# G. Kommunikationstheorie

Autor: Markus Möller

- Bezug: Vorlesung bei Herrn Schnittger und die dort angegebene Referenzliteratur.
- Dieser Extrakt entstand als Vorbereitung auf meine Diplomprüfung (Teil: Theoretische Informatik). Er faßt einige Themen einfach zusammen und mag etwas unorthodox erscheinen.
- Dieser Text ist unvollständig.
- Erstellt auf Apple Macintosh.

# 1. Feste Ein- /Ausgabezerlegung

**>>Kommunikationsmatrix:** Man betrachtet die Kommunikationsmatrix des Problems. Deren Zeilen sind mit den möglichen Eingaben für Prozessor A numeriert, ihre Spalten mit den möglichen Eingaben für Prozessor B. Die Matrixeinträge enthalten die jeweilige Ausgabe der beiden Prozessoren für die zugehörige Eingabekombination.

## 1.1 Fooling-Set Methode

Es geht darum zu zeigen, daß für f Matrixeinträge verschiedene Berechnungen notwenig sind. Für diese benötigt man dann mindestens $\log f$ Bits. Kann auf alle Funktionen mit $\{0,1\}^n \rightarrow \{0,1\}^m$ angewendet werden.

So ein Foolingset von f Einträgen muß einige Bedingungen erfüllen:

- Keine zwei liegen in derselben Zeile oder Spalte.
- Für je zwei Elemente muß mindesten eine der folgenden Bedingungen gelten:
  - Mache „Rechteck“. Auf horizontaler Ebene einmal keine Gleichheit für Prozessorausgabe A.
  - Auf vertikaler Ebene einmal keine Gleichheit für Prozessorausgabe B im selben „Rechteck“.

Der Beweis geht so, daß man behauptet die Foolingsetelemente hätten gleiche Berechnung und dann nachweist, daß das für die anderen beiden im „Rechteck" ebenfalls zutrifft. Dies ist dann ein Widerspruch zur Definition des Foolingset.

Eine einfache, jedoch schwächere Anwendung ist, bei Funktionen mit $\{0,1\}^n \rightarrow \{0,1\}$ eine Einheitsmatrix in der Kommunikationsmatrix als Foolingset zu suchen; notfalls durch Zeilen- und Spaltenpermutation.

Diese Methode ist einfach anzuwenden, liefert aber im allgemeinen keine guten unteren Schranken.

## 1.2 Rang-Methode

Man betrachtet Funktionen $\{0,1\}^n \rightarrow \{0,1\}$. Dann ist der Kommunikationsaufwand $C(f, \Pi_{ein}, \Pi_{aus}) \geq \lceil \log \mathrm{Rang}(\mathrm{Komm.matrix}) \rceil$.

Der Beweis geht über die Berechnung. Die Prozessoren senden entweder eine 1 oder eine 0. Damit läßt sich die Matrix immer aufteilen. Am Ende der Berechnung stehen in der letzten Matrix dieselben Ausgaben, weil sie dieselbe Berechnung hatten. Man betrachtet in jedem Schritt die Teilmatrix, deren Rang größer als der halbe Rang der Muttermatrix ist. Geht, weil die Ränge von Teilmatrizen in der Summe mindestens so groß sind wie der Rang der Muttermatrix.

Es ist nicht bekannt, wie gut die Rangmethode im allgemeinen ist.

## 1.3 Zerlegungsmethode

Eine Zerlegung einer Funktion, die durch eine Kommunikationsmatrix dargestellt wird, ist eine Zahl k. k ist die *kleinste Anzahl disjunkter monochromatischer Teilmatrizen*. Die Kommunikationkomplexität ist dann ≥log(k)-1. So eine kleinste Matrix enthält dann alle Koordinaten mit gleicher Berechnung. Man kommt darauf, wenn man überlegt, daß durch die Berechnung die KM immer zeilen- bzw. spaltenweise aufgeteilt wird bzgl. gesendeter Bits. Also ist die Anzahl verschiedener Berechnungen mit der Anzahl dieser Teilmatrizen identisch. Die Komplexität ist der log davon, weil man mit 2 hoch log(k) Bits k Berechnungen machen kann.

Ist schwierig anzuwenden, aber ziemlich exakt.

