Kommunikationskomplexität

Danny Hucke

Universität Siegen

Wintersemester 2018/2019

Grundprinzip

Gegeben: Funktion $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$

- Zwei Parteien (**Alice** und **Bob**) wollen gemeinsam f berechnen
- Alice bekommt $x \in \{0,1\}^n$ (unbekannt für Bob)
- Bob bekommt $y \in \{0,1\}^n$ (unbekannt für Alice)
- Beide dürfen kommunizieren, d.h. Nachrichten hin und her senden
- **Ziel:** Ausgabe von $f(x,y)$

Kommunikationskomplexität von f:

$\to$ Minimale erforderliche Kommunikation um f zu berechnen.

Anmerkung: Ressourcen für Berechnungen der beiden werden ignoriert

Anwendungen von Kommunikationskomplexität

Kommunikationskomplexität wurde eingeführt in

A. C. Yao, "Some Complexity Questions Related to Distributed Computing", Proc. of STOC, 1979

Kommunikationskomplexität wird für **untere Schranken** genutzt.

Einige Anwendungen:

- Anzahl der Zustände von endlichen Automaten
- Zeit/Platz Tradeoffs von Turingmaschinen
- Platzbedarf von Streaming-Algorithmen
- Größe von Entscheidungsbäumen und OBDDs [1]
- Fläche/Zeit Tradeoffs im VLSI Modell [2]

[1] Ordered Binary Decision Diagram

[2] very-large-scale integration

Kommunikationsprotokoll

Definition 1 (Deterministisches Kommunikationsprotokoll)

Ein (r-Runden) Kommunikationsprotokoll ist ein Tupel $P = (A, B)$ mit

- A (Alice) ist eine Funktion $A : \bigcup_{i=0}^{(r-1)/2} (\{0,1\}^*)^{1+2i} \to \{0,1\}^*$ mit der Einschränkung dass $|A(w_1, \ldots, w_{1+2i})| \geq 1$ für $i > 0$,
- B (Bob) ist eine Funktion $B : \bigcup_{i=0}^{(r-1)/2} (\{0,1\}^*)^{2+2i} \to \{0,1\}^+$.

Die Kommunikation $s_P(x, y)$ bzgl. P bei Eingabe $x, y \in \{0,1\}^*$ ist

$$s_P(x, y) = (m_1, m_2, \ldots, m_r, m_{r+1})$$

wobei für $i = 0, \ldots, (r-1)/2$ gilt:

- $m_{1+2i} = A(x, m_1, m_2, \ldots, m_{2i})$
- $m_{2+2i} = B(y, m_1, m_2, \ldots, m_{1+2i})$

Die m_i's sind Nachrichten, die Länge $|m_i|$ ist die Anzahl der Bits von m_i

Kommunikationsprotokoll

Sei $s_P(x, y) = (m_1, m_2, \ldots, m_r, m_{r+1})$ für ein Protokoll P

Die Ausgabe $o_P(x, y)$ von P für $x, y \in \{0,1\}^*$ ist

$$o_P(x, y) = m_{r+1}$$

Definition 2

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ eine Funktion auf Eingaben der Länge n.

P berechnet f falls für alle $x, y \in \{0,1\}^n$ gilt:

$$o_P(x, y) = f(x, y)$$

Wir schreiben auch f_P^n für die von P berechnete Funktion, die auf Eingaben der Länge n definiert ist ($f_P^n(x, y) = o_P(x, y)$).

Kommunikationskomplexität

Sei $s_P(x, y) = (m_1, m_2, \ldots, m_r, m_{r+1})$ für ein Protokoll P

Die Länge von $s_P(x, y)$ ist $|s_P(x, y)| = \sum_{i=1}^{r+1} |m_i|$

Definition 3 (Kommunikationskomplexität)

Sei P ein Kommunikationsprotokoll.

