber das macht nichts, da alles linear ist. A sendet das Programm zu B, der es anwendet. Die Programmlänge bei Erfolgswahrscheinlichkeit $1 - 1/k$ ist $O(\log k)$ Qubits. B berechnet den Zustand μ mit dem programmierbaren Gatter mit Wahrscheinlichkeit $1 - 1/k$ korrekt und fährt fort bis zum nächsten Pfad. Hat das Gatter versagt, gibt B auf.
Insgesamt ist die Anzahl der Blätter von unten beschränkt durch die Summe der Kommunikationen (bzw. Pfade) multipliziert mit $O(\log k)$, was wiederum von unten beschränkt ist durch $O(\log k)$ multipliziert mit der Quanten Nečiporuk Funktion.
Der Fehler der Formel ist unverändert, wenn alle Aufrufe des programmierbaren Gatters erfolgreich sind. Da es in jedem der Spiele nur höchstens $size_i$ viele solche Aufrufe gibt (wenn es $size_i$ viele Blätter gibt), reicht $k = poly(size_i)$ für eine Fehlerwahrscheinlichkeit $1/poly(size_i)$ im Las Vegas Sinn, zusätzlich zum Fehler, den die Formel selbst macht. Also ist die Kommunikation durch $Q^{(1)}(f_i) \leq O(size_i \log size_i)$ beschränkt und somit ist $size_i \geq \Omega(Q^{(1)}(f_i)/Q^{(1)}(f_i))$. Summation über alle i ergibt die untere Schranke. □
Die obige Konstruktion verliert einen logarithmischen Faktor, aber in der kombinatorischen Schranke, die wir anwenden, können wir dies verbessern.

Theorem 5.13 *Die VC-Nečiporuk Funktion ist asymptotisch eine untere Schranke für die Länge verallgemeinerter Quanten Formeln mit beschränktem Fehler.*
Die Nečiporuk Funktion ist asymptotisch eine untere Schranke für die Länge verallgemeinerter Quanten Las Vegas Formeln.

Beweis: Die Relation, die in Theorem 5.12 bewiesen wurde, gibt uns ein Ergebnis mit Verlust eines logarithmischen Faktors. Betrachten wir das Protokoll näher. Der logarithmische Faktor geht verloren, weil A sicherstellen muß, daß B bei jedem Aufruf des programmierbaren Quanten Gatters Erfolg mit hoher Wahrscheinlichkeit hat. Das programmierbare Gatter versucht bekanntlich, eine unitäre Operation auszuführen, was mit konstanter Wahrscheinlichkeit gelingt, gelingt es nicht, wird ein neuer Versuch gemacht. Im Durchschnitt braucht das Gatter nur ein konstant langes Anfangsstück des Programms bis zu einem Erfolg. Könnte also A eine Nachricht von B erhalten, ob die Anwendung des Gatters erfolgreich war, oder ob noch eine Kopie des Programms notwendig ist, so könnte A das Programm genau so oft wie nötig senden und die Gesamtkommunikation auf $O(size_i)$ gesenkt werden, da durchschnittlich pro Pfad nur $O(1)$ Qubits gebraucht würden, und insgesamt mit hoher Wahrscheinlichkeit eben nur $O(size_i)$. Aber A braucht eine Rückmeldung von B. Glücklicherweise braucht B dabei keine Information über seine Eingabe zu geben.
Wir betrachten also das Nichtstandard Kommunikationsmodell aus Lemma 5.4 in dem B zu A sprechen darf, aber ohne irgendeine Quanten Information über seine Eingabe an A zu geben.
Wenn A bei einer Anwendung des programmierbaren Gatters alle Programmqubits an B sendet, für ein mit konstanter Wahrscheinlichkeit erfolgreiches Programm mit $O(1)$ Qubits, so wendet B das Gatter an, und teilt A mit, ob er erfolgreich war oder nicht. Es ist klar, daß A so keine Information über Bs Eingabe erhalten kann. Die Gesamtkommunikation sinkt auf $O(size_i)$.
Nach Lemma 5.4 ergibt sich die untere Schranke für beschränkten Fehler und Las Vegas. □

Korollar 5.8 *Eine verallgemeinerte Quanten Formel, welche ISA mit beschränktem Fehler berechnet hat eine Länge von* $\Omega(n^2/\log n)$.

