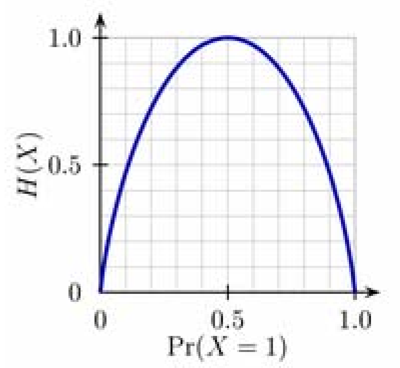
# Information und Kommunikation

Hartmut Klauck
Universität Frankfurt
SS 07
25.6.



# Nichtdeterminismus

- **Theorem 20.1**
  $N(f) = \log C^1(f)$
  $N(\neg f) = \log C^0(f)$
- Beweis:
  - Teil 2 folgt direkt aus 1
  - Teil 1:
  - $\leq$: Gegeben sei eine Überdeckung U der 1-Eingaben in $C^1(f)$ Rechtecke
  - Wir beschreiben ein nichtdet. Protokoll:
    - Alice rät den Namen eines 1-chromatischen Rechtecks, welches Eingaben (x,.) enthält
    - Alice sendet den Namen des Rechtecks
    - Bob akzeptiert, wenn das Rechteck Eingaben (.,y) enthält
  - Klar: f(x,y)=1, dann gibt es ein R in der Überdeckung, welches (x,y) enthält
  - Wenn R geraten wird, wird akzeptiert
  - Wenn f(x,y)=0, gibt es kein Rechteck in U, das (x,y) enthält
  - Das gesendete Rechteck enthält (x,.) , daher nicht (.,y), also wird nicht akzeptiert

# Nichtdeterminismus

- $\geq$: Wir zeigen $C^1(f) \leq 2^{N(f)}$
- Gegeben ist ein nichtdeterministisches Protokoll für f
- Jede Berechnung führt zu einem Rechteck in der Kommunikationsmatrix
- Die akzeptierenden Berechnungen entsprechen 1-chromatischen Rechtecken
- Es gibt damit soviel Rechtecke wie Nachrichten
- Die 1-chromatischen Rechtecke bilden eine (nicht disjunkte) Überdeckung von $M_f$

# Nichtdeterminismus

- Wir erhalten: $\log C^1(\neg EQ) \leq \log n+1$
  - n Rechtecke von der Form $\{x,y: x_i=0, y_i=1\}$ für alle i sowie n Rechtecke von der Form $\{x,y: x_i=1, y_i=0\}$ für alle i
- $D(EQ)=n+1$
- Es ist leicht zu sehen, dass $C^1(EQ)=2^n$
- D.h. $N(EQ)=n$
- **Theorem 20.2**
  - Es gibt eine Funktion f mit $D(f)=n+1$, $N(f)=\log n+1$, $N(\neg f)=n$
- D.h. „P≠NP“ und „NP≠co-NP“ in der Kommunikationskomplexität

# Nichtdeterminismus

- **Theorem 20.3**
  - $N(f) \geq \log D^1(f)$
- Beweis:
  - Gegeben sei ein nichtdeterministisches Protokoll P für f
  - Alice sendet für jede der Nachrichten von P ein Bit: „Nachricht ist vereinbar mit x oder nicht"
  - Bob kann so feststellen, ob es eine Berechnung gibt, die akzeptieren würde
  - D.h. $D^1(f) \leq 2^{N(f)}$
- Wir erhalten außerdem folgendes:
  - $D^1(f) \leq 2^{N(f)} \leq 2^{D(f)}$
  - D.h. Einweg Kommunikation ist maximal exponentiell schlechter als Zweiweg
  - Beispiel: $D^1(\text{Index})=n$, $D(\text{Index})=\log n+1$, $N(\text{Index}) \geq \log n$