Die (worst-case) Kommunikationskomplexität von P für Eingaben der Länge n ist

$$D_n(P) = \max_{x,y \in \{0,1\}^n} |s_P(x, y)|.$$

Sei $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ eine Funktion auf Eingaben der Länge n.

Die (deterministische) Kommunikationskomplexität von f ist

$$D(f) = \min_{P : f_P^n = f} D_n(P).$$

Anmerkungen

- Die erste Nachricht von A kann leer sein, ansonsten müssen alle Nachrichten mindestens Länge 1 haben.
- A und B sind beliebig komplex (möglicherweise nicht berechenbar).
- Die Länge der Eingaben n ist der Parameter in dem wir die Kommunikationskomplexität messen.
- Statt Funktionen $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ kann man auch Funktionen $f : X \times Y \to Z$ für endliche Mengen X, Y, Z betrachten und die Kommunikationskomplexität in $|X|$, $|Y|$ und $|Z|$ messen.
- Man sagt r-Runden Protokoll (statt $r+1$) da m_{r+1} die Ausgabe ist.
- Da Bob immer die letzte Nachricht schickt, ist r immer ungerade.
- Die Kommunikationskomplexität $D(f)$ wird für Protokolle mit beliebiger Rundenzahl bestimmt. Es gibt auch eingeschränkte Versionen (z.B. $r = 1$: *One-way communication complexity*).

Einstiegsbeispiel

Für $x = x_1 x_2 \dots x_n \in \{0,1\}^n$ und $y = y_1 y_2 \dots y_n \in \{0,1\}^n$ sei

$$\mathrm{PAR}(x,y) = \begin{cases} 1 & \text{falls } \sum_{i=1}^n x_i \equiv \sum_{i=1}^n y_i \pmod 2 \\ 0 & \text{sonst} \end{cases}$$

$D(\mathrm{PAR}) = 2$:

- 1-Runden Protokoll $P = (A,B)$ mit $f_P^n = \mathrm{PAR}$ und $D_n(P) = 2$:
  - $A(x) = (\sum_{i=1}^n x_i) \mod 2$
  - $B(y,m) = \begin{cases} 1 & \text{falls } m = (\sum_{i=1}^n y_i) \mod 2 \\ 0 & \text{sonst} \end{cases}$
- $D(\mathrm{PAR}) \geq 2$, da PAR echt von x abhängt $(A(x) \neq \varepsilon)$.

Einige wichtige Funktionen

Seien $x = x_1 x_2 \dots x_n \in \{0,1\}^n$ und $y = y_1 y_2 \dots y_n \in \{0,1\}^n$.

- **Gleichheit**: $\mathrm{EQ}(x,y) = \begin{cases} 1 & \text{falls } x = y \\ 0 & \text{sonst} \end{cases}$
- **Skalarprodukt**: $\mathrm{IP}(x,y) = \left(\sum_{i=1}^{n} x_i \cdot y_i\right) \mod 2$
- **Größer-als**: $\mathrm{GT}(x,y) = \begin{cases} 1 & \text{falls } x > y \\ 0 & \text{sonst} \end{cases}$

  x und y werden als binäre Darstellungen der Zahlen aus $0, \dots, 2^n - 1$ interpretiert
- **Disjunktheit**: $\mathrm{DISJ}(x,y) = \begin{cases} 0 & \text{falls } \exists i \in \{1,\dots,n\} : \ x_i = y_i = 1 \\ 1 & \text{sonst} \end{cases}$

  x, y werden als Teilmengen von $\{1, \dots, n\}$ gedeutet: i-ter Eintrag zeigt ob i enthalten ist

Allgemeine obere Schranke

Bei der trivialen oberen Schranke schickt Alice ihren Input an Bob:

Theorem 4 (Triviales Protokoll)

Für jede Funktion $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *gilt* $D(f) \leq n+1$.

Beweis:

Sei $P = (A, B)$ das 1-Runden Protokoll mit $f_P^n = f$ und $D_n(P) = n+1$:

- $A(x) = x$,
- $B(y, m) = f(m, y)$. □

Wir werden sehen, dass es für viele Funktionen nicht besser geht.

