Kommunikationskomplexität

Seminar über Algorithmen, Prof. Dr. Alt, Sommersemester 2010

Matthias Rost

Freie Universität Berlin

1 Motivation

Ende der 70er Jahre hat Yao [5] als erster Kommunikationskomplexität als eigenes Themengebiet begriffen und den ersten allgemeinen Artikel über dieses Thema veröffentlicht. Kurz zuvor wurde die Kommunikationskomplexität zum ersten Mal in dem Bereich des VLSI-Chip-Designs implizit verwandt.

1.1 AT^2-Schranke zur Berechnung der diskreten Fourier-Transformation auf einem VLSI-Chip

Kommunikationskomplexität wurde zuerst Ende der 1970er Jahre benutzt, um untere Schranken für VLSI (Very-Large-Scale Integration) Schaltkreise zu beweisen. Im folgenden werden wir kurz die Grundideen vorstellen. Das Chip-Design wird heute wie damals davon bestimmt möglichst viel Logik auf einem kleinen Chip unterzubringen, dies reduziert die Kosten. Andererseits soll der Chip möglichst schnell ein Ergebnis berechnen.

Für die VLSI-Designer waren daher untere Schranken in Bezug auf die Rechenzeit und die Größe des Chips von großem Interesse. Für die diskrete Fourier-Transformation konnte so z.B. bewiesen werden, dass $AT^2 \geq \frac{n^2}{16}$ ist. Hierbei bezeichnet A die Fläche des Chips, T die benötigten Taktzyklen und n die Länge der binären EingabeThompson [4].

1.2 Allgemeine AT^2-Schranken für VLSI-Chips

AT^2-Schranken lassen sich auch für viele andere Funktionen aufstellen. Zuvor müssen wir jedoch die folgenden Annahmen treffen:

1. Der Chip habe die Abmessungen $a \times b$, mit $a \leq b$. Hierbei werden die Abmessungen in der Einheit λ gemessen, welche den minimalen Abstand zweier Leitungen bezeichnet. Die Fläche des Chips wird mit A bezeichnet.
2. Dem Chip werden n Eingabebits übergeben.
3. Pro Taktzyklus kann über eine Leitung des Chips nur ein Bit übertragen werden.

Ein Beweis über die AT^2-Komplexität kann nun wie folgt vorgenommen werden:

1. Wir wählen einen Schnitt durch den Chip, so dass die eine Hälfte der Eingabebits von der anderen Hälfte der Eingabebits getrennt wird. Für einen rechteckigen Chip liegt die Länge des Schnitts $|w|$ in $O(b)$.
2. In Abhängigkeit der betrachteten Funktion nehmen wir an, dass mindestens αn, mit $\alpha > 0$, viele Bits diesen Schnitt überqueren müssen, damit das Ergebnis richtig berechnet werden kann.
3. Innerhalb von T Taktzyklen können maximal $|w|T$ viele Bits zwischen den beiden Hälften übertragen werden.
4. Somit gilt $\alpha n \leq |w|T$.
5. Da $|w| \in O(b)$ können wir annehmen, dass $|w|^2 \in O(A)$ ist.
6. Durch vorheriges Quadrieren und Einsetzen erhalten wir $\alpha^2 n^2 \leq |w|^2 T^2$ bzw. $n^2 \in O(AT^2)$

2 Grundlegende Definition

Definition 1. *Protokoll*

Ein Protokoll P auf der Menge $X \times Y$ von Eingaben und dem Bild Z ist ein binärer Baum, in dem jeder interner Knoten v mit einer Funktion $a_v : X \to \{0,1\}$ oder einer Funktion $b_v : Y \to \{0,1\}$ beschriftet ist und jedes Blatt ein Element aus Z ist.

Das Ergebnis des Protokolls bei Eingabe $(x, y) \in X \times Y$ ist der Wert des Blatts in den das Protokoll gelangt, wenn der Baum beginnend bei der Wurzel wie folgt traversiert wird:

Sofern der innere Knoten v mit einer Funktion a_v beschriftet ist, wird als nächster Knoten das linke Kind ausgewählt, falls $a_v(x) = 0$ und das rechte Kind, wenn $a_v(x) = 1$. Wenn der Knoten mit einer Funktion b_v beschriftet ist, wird analog in Abhängigkeit von $b_v(y)$ vorgegangen.

Definition 2. *Kosten eines Protokolls*

Die Kosten eines Protokolls bei Eingabe $(x, y) \in X \times Y$ ist die Länge des Pfades von der Wurzel bis zum Blatt. Die Kosten des Protokolls ohne spezifische Eingabe ist die Höhe des Protokoll-Baums.

