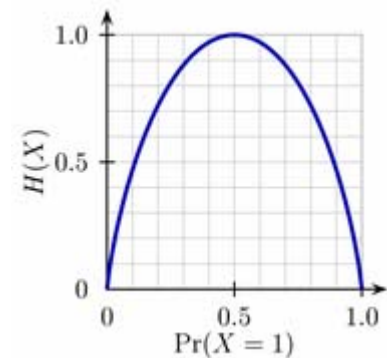


# Information und Kommunikation

Hartmut Klauck  
Universität Frankfurt  
SS 07  
18.6.



# Untere Schranken

- Wenn wir eine untere Schranke für  $R^1(f) \geq R^{1,\text{pub}}(f)$  zeigen wollen ist unser Rezept:
  - Finde eine schwierige Verteilung auf den Eingaben
  - Beschränke die deterministische Komplexität mit Fehler
- Die Verteilungen sind dabei immer ungefähr balanciert, d.h.  
 $1/3 \leq \text{Prob}(f^{-1}(1)) \leq 2/3$

# Die Index Funktion

- Wir betrachten zunächst die Index Funktion  $\text{Ind}(x,i)=x_i$  mit  $x \in \{0,1\}^n$  und  $i \in \{1, \dots, n\}$
- **Theorem 18.1**
  - $R^1(f) \geq R^{1,\text{pub}}(f) \geq D^{1,\mu}_{1/3}(f)$   
 $\geq (1-H(1/3))n$  für die uniforme Verteilung  $\mu$
- **Beweis:**
  - Ein Protokoll für Ind entspricht einem random access code!
  - Die untere Schranke folgt mit Theorem 4.1

# Anwendungen

- Automaten:
  - Man kann probabilistische Automaten definieren (mit probabilistischen Zustandsübergängen)
  - Es ist leicht zu sehen, dass z.B. prob. Automaten für  $\{x, x: x \in \{0,1\}^n\}$  nur Größe  $\text{poly}(n)$  benötigen, während deterministische Automaten Größe  $2^n$  haben müssen
  - Wie im deterministischen Fall liefert die randomisierte Einweg-Kommunikationskomplexität untere Schranken für prob. Automaten

# Anwendungen

- Formeln:
  - Probabilistische Formeln können zusätzlich zur Eingabe Zufallsbits lesen
  - Rechnen mit Fehler 1/3
  - Formellänge etc. definiert in offensichtlicher Weise
- Wir erhalten eine Neciporuk Methode für probabilistische Formeln, indem wir die deterministische Einweg-Komplexität durch die randomisierte Einweg-Komplexität ersetzen
- Da wir Ind bereits untersucht haben, erhalten wir, dass prob. Formeln für ISA Länge  $\Omega(n^2/\log n)$  haben

# Anwendungen

- Formeln
  - Es gibt auf der anderen Seite eine Funktion, für die
    - prob. Formeln Länge  $O(n)$  haben
    - det. Formeln Länge  $\Theta(n^{3/2})$  haben
    - Funktion: Matrix Multiplikations Verifikation:
      - Gegeben  $A, B, C$  Matrizen über  $Z_2$
      - ist  $AB=C$ ?
      - Det. Formellänge  $\Theta(n^3)$  [Schulmethode optimal]
      - Prob. Formellänge  $O(n^2)$  mit Fingerprinting Technik

# Eine allgemeine Technik

- Wir wollen nun andere Funktionen untersuchen
- Zunächst betrachten wir Reduktionen von Ind auf andere Funktionen
- **Definition 18.2**
  - Die VC-Dimension einer Matrix mit 0/1-Einträgen ist das maximale  $k$ , so dass es eine  $2^k \times k$  Teilmatrix gibt, die der Kommunikationsmatrix des Indexproblems entspricht
  - Die VC Dimension einer Funktion  $f$ ,  $VC(f)$  ist die VC-Dimension der Kommunikationsmatrix von  $f$
- Begriff stammt aus der Lerntheorie und wird meist anderslautend aber äquivalent definiert