Problem: Wie zeigt man das?

Monochromatische Rechtecke

Definition 5 (Rechteck und monochromatische Mengen)

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ und $A, B \subseteq \{0,1\}^n$.

Dann wird $R = A \times B$ als Rechteck bezeichnet.

Eine Menge $M \subseteq \{0,1\}^n \times \{0,1\}^n$ ist f-monochromatisch (einfarbig) falls

$$\forall (x,y),(x',y') \in M : \quad f(x,y) = f(x',y')$$

- f-monochromatisch: Für alle $(x,y) \in M$ ist $f(x,y)$ identisch
- $\{0,1\}^n \times \{0,1\}^n$ wird als Matrix mit den f-Werten interpretiert
- A ist dann eine Teilmenge der Zeilen, B eine Teilmenge der Spalten

Beispiel: $A = 0\{0,1\}^{n-1}$, $B = 1\{0,1\}^{n-1}$

$\longrightarrow$ $R = A \times B$ ist ein EQ-monochromatisches Rechteck

Monochromatische Rechtecke

Lemma 6

Sei P ein r-Runden Protokoll und $(m_1, \ldots, m_r, m_{r+1}) \in (\{0,1\}^*)^{r+1}$.
Dann ist

$$M = \{(x, y) \in \{0,1\}^n \times \{0,1\}^n \mid s_P(x, y) = (m_1, \ldots, m_r, m_{r+1})\}$$

ein f_P^n-monochromatisches Rechteck (für jedes n).

Beweis:

M ist monochromatisch, da für alle $(x, y) \in M$ gilt, dass $f_P^n(x, y) = m_{r+1}$.

Es genügt also zu zeigen, dass M ein Rechteck ist.

Monochromatische Rechtecke

Wir zeigen per **Induktion** über i, dass die Menge M_i der Inputs, deren Kommunikation bzgl. P mit $(m_1, \ldots, m_i)$ beginnt, ein Rechteck ist.

IA $(i = 0)$: $M_0 = \{0,1\}^n \times \{0,1\}^n$

Jede Kommunikation beginnt mit der leeren Sequenz.

IH: Sei $M_i = A \times B$ ein Rechteck.

IS: O.B.d.A. nehmen wir an, dass Alice die Nachricht m_{i+1} sendet.

Sei $A' = \{x \in \{0,1\}^n \mid A(x, m_1, \ldots, m_i) = m_{i+1}\} \subseteq A$.

Dann ist $M_{i+1} = A' \times B$ ein Rechteck.

Falls Bob die Nachricht m_{i+1} sendet ist die Argumentation analog. □

Partitionierung in monochromatische Rechtecke

Zwei Rechtecke R und R' sind disjunkt, falls kein (x, y) existiert, so dass

$$(x, y) \in R \text{ und } (x, y) \in R'.$$

Definition 7 (Partitionierung in f-monochromatische Rechtecke)

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

Wir nennen paarweise disjunkte, f-monochromatische Rechtecke $R_1, \ldots, R_m$ eine f-Partitionierung der Größe m, falls

$$\bigcup_{i=1}^{m} R_i = \{0,1\}^n \times \{0,1\}^n.$$

Sei $C(f)$ die minimale Größe einer f-Partitionierung.

Kommunikationskomplexität $\leftrightarrow$ Partitionierung

Lemma 8

Für jede Funktion $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *gilt*

$$D(f) \geq \log_2 C(f).$$

Beweis:

Sei P ein optimales Protokoll mit $f_P^n = f$ und $D_n(P) = D(f)$.

Nach Lemma 6 liefert P eine f-Partitionierung:

(x, y), (x', y') gehören zum gleichen Rechteck, falls $s_P(x, y) = s_P(x', y')$.

Die Größe dieser f-Partitionierung ist höchstens $2^{D_n(P)}$.