Durch Betrachtung der Matrixmultiplikation MP erhalten wir

Korollar 5.9 *Es gibt eine Funktion, welche sowohl von einer verallgemeinerten Quanten Formel mit beschränktem Fehler als auch von einer fairen probabilistischen Formel mit beschränktem Fehler bei einer Länge von* $O(N)$ *berechnet werden kann. Jede verallgemeinerte Quanten Las Vegas Formel braucht hingegen für dieses Problem eine Länge von* $\Omega(N^{3/2})$. *Es gibt also einen Unterschied von* $\Omega(N^{1/2})$ *zwischen Las Vegas Formellänge und der Länge von Formeln mit beschränktem Fehler.*

Da die VC-Nečiporuk Funktion eine untere Schranke für verallgemeinerte Quanten Formeln ist, impliziert Theorem 3.6, daß der maximale Unterschied zwischen deterministischer Formellänge und der Länge von Quantenformeln mit beschränktem Fehler, der mit der konventionellen Nečiporuk Methode

nachweisbar ist, $O(\sqrt{n})$ für Eingabelänge n ist. Ein solcher Unterschied besteht bereits zwischen fairen probabilistischen Formeln mit beschränktem einseitigem Fehler und Quanten Las Vegas Formeln.

Kapitel 6

Zusammenfassung

Wir haben Interaktion in der Kommunikationskomplexität untersucht und dabei die drei Modi probabilistische, (beschränkt) nichtdeterministische und quantenmechanische Kommunikation betrachtet. Bei allen drei Modi haben wir herausgefunden, daß Interaktion für Effizienz oft unerläßlich ist, im nichtdeterministischen Fall gibt es eine Abhängigkeit zwischen dem Einfluß der Interaktion und der erlaubten Anzahl der nichtdeterministischen Ratebits.
Abgesehen von dem erreichten besseren Verständnis des Kommunikationsmodells haben wir verschiedene Anwendungen auf andere Berechnungsmodelle beschrieben, bei denen untere Schranken der Kommunikation zu unteren Schranken für andere Resourcen in diesen Modellen geführt haben.
Ein Beispiel eines kommunikations- und interaktionsbeschränkten Modells sind endliche Automaten, welche wir in allen drei Modi untersucht haben. Ein weiteres Beispiel sind Formeln, für die wir eine Verbindung zwischen Einweg Kommunikation und Formellänge herstellen konnten. Diese Verbindung führte zu unteren Schranken für probabilistische, nichtdeterministische und Quanten Formeln. Dabei sind die unteren Schranken für Quanten Formeln und probabilistische Formeln im wesentlichen gleich. Für monotone Schaltkreise haben wir gezeigt, wie nichtdeterministisches Raten die Tiefe drastisch reduzieren kann, und wie eine geringfügige Einschränkung der nichtdeterministischen Ratebits zu einer Tiefenhierarchie führt.
Insgesamt läßt sich feststellen, daß die Schwäche interaktionsbeschränkter Kommunikation mathematisch nachvollziehbar ist. Außerdem scheint ein solches Verhalten in der Welt einfacher Berechnungsmodelle häufig aufzutreten. Oder anders gesagt, viele Berechnungsmodelle sind deshalb einfacher zu verstehen, weil sie durch interaktionsbeschränkte Kommunikation analysierbar sind.

Literaturverzeichnis

[Ab93] F. Ablayev. Lower Bounds for One-Way Probabilistic Communication Complexity. *20th Int. Colloquium on Automata, Languages and Programming*, pp.241–252, 1993.

[AKN98] D. Aharonov, A. Kitaev, N. Nisan. Quantum Circuits with Mixed states. *Proc. 30th ACM Symp. on Theory of Comp.*, pp.20–30, 1998.

[AUY83] A. Aho, J. Ullman, M. Yannakakis. On notions of information transfer in VLSI circuits. *Proc. 15th ACM Symp. on Theory of Computing*, pp.133–139, 1983.

[Aj99] M. Ajtai. Determinism versus Non-Determinism for Linear Time RAMs. *Proc. 31st ACM Symp. on Theory of Computing*, pp.632–641, 1999.

[A96] A. Ambainis. The complexity of probabilistic versus deterministic finite automata. *7th Int. Symp. on Algorithms and Computation*, pp.233–237, 1996.

[AF98] A. Ambainis, R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. *Proc. 39th IEEE Symp. Foundations of Computer Science*, pp.332–341, 1998.