# VC-Dimension

- Theorem 18.3
  - Für alle Funktionen  $f$  gilt:  $R^{1,\text{pub}}(f) \geq \Omega(\text{VC}(f))$
- Beweis:
  - Es gibt eine Teilmatrix der Kommunikationsmatrix von  $f$ , die gleich der von  $\text{Ind}$  mit Eingabelänge  $k$  ist. Daher können wir eine Verteilung  $\mu$  definieren, die uniform auf dieser Teilmatrix ist. Damit gilt

$$R^{1,\text{pub}}(f) \geq D^{1,\mu}_{1/3}(f) \geq D^{1,\text{unif}}_{1/3}(\text{Ind}_k) \geq (1-H(1/3))k,$$

wobei  $k = \text{VC}(f)$

# Beispiele

- Disj:  $VC(DISJ)=n$ , also gilt  $R^1(DISJ)=\Theta(n)$
- Eq:  $VC(EQ)=1$ ;  $R^{1,pub}(f)=O(1)$
- Greater Than GT:
  - $GT(x,y)=1$  wenn  $x \geq y$  (als  $n$  Bit Zahlen)
  - $VC(GT)=1$
  - Man kann zeigen, dass  $R^{1,pub}(GT)=\Theta(n)$
  - D.h. unsere Methode ist in manchen Fällen sehr schlecht!

# VC-Dimension

- Wir betrachten  $D^{1,\mu}(f)$  und  $VC(f)$
- Sei  $\mu$  eine Produktverteilung, d.h. es gebe  $\mu_A$  und  $\mu_B$  mit  $\mu(x,y) = \mu_A(x) \cdot \mu_B(y)$
- **Theorem 18.4**  
 $\max_{\mu} D^{1,\mu}(f) = \Theta(VC(f))$ , wenn  $\mu$  über Produktverteilungen läuft
- D.h. für GT sind nur Verteilungen schwierig, die keine Produktverteilungen sind

# VC-Dimension

- Beweisidee:
- Der Beweis verwendet Ergebnisse der Lerntheorie
- Alice möchte, dass Bob die mit  $x$  indizierte Zeile der Kommunikationsmatrix approximativ "lernt" (d.h.  $f_x(y) = f(x,y)$  mit hoher Ws. bestimmen kann, bezüglich einer Verteilung auf den  $y$ )
  - diese Verteilung ist  $\mu_B$
- Dabei wird Bob's „Lernerfolg“ gemäß der Verteilung  $\mu_B$  auf den  $y$  bewertet
- Fakt: Es reichen  $O(VC(f))$  viele zufällige Beispiele aus  
ein Beispiel ist ein Paar  $(y, f(x,y))$
- Alice kann diese Beispiele liefern:
  - per public coin wird  $y$  zufällig gezogen
  - Alice kommuniziert das Bit  $f(x,y)$
  - Gesamtkommunikation  $O(VC(f))$

# Zusammenfassung

- Wir können somit für Produktverteilungen die deterministische Kommunikation mit Fehler exakt über die VC-Dimension charakterisieren
- Diese Technik versagt für Funktionen, die nur auf nicht-Produktverteilungen schwer sind
- Deshalb beschreiben wir eine Technik, die für alle Verteilungen funktioniert

# Greater Than

- Zuvor wollen wir allerdings  $R^1(GT)$  genauer bestimmen.
- **Theorem 18.5**
  - $R^{1,\text{pub}}(GT) \geq \Omega(n/\log n)$
- Bemerkung: Man kann auch  $\Omega(n)$  zeigen

# Beweis 18.5

- Angenommen wir erhalten ein rand. Einweg Protokoll für  $GT$  mit  $n/(100 \log n)$  Kommunikation und Fehler  $1/3$
- Durch  $10 \log n$  fache Ausführung und Mehrheitsentscheid erhalten wir ein Protokoll mit Fehler  $1/n^2$  und Kommunikation  $n/10$
- Bob kann die erhaltene Nachricht nun mehrmals verwenden:
  - Für jedes  $y$  gilt  $\text{Prob}(\text{auf } x,y \text{ wird nicht korrekt gerechnet}) \leq 1/n^2$
  - Für jede Menge von  $t$  Eingaben  $y_1, \dots, y_t$  gilt  $\text{Prob}(\exists i \in \{1, \dots, t\}: \text{auf } x, y_i \text{ wird nicht korrekt gerechnet}) \leq t/n^2$

# Beweis 18.5

- Bob verwendet nun binäre Suche, um den Wert von  $x$  exakt zu bestimmen
  - Ist  $x \geq 2^n/2$ ?
  - Wenn ja, ist  $x \geq 2^n(3/4)$ ?
  - etc.
- Dazu braucht Bob maximal  $n$  Vergleiche mit Zahlen  $y_1, \dots, y_n$
- Diese Berechnung funktioniert mit Wahrscheinlichkeit  $1-1/n$  für alle  $y_i$
- Bob kann also  $x$  mit hoher Ws. bestimmen
- Diese Aufgabe benötigt  $\Omega(n)$  Kommunikation, wenn  $x \in \{0,1\}^n$  uniform verteilt ist, denn die Entropie der Nachrichten muss dann  $\Omega(n)$  sein
- (Insbesondere bilden die Nachrichten einen random access code für  $x$ ).