# Überdeckungen

- Wir wollen jetzt unsere verschiedenen Maße miteinander vergleichen.
- Zuerst $C^P(f)$ und $D(f)$ [Anzahl der Blätter versus Kommunikation]
- **Theorem 20.4**
  - $\log_2(C^P(f)) \leq D(f) \leq 2\log_{3/2} C^P(f)$
- D.h. bis auf einen konstanten Faktor ist $\log C^p(f)$ gleich der det. Kommunikation

# Beweis 20.4

- Die untere Schranke ist trivial: die Anzahl der Blätter ist höchstens $2^{D(f)}$.
- Gegeben sei also ein beliebiges Protokoll für f mit $L=C^P(f)$ Blättern
- Wir müssen ein Protokoll mit möglichst geringer Tiefe angeben
- Dazu balancieren wir den Protokollbaum

# Beweis 20.4

- Zu einem internen Knoten v sei $L_v$ die Anzahl der Blätter unter v
- Es gibt immer einen Knoten v mit $L/3 \leq L_v \leq 2/3\ L$:
  - Wir beginnen an der Wurzel
  - An jedem Knoten u gehen wir zu seinem Nachfolger v mit $L_v \geq 2/3\ L$
  - Wenn es keinen solchen Nachfolger von u gibt, gilt $L_v \leq 2/3\ L$ für beide Nachfolger
  - Da für u gilt: $L_u \geq 2/3\ L$, muss für ein v gelten $L_v \geq L/3$
- $R_v$ sei das Rechteck der Eingaben, die v erreichen

# Beweis 20.4

- Wir geben jetzt das Protokoll an:
  - Alice und Bob testen, ob x,y in $R_v$ liegt (2 Bits Kommunikation)
  - Wenn ja: Rekursiv wird auf $R_v$ forgefahren
  - Wenn nein: Alice und Bob berechnen nun rekursiv die Funktion f‘, die erfüllt: $f'(x,y)=0$ für $x,y \in R_v$ und $f'(x,y)=f(x,y)$ sonst. Dazu wird v durch ein 0-Blatt ersetzt
- Im „ja“ Fall wird rekursiv ein Protokoll mit $L_v$ vielen Blättern verwendet
- Im „nein“ Fall ein Protokoll mit $L-L_v+1$ Blättern

# Beweis 20.4

- Das Protokoll ist korrekt, da die Eingaben in $R_v$ niemals das neue 0-Blatt erreichen
- $L/3 \leq L_v \leq 2/3\ L$, daher wir rekursiv in beiden Fällen ein Protokoll mit höchstens 2/3L Blättern gestartet
- Für die Tiefe D(L) eines Protokolls das wir zu einem gegebenen Protokoll mit L Blättern konstruieren gilt also $D(L) \leq 2+D(2L/3)$
- $D(1)=0$
- Daher folgt die Aussage.

# Überdeckungen

- Die Kommunikation ist also im wesentlichen über die Anzahl Blätter charakterisiert.
- $C^D(f)$ ist die Größe einer Partition der Kommunikationsmatrix in monochromatische Rechtecke
- Es ist nicht genau bekannt, wie sich $C^D(f)$ und $C^P(f)$ zueinander verhalten!
- Es gibt ein f, mit $D(f) \geq (2-o(1)) \log C^D(f)$

# Überdeckungen

- Statt $C^D(f)$ mit $C^P(f)$ zu vergleichen, betrachten wir $C(f)$ [disjunkte Überdeckungen]
- **Theorem 20.5**
  - $D(f) \leq O(\log C^0(f) \cdot \log C^1(f))$
- In anderen Worten:
  - $D(f) \leq O(N(f) \cdot N(\neg f))$
- Wenn also $N(f)$ und $N(\neg f)$ klein sind, ist auch $D(f)$ klein
- „P=NP$\cap$co-NP“ in der Kommunikation

# Beispiele

- Equality:
  - $N(EQ)=n$, $N(\neg EQ)=\log n+1$, $D(EQ)=n+1$
- Die Funktion ItEq: $f(x_1,\ldots,x_n,y_1,\ldots,y_n)=1$ wenn für alle i: $x_i \neq y_i$
  $x_i, y_i$ sind Strings der Länge m
- $N(ItEq)=O(n \log m)$:
  - Für alle i berechnet man $\neg EQ(x_i,y_i)$ nichtdeterministisch
- $N(\neg ItEq)=O(m)$:
  - Rate ein i von $1,\ldots,n$. Berechne $EQ(x_i,y_i)$ deterministisch
- $D(ItEq)=n \cdot m+1$