[ANTV99] A. Ambainis, A. Nayak, A. Ta-Shma, U. Vazirani. Dense quantum coding and a Lower Bound for 1-way quantum finite automata. *Proc. 31th ACM Symp. on Theory of Computing*, pp.376–383, 1999.

[BNS92] L. Babai, N. Nisan, M. Szegedy. Multiparty protocols, Pseudorandom Generators for Logspace, and Time-Space Trade-offs. *Journ. of Computer and System Sciences*, vol.45, pp.204–232, 1992.

[BBC+95] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, H. Weinfurter. Elementary gates for quantum computation. *Phys. Review A*, vol.52, pp.3457–3467, 1995.

[BCMW98] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf. Quantum Lower Bounds by Polynomials. *Proc. 39th IEEE Symp. Foundations of Computer Science*, pp.352–361, 1998.

[BL92] P. Beame, J. Lawry. Randomized versus Nondeterministic Communication Complexity. *Proc. 24th ACM Symp. on Theory of Computing*, pp.188–199, 1992.

[BW92] C.H. Bennett, S.J. Wiesner. Communication via One- and Two-Particle Operators on Einstein-Podolsky-Rosen states. *Phys. Review Lett.*, vol.69, pp.2881–2884, 1992.

[BV97] E. Bernstein, U. Vazirani. Quantum Complexity Theory. *SIAM Journal Computing*, vol.26, pp.1411–1473, 1997.

[Bl87] R.E. Blahut. Principles and Practice of Information Theory. Addison Wesley, 1987.

[B85] R.B. Boppana. Amplification of probabilistic Boolean formulas. *26th IEEE Symp. Foundations of Computer Science*, pp.20–29, 1985.

[BS90] R.B. Boppana, M. Sipser. The Complexity of Finite functions. *Handbook of Theoretical Computer Science A*. Elsevier, 1990.

[BBHT96] M. Boyer, G. Brassard, P. Hoyer, A. Tapp. Tight bounds on quantum searching. *4th Workshop on Physics and Comp.*, pp.36–43, 1996.

[BCW98] H. Buhrman, R. Cleve, A. Wigderson. Quantum vs. Classical Communication and Computation. *Proc. 30th ACM Symp. on Theory of Computing*, pp.63–68, 1998.

[BCWZ99] H. Buhrman, R. Cleve, R. de Wolf, C. Zalka. Bounds for Small-Error and Zero-Error Quantum Algorithms. *Proc. 40th IEEE Symp. Foundations of Computer Science*, pp.358–368, 1999.

[BW99] H. Buhrman, R. de Wolf. Communication Complexity Lower Bounds by Polynomials. quant-ph/991001 [unter `www.arXiv.org`], 1999.

[CC97] L. Cai, J. Chen. On the amount of nondeterminism and the power of verifying. *SIAM Journal on Computing*, vol.26, pp.733–750, 1997.

[CA96] N. Cerf, C. Adami. Quantum information theory of entanglement and measurement. *Proc. of Physics and Computation PhysComp*, pp.65–71, 1996.

[CY91] J. Chen, C.K. Yap. Reversal Complexity. *SIAM Journal on Computing*, vol.20, pp.622–638, 1991.

[CB97] R. Cleve, H. Buhrman. Substituting Quantum Entanglement for Communication. *Physical Review A*, vol.56, pp.1201–1204, 1997.

[CDNT97] R. Cleve, W. van Dam, M. Nielsen, A. Tapp. Quantum Entanglement and the Communication Complexity of the Inner Product Function. *Proc. 1st NASA Int. Conf. on Quantum Computing and Quantum Communications*, 1998.

[Da96] D. Damanik. Endliche Automaten mit eingeschränkter 2-Wege Bewegung. Diplomarbeit, Frankfurt 1996.

[D85] D. Deutsch. Quantum theory, the Church-Turing principle, and the universal quantum computer. *Proceedings of the Royal Society London A*, vol.400, pp.97–117, 1985. Auch unter: `www.qubit.org`.

[D89] D. Deutsch. Quantum Computational Networks. *Proceedings of the Royal Society London A*, vol.425, pp.73–90, 1989.

[DHS96] M. Dietzfelbinger, J. Hromkovič, G. Schnitger. A comparison of two lower bound methods for communication complexity. *Theoretical Computer Science*, vol.168, pp.39–51, 1996.

[Di47] P.A.M. Dirac. The principles of quantum mechanics. *Oxford University Press*, 1947.