# Eine allgemeine Technik

- Wir wollen nun die randomisierte Einweg-Kommunikation kombinatorisch charakterisieren
- Vergleich: deterministische Einweg Kommunikation ist gleich dem  $\log$  der Anzahl der verschiedenen Zeilen in der Komm. Matrix
- **Definition 18.6**  
Ein Einweg Rechteck  $R$  mit Fehler  $\varepsilon$  unter einer Verteilung  $\mu$  auf den Eingaben ist eine Teilmenge der Zeilen, so dass Bob auf  $R$  die Funktion  $f$  mit Fehler  $\varepsilon$  unter  $\mu$  (eingeschränkt auf  $R$ ) berechnen kann

# Einweg Rechtecke

- Ein deterministisches Protokoll für  $f$  unter  $\mu$  mit Fehler  $\varepsilon$  ist eine Partition der Zeilen entsprechend der gesendeten Nachrichten.
- Für die meisten Nachrichten muss Bob in der Lage sein, die Funktion  $f$  mit kleinem Fehler zu berechnen (auf den Zeilen die der Nachricht entsprechen)
- $D_{\varepsilon}^{1,\mu}(f)$  ist damit nichts anderes als der log der Anzahl notwendiger Einweg Rechtecke
- Die *Größe* eines Einweg Rechtecks  $R$  ist  $\mu(R) = \sum_{x,y \in R} \mu(x,y)$

# Einweg Rechtecke

- **Theorem 18.6**

Für alle  $\mu, \varepsilon$  gilt:  $D^{1, \mu_\varepsilon}(f) \geq \Omega(\log 1/s)$ , wenn

$s = \max_R \mu(R)$ , wobei  $R$  über alle Einweg Rechtecke mit Fehler  $2\varepsilon$  läuft

# Beweis 18.6

- Gegeben sei ein deterministisches Einweg Protokoll für  $f$  mit Fehler  $\varepsilon$  on  $\mu$
- Zu jeder Nachricht von Alice gibt es ein Einweg Rechteck (eine Menge von Zeilen)
- Der erwartete Fehler ist  $\varepsilon$
- $Z[x,y]=1$  gdw das Protokoll auf  $x,y$  irrt
- $E_{x,y} (Z[x,y]) \leq \varepsilon$
- $$\begin{aligned} E_{x,y} (Z[x,y]) &= \sum_R \sum_{x,y \in R} \mu(x,y) Z[x,y] \\ &= \sum_R \mu(R) \sum_{x,y \in R} \mu(x,y) / \mu(R) Z[x,y] \\ &= E_R E_{x,y \in R} Z[x,y] \end{aligned}$$
- D.h. wenn man  $R$  zufällig gemäß seiner Größe zieht, ist der erwartete Fehler  $\varepsilon$

# Beweis 18.6

- Was ist die erwartete Größe von  $R$ ?
- Ein det. Einweg Protokoll habe Kommunikation  $c$ , somit  $2^c$  Nachrichten/Rechtecke
- $\sum_R \mu(R)=1$
- Die Rechtecke mit Größe  $\leq 2^{-2c}$  tragen zu der Summe nur  $1/2^c$  bei.
- Somit muss es ein Einweg Rechteck der Größe  $2^{-2c}$  und mit Fehler  $\leq \varepsilon + 1/2^c \leq 2\varepsilon$  geben
- Dies ist äquivalent zur Aussage des Theorems

# Untere Schranken

- D.h. unser Rezept für untere Schranken ist wie folgt:
  - Finde eine schwierige Verteilung
  - Zeige, dass für alle Einweg Rechtecke mit Fehler  $\varepsilon$  ihre Größe  $\mu(R)$  klein ist
- Liefert dieses Vorgehen immer gute untere Schranken?

