

Seminar Kommunikationskomplexität

D. Sieling, RWTH-Aachen

Detlef.Sieling@udo.edu

August 2003, Alexander Lay

alex@nwadmin.de

Quelle:

Kushilevitz, E., Nisan, N. (1997). Communication Complexity. Cambridge University Press;
Kapitel 2 bis Seite 21

Nichtdeterministische Kommunikationsprotokolle

Einleitung

In den ersten drei Teilen von Kapitel 2 wird nichtdeterministische Kommunikation eingeführt und es werden verschiedene Schranken betrachtet. Im ersten Teil wird nichtdeterministische Kommunikation motiviert und anhand von Modellen und Definitionen eingeführt. Im zweiten Teil wird eine Schranke der Kommunikationskomplexität bei Protokollen mit entarteten Protokollbäumen angegeben und bewiesen. Der dritte Teil beschäftigt sich mit einem Vergleich zwischen deterministischer und nichtdeterministischer Kommunikationskomplexität.

1. Motivation und Einführung von Nichtdeterminismus

Im Gegensatz zu Partitionen können sich Rechtecke in Überdeckungen überlappen. Überdeckungen benötigen z.T. weniger Rechtecke als Partitionen. Disjunkte Rechtecke, die also eine Partition bilden, stellen ein Modell für deterministische Kommunikation dar. Nichtdisjunkte Rechtecke, die also eine Überdeckung erzeugen, stellen ein Modell für nichtdeterministische Kommunikation dar:

A und B berechnen die Kommunikationsmatrix komplett für alle Eingaben für sich selbst, danach kann A raten, ob das Ergebnis von (x,y) in einem 1-Rechteck liegt und B mitteilen, welches 1-Rechteck dieses ist. B kann dann prüfen, ob y ebenfalls in diesem Rechteck liegt.

Die nichtdeterministische Kommunikationskomplexität ist dabei \log (Anzahl der 1-Rechtecke). Dies ist die Kommunikationskomplexität, die benötigt wird, damit A die Nummer des 1-Rechtecks an B übertragen kann.

Zur genaueren Untersuchung von Rechtecken in Partitionen und Überdeckungen wird folgendes definiert:

Definition 2.1: Sei $f: X \times Y \rightarrow [0,1]$ eine Funktion:

1. Die „protocol partition number of f “ $C^P(f)$:
min. Anzahl von Blättern in einem Protokollbaum von f
2. Die „partition number of f “ $C^D(f)$:
min. Anzahl von monochromatischen Rechtecken in einer Partition von $X \times Y$
3. Die „cover number of f “ $C(f)$:
min. benötigte Anzahl von monochromatischen Rechtecken, um eine Überdeckung auf $X \times Y$ zu bilden
4. $C^z(f)$:
Anzahl der benötigten monochromatischen Rechtecken, um eine Überdeckung der z -inputs von f zu bilden, $z \in \{0,1\}, f(x,y) = z$

Dabei gelten die folgenden Zusammenhänge:

Proposition 2.2: $\forall f: X \times Y \rightarrow [0,1]$:

1. $C(f) \leq C^D(f) \leq C^P(f) \leq 2^{D(f)}$
2. $C(f) = C^0(f) + C^1(f)$
 - $C(f) \leq C^D(f)$:
Anzahl der Rechtecke in einer Überdeckung \leq Anzahl der Rechtecke in einer disjunkten Überdeckung (einer Partition),
da bei einer Partition keine Überlappungen möglich sind
 - $C^D(f) \leq C^P(f)$:
Anzahl der Rechtecke einer Partition = min. Anzahl der Bätter des Protokollbaums,
aus Kapitel 1 bekannt
 - $C^P(f) \leq 2^{D(f)}$:
Aus Kapitel 1: $\log_2 t \leq D(f)$ mit $t = C^P(f)$

Aufgrund der bisherigen Modelle und Definitionen wird nichtdeterministische Kommunikationskomplexität wie folgt definiert:

Definition 2.3: Die nichtdeterministische Kommunikationskomplexität einer booleschen Funktion $f: X \times Y \rightarrow [0,1]$ ist $N^1(f) = \log_2 C^1(f)$.

Die co-nichtdeterministische Kommunikationskomplexität von f ist $N^0(f) = \log_2 C^0(f)$ und es gilt $N(f) = \log_2 C(f)$

Eine Folgerung aus dem Teil 2 ist u.a., dass die „fooling set“-Methode auch bei nichtdeterministischer Kommunikation zu einer unteren Schranke führt. Die Rangmethode gibt hier nur eine untere Schranke an, wenn der Rang auf dem Ring $(\{0,1\}, \text{AND}, \text{OR})$ definiert wird.

2. Komplexitätsschranken bei Protokollen mit entarteten Bäumen

In diesem Teil wird eine obere und untere Schranke der deterministischen Kommunikationskomplexität anhand der Anzahl der Blätter des Protokollbaums gezeigt. Unter Umständen kann der Protokollbaum sehr tief sein und wenig Blätter enthalten, daraus folgt, dass die Kommunikationspartner viele Bits austauschen müssen. Daher sind solche Schranken interessant:

Lemma 2.8: $D(f) = \theta(\log_2 C^P(f))$, genauer: $\log_2 C^P(f) \leq D(f) \leq 2 \log_{3/2} C^P(f)$