[DZ97] M. Dubiner, U. Zwick. Amplification by Read-Once Formulae. *SIAM Journal on Computing*, vol.26, pp.15–38, 1997.

[DGS87] P. Duris, Z. Galil, G. Schnitger. Lower Bounds on Communication Complexity. *Information and Computation*, vol.73, pp.1–22, 1987.

[DHRS97] P. Duriš, J. Hromkovič, J.D.P. Rolim, G. Schnitger. Las Vegas versus Determinism for One-Way Communication Complexity, Finite Automata, and Polynomial-time Computations. *14th Symp. on Theoretical Aspects of Computer Science*, pp.117–128, 1997.

[DS89] C. Dwork, L. Stockmeyer. On the power of two-way probabilistic finite state machines. *30th IEEE Symp. Foundations of Computer Science*, pp.480–485, 1989.

[EPR35] A. Einstein, B. Podolsky, N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, vol.47, pp.777–780, 1935.

[F82] R.P. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, vol.21, pp.467–488, 1982.

[Fr81] R. Freivalds. Probabilistic two-way machines. *Proc. Symp. Mathematical Foundations of Computer Science*, pp.33–45, 1981.

[Fü87] M. Fürer. The Power of Randomness for Communication Complexity. *19th ACM Symp. on Theory of Computing*, pp.178–181, 1987.

[GLM96] J. Goldsmith, M.A. Levy, M. Mundhenk. Limited Nondeterminism. *SIGACT News*, vol.27(2), pp.20–29, 1996.

[GKW90] J. Goldstine, C.M.R. Kintala, D. Wotschke. On Measuring Nondeterminism in Regular Languages. *Information and Computation*, vol.86, pp.179–194, 1990.

[GLW92] J. Goldstine, H. Leung, D. Wotschke. On the Relation between Ambiguity and Nondeterminism in Finite Automata. *Information and Computation*, vol.100, pp. 261–270, 1992.

[G90] R.M. Gray. Entropy and Information Theory. Springer, 1990.

[G96] L.K. Grover. A fast quantum mechanical algorithm for database search. *28th ACM Symposium on the Theory of Computing*, pp.212–219, 1996.

[Gr99] J. Gruska. Quantum Computing. Wiley Interscience. 1999.

[H35] P. Hall. On Representatives of Subsets. *J. London Math. Soc.*, vol.10, pp. 26–30, 1935.

[HR93] B. Halstenberg, R. Reischuk. Different Modes of Communication. *SIAM Journal Comput.*, vol.22, pp.913–934, 1993.

[H73] A.S. Holevo. Some estimates on the information transmitted by quantum communication channels. *Problems of Information Transmission*, vol.9, pp.177–183, 1973.

[HU79] J.E. Hopcroft, J.D. Ullmann. Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 1979.

[Hr91] J. Hromkovič. Reversals-Space-Parallelism Tradeoffs for Language Recognition. *Math. Slovaca*, vol.2, pp.121–136, 1991.

[Hr97] J. Hromkovič. Communication Complexity and Parallel Computing. Springer, 1997.

[HKKSS00] J. Hromkovič, J. Karhumäki, H. Klauck, G. Schnitger, S. Seibert. Measures of Nondeterminism in Finite Automata. *27th Int. Colloquium on Automata, Languages and Programming*, pp.199–210, 2000.

[HrSa00] J. Hromkovič, M. Sauerhoff. Tradeoffs between Nondeterminism and Complexity for Communication Protocols and Branching Programs. *17th Symp. on Theor. Aspects of Comp. Science*, pp.145–156, 2000.

[HrS96] J. Hromkovič, G. Schnitger. Nondeterministic Communication with a Limited Number of Advice Bits. *Proc. 28th ACM Symp. on Theory of Computing*, pp.451–560, 1996.

[HrS99] J. Hromkovič, G. Schnitger. On the Power of Las Vegas II: Two-Way Finite Automata. *26th Int. Colloquium on Automata, Languages and Programming*, pp.433–442, 1999.

[KS92] B. Kalyanasundaram, G. Schnitger. The Probabilistic Communication Complexity of Set Intersection. *SIAM Journ. Discrete Math.,* vol.5, pp.545–557, 1992.

[K88] M. Karchmer. Communication Complexity: A New Approach to circuit Depth. Dissertation, 1988.