# Untere Schranken

- **Theorem 18.7**
  - Sei  $b(v, \varepsilon, f)$  die maximale Größe eines Einweg Rechtecks mit Fehler  $\varepsilon$  unter der Verteilung  $v$  auf den Eingaben
  - $b(\varepsilon, f) = \min_v b(v, \varepsilon, f)$
  - Für alle Verteilungen  $\mu$  auf den Eingaben für Funktionen  $f$  gilt:  $D_{\varepsilon}^{1, \mu}(f) \leq O(-\log b(\varepsilon, f))$
- Damit folgt:
- **Korollar 18.8**
  - $R^{1, \text{pub}}(f) = \Theta(-\log b(1/3, f))$
- In Worten: die maximale Größe von Einweg Rechtecken mit kleinem Fehler unter einer schwierigsten Verteilung bestimmt die randomisierte Einweg Kommunikation

# Beweis 18.8

- Für alle  $\mu$ :  $D_{\varepsilon}^{1,\mu}(f) \leq O(-\log b(\varepsilon, f))$
- $R^{1,\text{pub}}(f) = \max_{\mu} D_{1/3}^{1,\mu}(f)$
- Also  $R^{1,\text{pub}}(f) \leq O(-\log b(\varepsilon, f))$
- Die  $\geq$  Richtung folgt aus 18.6

# Beweisskizze 18.7

- Wir konstruieren für jede Verteilung  $\mu$  ein deterministisches Einwegprotokoll mit Fehler  $\varepsilon$
- Wir wissen, dass für jede Verteilung  $\nu$  ein großes Einweg Rechteck mit kleinem Fehler existiert
- Wir starten mit  $\mu$  und erhalten ein großes Rechteck. Diese wird eine der Nachrichten von Alice
- Wir schränken  $\mu$  auf die noch nicht abgedeckten Zeilen ein und erhalten eine Verteilung  $\mu'$
- Wieder bekommen wir ein großes Einweg Rechteck, das wir als Nachricht von Alice verwenden. Sollte dieses Rechteck sich mit dem ersten überschneiden, entfernen wir die entsprechenden Zeilen aus dem neuen Rechteck
- Usw. Nach  $O(1/b(\varepsilon, f))$  Schritten sind fast alle Zeilen überdeckt
- Die restlichen Zeilen können

# Zweiweg Kommunikation

- Wir kommen jetzt zu Protokollen mit Interaktion zwischen Alice und Bob
- $f: X \times Y \rightarrow Z$  sei durch Alice und Bob zu berechnen
- Die Kommunikation verläuft in Runden
- In jeder Runde berechnet der jeweilige Spieler aus seiner Eingabe und den bisherigen Nachrichten die neue Nachricht und sendet diese an den anderen Spieler
- Das Protokoll endet, indem beide Spieler die Ausgabe berechnen

# Zweiweg Kommunikation

- **Definition 18.9**

- Ein Protokoll ist durch einen binären Baum gegeben.
- Jeder innere Knoten ist markiert mit einer Funktion  $a_v: X \rightarrow \{0,1\}$  oder  $Y \rightarrow \{0,1\}$
- Jedes Blatt ist mit einer Ausgabe  $z \in Z$  markiert
- Das Protokoll rechnet auf Eingabe  $x, y$ , indem an der Wurzel startend an jedem Knoten mit Markierung  $a_v$  entweder  $a_v$  auf  $x$  oder auf  $y$  angewendet wird. Wenn das Ergebnis 0 ist, wird im Baum zum linken Nachfolger, sonst zum rechten gelaufen
- Die Ausgabe wird am erreichten Blatt abgelesen
- Die Kommunikationskosten eines Protokolls sind durch die Länge des längsten Pfads im Baum gegeben

# Zweiweg Kommunikation

- **Definition 18.10**
  - Die deterministische Kommunikationskomplexität von  $f$  ist  $D(f)$ , die minimalen Kommunikationskosten eines Protokolls, dass  $f$  berechnet
- Wie können wir  $D(f)$  beschränken?
- Betrachten wir EQ
- Klar:  $D(EQ) \leq n+1$

# Die triviale obere Schranke

- **Theorem 18.11**
  - Für alle  $f: X \times Y \rightarrow Z$  gilt  $D(f) \leq \log |X| + \log |Z|$
- **Beweis:**
  - Alice sendet  $x$  zu Bob. Bob berechnet  $f(x, y)$  und sendet dies zu Alice