# Beweis 20.5

- Folgende Eigenschaft von Rechtecken ist wesentlich:
  - Für ein 0-chromatisches Rechteck $R=A\times B$ und ein 1-chromatisches Rechteck $R'=A'\times B'$ gilt:
    - $A\cap A'=\emptyset$ oder $B\cap B'=\emptyset$

# Beweis 20.5

- Gegeben sei also eine Überdeckung der 1-Eingaben mit $C^1(f)$ Rechtecken und eine Überdeckung der 0-Eingaben mit $C^0(f)$ Rechtecken
- Gesucht ist ein deterministisches Protokoll
- Betrachte ein 0-Rechteck R:
  - Für jedes 1-Rechteck R' gilt: R' s Zeilen sind disjunkt von R s Zeilen oder R' s Spalten sind disjunkt von R s Spalten
  - E.h. entweder sind die Hälfte alle 1-Rechtecke zu R spaltendisjunkt, oder die Hälfte sind zu R zeilendisjunkt

# Beweis 20.5

1. Zu Beginn sind alle Rechtecke „lebendig“
2. Alice sucht lebendige 1-Rechtecke, die x enthalten (d.h. eine Eingabe (x,.))
3. Existiert kein solches Rechteck, ist die Ausgabe 0
4. Wenn es ein lebendiges 0-Rechteck R gibt, das x enthält, und zu dem die Hälfte aller lebendigen 1-Rechtecke zeilendisjunkt sind, sendet Alice den Namen von R mit $\log(C^0(f))$ Bits
5. Sonst teilt Alice Bob mit, dass kein solches Rechteck existiert
6. Analog sucht Bob ein lebendiges 0-Rechteck, das y enthält, und mit mindestens der Hälfte aller lebendigen 1-Rechtecke spaltendisjunkt ist, und sendet dessen Namen
7. Wenn kein solches Rechteck existiert wird akzeptiert
8. Die Rechtecke, die mit den Zeilen (in 4) bzw. Spalten (in 6) des gesendeten R kompatibel sind (d.h. gemeinsame Zeilen bzw. Spalten haben) sind, bleiben lebendig
9. Gehe zu 2.

# Beweis 20.5

- Zur Kommunikation:
  - In jeder Runde wird die Anzahl lebendiger 1-Rechtecke halbiert
  - Daher gibt es höchstens log $C^1(f)$ viele Runden
  - In jeder Runde werden log $C^0(f)+O(1)$ Bits kommuniziert
  - Daher sind die gesamten Kosten log $C^1(f)(\log C^0(f) + O(1))$

# Beweis 20.5

- Zur Korrektheit:
  - Es wird verworfen, wenn keine 1-Rechtecke mehr übrig sind.
    - Alle entfernten 1-Rechtecke sind enthalten entweder x nicht oder y nicht
    - D.h. wenn alle 1-Rechtecke entfernt sind gilt auch f(x.y)=0
  - Es wird akzeptiert, wenn es kein 0-Rechteck mit den geforderten Eigenschaften gibt
    - D.h. alle 0-Rechtecke erfüllen:
      - enthält x nicht
      - oder enthält y nicht
      - oder enthält x,y und schneidet mehr als die Hälfte der 1-Rechtecke in Zeilen und mehr als die Hälfte aller 0-Rechtecke in Spalten
      - Der letzte Fall kann nicht eintreten
      - D.h. f(x,y)=1

# Überdeckungen

- **Korollar 20.6**
  - $D(f) \leq O((\log C(f))^2) \leq O((\log C^D(f))^2)$
- Über ItEq sieht man leicht, dass die erste Ungleichung nicht zu verbessern ist, d.h. $D(ItEQ)=n^2$, $\log C(ItEQ)=O(n)$