[KNSW94] M. Karchmer, I. Newman, M. Saks, A. Wigderson. Nondeterministic Communication Complexity with few Witnesses. *Journ. of Computer and System Sciences*, vol.49, pp. 247–257, 1994.

[KW90] M. Karchmer, A. Wigderson. Monotone Circuits for Connectivity Require Superlogarithmic Depth. *SIAM Journ. Discrete Math.*, vol.3, pp.255–265, 1990.

[KW80] C.M.R. Kintala, D. Wotschke. Amounts of nondeterminism in finite automata. *Acta Informatica*, vol.13, pp.199–204, 1980.

[Kl97] H. Klauck. On the Size of Probabilistic Formulae. *8th Int. Symp. on Algorithms and Computation*, pp.243–252, 1997.

[Kl98] H. Klauck. Lower bounds for computation with limited nondeterminism. *13th IEEE Conference on Computational Complexity*, pp.141–153, 1998.

[Kl00a] H. Klauck. On Quantum and Probabilistic Communication: Las Vegas and One-Way Protocols. *32th ACM Symp. on Theory of Computing*, pp.644–651, 2000.

[Kl00b] H. Klauck. Quantum Communication Complexity. *Workshop on Boolean Functions and Applications at ICALP*, pp.241–252, 2000.

[KPY84] M. Klawe, W. Paul, N. Pippenger, M. Yannakakis. On Monotone Formulae with Restricted Depth. *Proc. 16th ACM Symp. on Theory of Computing*, pp.480–487, 1984.

[KW97] A. Kondacs, J. Watrous. On the power of quantum finite state automata. *38th IEEE Symp. on Foundations Computer Science*, pp.66–75, 1997.

[KN97] E.Kushilevitz, N.Nisan. Communication Complexity. Cambridge University Press, 1997.

[K95] I. Kremer. Quantum Communication. Master's thesis (Hebrew University), 1995.

[KNR95] I. Kremer, N. Nisan, D. Ron. On Randomized One-Round Communication Complexity. *27th ACM Symp. on Theory of Computing*, pp.596–605, 1995.

[LC98] H. Lo, H. Chau. Why quantum bit commitment and ideal quantum coin tossing are impossible. *Physica D*, vol.120, pp.177–187, 1998.

[L90] L. Lovasz. Communication Complexity: A Survey. in: *Paths, Flows, and VLSI Layout*, Springer 1990.

[M97] D. Mayers. Unconditionally secure quantum bit commitment is impossible. *Phys. Rev. Lett.*, vol.78, pp.3414–3417, 1997.

[MO00] A.J. Menezes, P.C. van Oorschot. Coding Theory and Cryptology. In *Handbook of Discrete and Combinatorial Mathematics*, CRC Press, pp.889–954, 2000.

[MF71] A.R. Meyer, J. Fischer. Economy of description by automata, grammars, and formal systems, *Proc. 12th Annual Symp. on Switching and Automata Theory*, pp.188–191, 1971.

[MNSW95] P.B. Miltersen, N. Nisan, S. Safra, A. Wigderson. Data Structures and Asymmetric Communication Complexity. *Proc. 27th ACM Symp. on Theory of Computing*, pp.103–111, 1995.

[MC97] C. Moore, J. Crutchfield. Quantum automata and quantum grammars. Santa Fe Institute Working paper 97-07-062, 1997.

[MR95] R. Motwani, P. Raghavan. Randomized Algorithms. Cambridge University Press, 1995.

[Na99] A. Nayak. Optimal Lower Bounds for Quantum Automata and Random Access Codes. *40th IEEE Symp. Foundations of Computer Science*, pp.369–377, 1999.

[N66] E.I. Nečiporuk. A Boolean Function. *Sov. Math. Dokl.*, vol.7, pp.999–1000, 1966.

[vN32] J. von Neumann. Mathematische Grundlagen der Quantenmechanik. Springer, 1932.

[Ne91] I. Newman. Private vs. Common Random Bits in Communication Complexity. *Information Processing Letters*, vol.39, pp.67–71, 1991.

[NC97] M.A. Nielsen, I. Chuang. Programmable quantum gate arrays. *Phys. Rev. Lett.*, pp.321–324, 1997.

[NW93] N. Nisan, A. Wigderson. Rounds in communication complexity revisited. *SIAM Journal on Computing*, vol.22, pp.211–219, 1993.

[PS84] C. Papadimitriou, M. Sipser. Communication Complexity. *Journ. of Computer and System Sciences*, vol.28, pp.260–269, 1984.