Einfachere Möglichkeiten:

- Komplexität ≥ log(Anzahl der 1en in KM/Größe der größten 1-chromatischen Teilmatrix von KM)-1 bzw. analog 0en und 0-chromatisch und dann max von beiden. Logisch, denn wenn man die *größte monochromatische Teilmatrix* nimmt (sei es o.B.d.A. mit 1en) und die Gesamtzahl der 1en der KM dadurch teilt, bekommt man eine untere Schranke für die Anzahl der Teilmatrizen. Denn je größer die Teilmatrizen (disjunkte!), desto kleiner ist k oben. Die Teilmatrizen der anderen Farbe (hier 0) gar nicht mitgerechnet…
- Wähle eine *Teilmatrix* von KM und führe den Punkt oben auf dieser durch anstatt auf KM. Jede Zerlegung von der gesamten KM hat nämlich mindestens so viele Teilmatrizen wie die betrachtete.

## 1.4 Vergleich der Methoden

Die Zerlegungsmethode ist besser als die beiden anderen. Sie liefert mindestens so große untere Schranken wie die beiden anderen.

Für die Zerlegungsmethode gilt: log(Zerl.)-1 ≤ Komm.kompl. ≤ $(\log(\text{Zerl.})+1)^2$.

Die Rangmethode liefert eine um höchstens den Faktor 1/2 schwächere untere Schranke als die Fooling-Set-Methode.

Die Fooling-Set-Methode ist nie entscheidend besser als die Rangmethode, kann aber drastisch schlechter sein.

# 2. Freie Ein- /Ausgabezerlegung

Bei freier Wahl der Zerlegungen wird die Kommunikationskomplexität definiert als das Minimum über alle Kommunikationskomplexitäten mit ausgewogener Eingabezerlegung und beliebiger Ausgabezerlegung.

## 2.1 Anwendung für VLSI

Ein Eingabebit erscheint bei einem Chip an einem vorher festgelegten Eingabeport zu einem vorher festgelegten Zeitpunkt. Analog die Ausgabe. Er rechnet synchron, d.h. alle Bausteine folgen dem Takt ein globalen Uhr. In jedem Zeittakt kann über jeden Draht höchstens ein Bit gesendet werden.

Bausteine sind Punkte mit ganzzahligen Koordinaten, Drähte verlaufen parallel zu den Achsen.

Um die beiden Kommunikationspartner in einem Chip festzulegen, wird dieser kurzerhand geteilt. Man führt einen Schnitt mit maximal einer Stufe durch. Er verläuft zwischen den Gitterbahnen, trifft somit nie einen Baustein und kreuzt Drähte nur in einem Punkt. Der Schnitt ist höchstens $4\sqrt{\text{Fläche(Chip)}} + 1$ groß und die beiden Teile besitzen maximal jeweils 2/3 aller Punkte.

Man ist besonders an der Fläche und der Zeit interessiert, die ein Chip benötigt, um eine Funktion zu berechnen. Die Fläche eines Chips bestimmt die Produktionskosten.

Sei f mit $\{0,1\}^n \to \{0,1\}^m$ gegeben. Wenn C ein Chip ist, der f mit Fläche A in Zeit T berechnet, dann gilt: $A \quad T^2 = \Omega(C(f)^2)$.

Prozessor A erhält alle Eingaben in Teil 1 und Prozessor B die von Teil 2. A bzw. B sind für alle Ausgaben von 1 bzw. 2 verantwortlich. Das Protokoll simuliert den Informationsaustausch zwischen 1 und 2.

Zeigt man über die Größe des Schnittes und die Anzahl der Schritte. In jedem Takt können nur so viele Bits kommuniziert werden, wie der Schnitt groß ist. Damit ist Schritte mal Schnitt ≥ C(f). Umformen… fertig.

# 2.2 Untere Schranken bei freier Zerlegung

Sei f mit $\{0,1\}^n \to \{0,1\}^m$ gegeben mit Zerlegungen $\Pi_{ein} = (X_A, X_B)$ und $\Pi_{aus} = (Y_A, Y_B)$. Wenn es eine Eingabe $y_0 \quad \{0, 1\}^{|X_B|}$ gibt, für die gilt: $\left|\{ f_B(x, y_0) \| x \quad \{0, 1\}^{|X_A|}\}\right| > 2^{w-1}, w \quad$ À , dann ist $C(f, \Pi_{ein}, \Pi_{aus}) \geq w$.

D.h., wenn man für B eine Eingabe findet, die mehr als $2^{w-1}$ Ausgaben –für die B zuständig ist– ermöglicht (abhängig von allen Eingaben für A), dann müssen, um alle verschiedenen Ausgaben erzeugen zu können, mindestens w Bits ausgetauscht werden. Dadurch hat man eine untere Schranke für die KK, da nicht berücksichtigt wird, welche Infos A für seinen Teil der Ausgabe benötigt.

# 3. Nichtdeterministische und probabilistische Kommunikation