Definition 3. *Deterministische Kommunikationskomplexität*

Sei eine Funktion $f : X \times Y \to Z$ gegeben. Die deterministische Kommunikationskomplexität $D(f)$ Funktion f ist das Minimum über alle Kosten derjeniger Protokolle, die f berechnen.

Definition 4. *Kombinatorisches Rechteck*

Ein kombinatorisches Rechteck aus $X \times Y$ ist eine Teilmenge $R \subseteq X \times Y$, so dass $R = A \times B$ mit $A \subseteq X$ und $B \subseteq X$.

Theorem 1. *$R \subseteq X \times Y$ ist ein (kombinatorisches) Rechteck $\Leftrightarrow (x_1, y_1) \in R \wedge (x_2, y_2) \in R \Rightarrow (x_1, y_2) \in R$ gilt.*

Beweis. $\Leftarrow$: Seien $A = \{x | \exists y' : (x, y') \in R\}$ und $B = \{y | \exists x' : (x', y) \in R\}$. Wir zeigen, dass $R = A \times B$ ist.

- $R \subseteq A \times B$: Da $(x, y) \in R$ gilt nach Definition von A bzw B auch $x \in A$ und $y \in B$.
- $A \times B \subseteq R$: Sei $(x, y) \in A \times B$. Da $x \in A$ muss es ein y' geben, so dass $(x, y') \in R$. Analog muss es ein x' geben, so dass $(x', y) \in R$, da $y \in R$. Gemäß der Annahme $(x_1, y_1) \in R \wedge (x_2, y_2) \in R \Rightarrow (x_1, y_2) \in R$ ist daher auch $(x, y) \in R$.

$\Rightarrow$: Sei $R = A \times B$ ein Rechteck. Wenn $(x_1, y_1) \in R$ und $(x_2, y_2) \in R$ so gilt $x_1 \in A$ und $y_2 \in B$. Da $R = A \times B$ muss daher auch $(x_1, y_2) \in R$ sein.

Definition 5. *Sei P ein Protokoll mit Definitionsbereich $X \times Y$ und v ein Knoten innerhalb des Protokoll-Baums. Wir definieren als R_v die Menge von Eingaben $(x, y) \in X \times Y$ welche bei Traversierung des Baums den Knoten v erreichen.*

Korollar 1. *Wenn L die Menge an Blättern des Protokoll-Baums ist, so ist $\{R_{l \in L}\}$ eine Partition des Eingaberaums.*

Beweis. Wir können annehmen, dass für jeden Knoten v eines Protokollbaums $R_v \neq \emptyset$ gilt. Sollte dies nicht gelten, kann der Knoten v mit allen seinen Kindern aus dem Baum entfernt werden. Für eine Eingabe $(x, y) \in X \times Y$ gibt es genau ein Blatt, dass bei der Traversierung erreicht wird und somit dem Wert der beschriebenen Funktion f gleicht.

Theorem 2. *Sei P ein Protokoll und v ein Knoten des entsprechenden Protokollbaums, so ist R_v ein Rechteck.*

Beweis. Wir führen den Beweis durch Induktion über die Tiefe des Knotens v im Baum. Der Induktionsanker ist der Wurzelknoten $wurzel$. Da $R_{wurzel} = X \times Y$ gilt, ist R_{wurzel} offensichtlich ein Rechteck. Für einen Knoten v unterhalb der Wurzel können wir o.B.d.A. das folgende annehmen:

- Der Elternknoten von v ist w.
- v ist das linke Kindelement von w.
- Der Knoten w ist mit einer Funktion $a_w : X \to \{0, 1\}$ beschriftet.

Es folgt, dass $R_v = R_w \cap \{(x, y) | a_w(x) = 0\}$. Gemäß der Induktionsannahme gilt für den Elternknoten w, dass $R_w = A_w \times B_w$ ein Rechteck ist. Somit ist $R_v = (A_w \cap \{x | a_w(x) = 0) \times B_w$.

Definition 6. *Eine Teilmenge $R \subseteq X \times Y$ wird als f-monochromatisch bezeichnet, falls die Funktion f auf R konstant ist.*

Lemma 1. *Jedes Protokoll P einer Funktion f induziert eine Partition von $X \times Y$ in f-monochromatische Rechtecke. Die Anzahl an Rechtecken ist die Anzahl der Blätter von P.*

Korollar 2. *Wenn jede Partition von $X \times Y$ in f-monochromatische Rechtecke mindestens t Rechtecke erfordert, gilt $D(f) \geq log_2 t$.*

Beweis. Gemäß Lemma 1 induzieren die Blätter eines jeden Protokoll eine Partition in f-monochromatische Rechtecke. Wenn jede Partition immer mindestens t f-monochromatische Rechtecke erfordert, muss es daher auch mindestens t viele Blätter geben. Somit ist die Höhe des Baums von unten durch $log_2 t$ beschränkt.