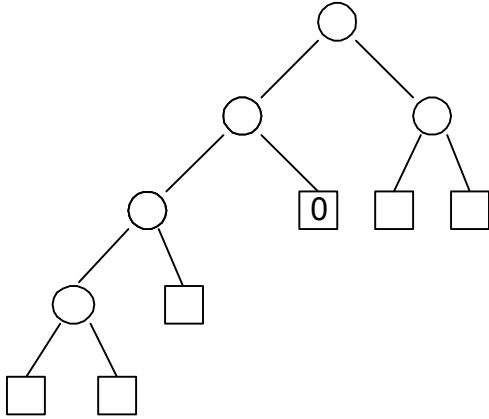
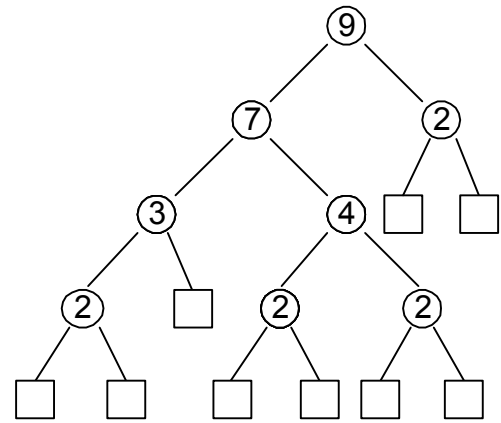
Der Beweis beruht unter anderem auf der Idee, dass ein entarteter Protokollbaum zunächst in einem Preprocessing Schritt besser ausbalanciert wird. Dazu wird ein Unterbaum mit vielen Blättern gesucht, ohne den der gesamte Baum besser balanciert ist. Dieser Unterbaum wird dann entfernt und getrennt betrachtet. Dazu wird keine Kommunikation benötigt, A und B führen beide diesen Schritt alleine durch und kommen dabei immer zu den selben Ergebnissen.

In einem neuen Protokoll müssen A und B zunächst aufgrund der Eingaben x und y entscheiden, ob dieser Unterbaum erreicht wird oder nicht. Wird der Unterbaum erreicht, wird dieser weiter betrachtet. Wird der Unterbaum nicht erreicht, so wird der bisherige Baum ohne den entfernten Unterbaum betrachtet.

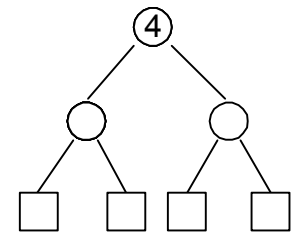
Durch rekursives Wiederholen wird der Protokollbaum, auf dem A und B mit den Eingaben x und y laufen, in jeder Rekursion weiter und besser balanciert. Dies wird so lange durchgeführt, wie es möglich ist. Dann werden die Schranken in Lemma 2.8 erreicht.

Im folgenden Beispiel wird der erste Schritt, das Rebalancieren dargestellt:

Im Baum in dem Beispiel aus rechten Abbildung sind die Knoten jeweils mit der Anzahl der Blätter des Unterbaums beschriftet. In einem ersten Schritt wird hier der Knoten mit 4 Blättern im Unterbaum entfernt und ggf. getrennt betrachtet. Der Baum ohne diesen Unterbaum ist wesentlich besser balanciert (siehe Baum in der unteren Abbildung).



Der entfernte Unterbaum, zu sehen in der Abbildung unten, wird einzeln betrachtet, falls der Knoten mit der Beschriftung 4 durch das Protokoll von A und B auf den Eingaben x und y erreicht wird.



3. Vergleich von det. und nichtdet. Kommunikationskomplexität

Hier geht es um einen Vergleich von deterministischer und nichtdeterministischer Kommunikationskomplexität. Es ist bereits bekannt, dass $D(f)$ exponentiell größer sein kann als $C^1(f)$. Offen ist, ob $D(f) = O(\log C^D(f))$.

Theorem 2.11: \forall Funktionen $f: X \times Y \rightarrow [0,1]$:

$$D(f) = O(N^0(f)N^1(f)) \quad [N(f) = \log C(f)]$$

Unterschied zwischen $D(f)$ und $N(f)$ existiert, ist aber quadratisch.

Das Theorem impliziert, dass der Unterschied zwischen $D(f)$ und $O(\log C^D(f))$ nicht sehr groß ist.

Der **Beweis** beruht auf der Eigenschaft, dass ein 0-Rechteck und ein 1-Recheck entweder nicht in einer Zeile oder nicht in einer Spalte liegen können.

Dann kann ein Protokoll für A und B konstruiert werden, in dem A und B jeweils ein 1-Rechteck aus ihrer Zeile (A) oder Spalte (B) auswählen, das mit höchstens der Hälfte der noch nicht gestrichenen 0-Rechtecke zusammen in einer gleichen Zeile (A) oder Spalte (B) liegt. Alle Rechtecke, die nicht in den gleichen Zeilen (A) oder Spalten (B) wie das gewählte 1-Rechteck liegen, werden gestrichen. Anschließend wird dem Kommunikationspartner mitgeteilt, welches 1-Rechtecke gewählt wurde und ob noch eines gewählt werden konnte. Das entspricht einer Kommunikationskomplexität von $\log C^1(f) + O(1)$ Bits. Da in jedem Schritt die Anzahl der nicht-gestrichenen 0-Rechtecke mindestens halbiert wird, werden maximal $\log C^0(f)$ Schritte benötigt. Existieren keine 0-Rechtecke mehr, so wird $f(x,y) = 1$ ausgegeben. Kann kein 1-Rechteck gefunden, das die nötigen Bedingungen erfüllt, wird $f(x,y) = 0$ ausgegeben.

Zusammen folgt daraus $D(f) = O(\log C^0(f) \log C^1(f)) = O(N^0(f)N^1(f))$.