[PST83] W. Paul, N. Pippenger, E. Szemerédi, W. Trotter. On Determinism Versus Non-Determinism and Related Problems. *24th Symp. Foundations Computer Science*, pp.429–438, 1983.

[PRV99] S.J. Ponzio, J. Radhakrishnan, S. Venkatesh. The Communication complexity of pointer chasing: applications of entropy and sampling. *Proc. 31st ACM Symp. on Theory of Computing*, pp.602–611, 1999.

[Pr98] J. Preskill. Lecture notes on quantum information and quantum computation. California Institute of Technology. Web address: `www.theory.caltech.edu/people/preskill/ph229`, 1998.

[R63] M. Rabin. Probabilistic automata. *Information and Control*, vol.6, pp.230–245, 1963.

[R98] R. Raz. A Parallel Repetition Theorem. *SIAM Journal on Computing*, vol.27, pp.763–803, 1998.

[R99] R. Raz. Exponential Separation of Quantum and Classical Communication Complexity. *Proc. 31th ACM Symp. on Theory of Computing*, pp.358–367, 1999.

[RM97] R. Raz, P. McKenzie. Separation of the Monotone NC Hierarchy. *38th IEEE Symp. Foundations of Computer Science*, pp.234–243, 1997.

[RW89] R. Raz, A. Wigderson, Probabilistic Communication Complexity of Boolean Relations *30th IEEE Symp. Foundations Computer Science*, pp.562–567,1989.

[RW92] R. Raz, A. Wigderson. Monotone Circuits for Matching Require Linear Depth. *Journal of the ACM*, vol.39, pp.736–744, 1992.

[R92] A.A. Razborov. On the Distributional Complexity of Disjointness. *Theoretical Computer Science*, vol.106, pp.385–390, 1992.

[RV99] V.P. Roychowdhury, F. Vatan. An Almost-Quadratic Lower Bound for Quantum Formula Size. quant-ph/9903042, 1999.

[SV81] W.J. Savitch, D. Vermeir. On the amount of nondeterminism in pushdown automata. *Fund. Inform.*, vol.4, pp.401-418, 1981.

[S97] P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, vol.26, pp.1484–1509, 1997.

[S80] M. Sipser. Lower Bounds on the Size of Sweeping Automata. *Journ. of Computer and System Sciences*, vol.21, pp.195–202, 1980.

[T99] A. Ta-Shma. Classical versus Quantum Communication Complexity. *SIGACT News*, vol.30(3), pp.25–34, 1999.

[V84] L.G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, vol.5, pp.363–366, 1984.

[VC71] V.N. Vapnik, A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, vol.16, pp.264–280, 1971.

[W87] I. Wegener. The Complexity of Boolean Functions. Wiley, 1987.

[W00] R. de Wolf. Characterization of Non-Deterministic Quantum Query and Quantum Communication Complexity. *15th IEEE Conference on Computational Complexity*, 2000.

[Y77] A.C. Yao. Probabilistic Computations: towards a unified measure of complexity. *Proc. 18th IEEE Symp. Foundations of Computer Science*, pp.420–428, 1977.

[Y79] A.C. Yao. Some Complexity Questions Related to Distributed Computing. *Proc. 11th ACM Symp. on Theory of Computing*, pp.209–213, 1979.

[Y93] A.C. Yao. Quantum Circuit Complexity. *34th IEEE Symp. Foundations of Computer Science*, pp.352–361, 1993.

[Z95] U. Zwick. Boolean Circuit Complexity. Lecture Notes, Tel Aviv University, 1995.

Tabellarischer Lebenslauf

Name Adresse	Hartmut Klauck Fechenheimer Str.7 60385 Frankfurt
14.8.1969	Geboren in Detmold
1976-1980	Besuch der Grundschule in Lage
1980-1989	Besuch des Gymnasiums in Lage
1989	Abitur
1989-1995	Studium der Informatik und Philosophie an der Universität Paderborn
1995	Diplom in Informatik mit Auszeichnung, Diplomarbeit „On the Complexity of Approximation, Local Search, and Local Approximation“ bei Prof.Dr. Georg Schnitger
1995-2000	Wissenschaftlicher Mitarbeiter der Johann Wolfgang Goethe-Universität am Lehrstuhl Theoretische Informatik bei Prof.Dr. Georg Schnitger