3 Untere Schranken

3.1 Fooling Sets und Rechteck Größe

Das Korolloar 2 erlaubt es uns, über die Anzahl an erforderlichen Rechtecken eine Aussage über die Kommunikationskomplexität zu machen.

Definition 7. *Sei $f : X \times Y \to \{0,1\}$. Eine Menge $S \subset X \times Y$ wird als Fooling Set bezeichnet, wenn es einen Wert $z \in \{0,1\}$ gibt, so dass*

- *Für jedes $(x,y) \in S$ gilt $f(x,y) = z$.*
- *Für alle paarweise verschiedenen Elemente (x_1,y_1) und (x_2,y_2) aus S gilt entweder $f(x_1,y_2) \neq z$ oder $f(x_2,y_1) \neq z$.*

Theorem 3. *Wenn die Funktion f ein Fooling Set S der Größe t hat, gilt $D(f) \geq log_2 t$.*

Beweis. Wir zeigen, dass zwei Elemente (x_1,y_1) und (x_2,y_2) aus S nie in dem gleichen monochromatischen Rechteck liegen können. Angenommen in einem monochromatischen Rechteck R würden mindestens zwei Elemente aus S liegen. Aus Theorem 1 folgt, dass dann auch (x_1,y_2) sowie (x_2,y_1) im gleichen monochromatischen Rechteck R liegen müssten. Da nach Definition von S aber entweder $f(x_1,y_2)$ oder $f(x_2,y_1)$ von dem Wert z des Fooling Sets S abweichen muss, können diese nicht in dem gleichen monochromatischen Rechteck liegen.

Somit muss es mindestens t verschiedene monochromatische Rechtecke geben. Das Theorem folgt aus Korollar 2.

3.2 Rang-Methode

Definition 8. M_f

Sei $f : X \times Y \to \{0,1\}$ eine Funktion. Wir assoziieren mit der Funktion f die Matrix M_f, in welcher der Funktionswert von f abgespeichert ist. Die Matrix sei über die Eingabewerte (x,y) indizierbar.

Theorem 4. *Für eine Funktion $f : X \times Y \to \{0,1\}$ gilt $D(f) \geq log_2 rang(f)$.*

Beweis. Sei P ein zu f gehöriges Protokoll und sei B_1 die Menge an Blättern, welche das Ergebnis 1 repräsentieren. Für jedes Blatt $b \in B_1$ definieren wir eine Matrix M_b wobei $M_b(x,y) = 1 \Leftrightarrow (x,y) \in R_b$ und $M_b(x,y) = 0 \Leftrightarrow (x,y) \notin R_b$.

Für jedes (x,y) der Eingabe gilt:

- Sofern $f(x,y) = 1$ gibt es genau eine Matrix M_b mit $M_b(x,y) = 1$. Für alle anderen Matrizen mit $c \neq b$ gilt $M_c(x,y) = 0$.
- Sofern $f(x,y) = 0$, so ist auch jede Matrix M_b an der Stelle (x,y) 0.

Somit gilt $M_f = \sum_{b \in B_1} M_b$. Da für beliebige Matrizen weiterhin $rang(A+B) \leq rang(A) + rang(B)$ gilt, folgt $rang(M_f) \leq \sum_{b \in B_1} rang(M_b)$.

Da jede Matrix M_b ein kombinatorisches Rechteck beschreibt, folgt $rang(M_b) = 1$. Letztlich können wir feststellen, dass $rang(M_f) \leq |B_1| \leq |B|$ ist. Somit muss ein Protokoll P immer mindestens $rang(M_f)$ viele Blätter haben. Das Theorem folgt aus dem Korollar 2.

Literaturverzeichnis

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 2009.

[2] Randal E. Bryant. On the complexity of vlsi implementations and graph representations of boolean functions with application to integer multiplication. *IEEE Transactions on Computers*, 40:205–213, 1998.

[3] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, New York, NY, USA, 1997.

[4] C. D. Thompson. Area-time complexity for vlsi. In *STOC '79: Proceedings of the eleventh annual ACM symposium on Theory of computing*, New York, NY, USA, 1979. ACM.

[5] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *STOC '79: Proceedings of the eleventh annual ACM symposium on Theory of computing*, New York, NY, USA, 1979. ACM.