(Die Anzahl möglicher, verschiedenerer Kommunikationen bzgl. P)

Daher folgt, dass $C(f) \leq 2^{D_n(P)} = 2^{D(f)}$. □

Fooling sets

Fooling sets [1] eignen sich um untere Schranken für $C(f)$ zu zeigen.

Definition 9 (Fooling set)

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

Eine Menge $F \subseteq \{0,1\}^n \times \{0,1\}^n$ heißt fooling set bzgl. f, falls

- F ist f-monochromatisch,
- Für alle $(x,y),(x',y') \in F$ mit $(x,y) \neq (x',y')$ gilt

$$f(x,y') \neq f(x,y) \text{ oder } f(x',y) \neq f(x,y).$$

- $f(x,y)$ kann man durch $f(x',y')$ ersetzen (F ist f-monochromatisch)
- Ein fooling set ist per Definition **kein** Rechteck

[1] R. J. Lipton, and R. Sedgewick, "Lower Bounds for VLSI", Proc. of STOC, 1981

Kommunikationskomplexität $\leftrightarrow$ fooling sets

Lemma 10

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *und* F *ein fooling set bzgl.* f*. Es gilt*

$$D(f) \geq \log_2 |F|.$$

Beweis:

Nach Lemma 8 genügt es $C(f) \geq |F|$ zu zeigen.

Wir führen den Beweis **indirekt**,

sei also $R_1, \ldots, R_m$ eine f-Partitionierung mit $m < |F|$

$\to (x,y),(x',y') \in F$ mit $(x,y) \neq (x',y')$, so dass $(x,y),(x',y') \in R_i$

Da R_i ein Rechteck ist, gilt auch $(x,y'),(x',y) \in R_i$.

Es gilt aber $f(x,y) \neq f(x,y')$ oder $f(x,y) \neq f(x',y)$ (F ist ein fooling set)

$\to$ Widerspruch dazu, dass R_i f-monochromatisch ist. $\square$

Untere Schranken für EQ, GT und DISJ

Theorem 11

1. $D(\mathrm{EQ}) \geq n$
2. $D(\mathrm{GT}) \geq n$
3. $D(\mathrm{DISJ}) \geq n$

Beweis:

Nach Lemma 10 genügt es fooling sets der Größe 2^n anzugeben:

1. Fooling set bzgl. EQ: $\{(x, x) \mid x \in \{0,1\}^n\}$
2. Fooling set bzgl. GT: $\{(x, x) \mid x \in \{0,1\}^n\}$
3. Fooling set bzgl. DISJ: $\{(x, \bar{x}) \mid x \in \{0,1\}^n\}$

   (Für $x = x_1 \dots x_n$ ist $\bar{x}$ der invertierte String zu x, d.h. aus x_i wird $1 - x_i$)

Die Funktion INDEX

Eine weitere interessante Funktion ist
$\mathrm{INDEX} : \{0,1\}^n \times \{1,\ldots,n\} \to \{0,1\}$ mit $\mathrm{INDEX}(x_1 x_2 \ldots x_n, i) = x_i$.

Alternative Definition um Eingaben aus $\{0,1\}^n \times \{0,1\}^n$ zu haben:

Sei $\mathrm{INDEX}' : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ mit

$$\mathrm{INDEX}'(x_1 x_2 \ldots x_n, y) = \begin{cases} x_i & \text{falls } y = 0^{i-1}10^{n-i} \\ 0 & \text{sonst} \end{cases}$$

Theorem 12

$D(\mathrm{INDEX}) = D(\mathrm{INDEX}') \in \Theta(\log n)$.

Beweis:

$D(\mathrm{INDEX}) = D(\mathrm{INDEX}')$ gilt, da sich jedes Protokoll für INDEX in ein gleich langes Protokoll für INDEX' umformen lässt und umgekehrt.

Die Funktion INDEX

$D(\mathrm{INDEX}) \in O(\log n)$:

Bob sendet die binäre Repräsentation von i an Alice.

Alice sendet x_i an Bob, welches dieser ausgibt.

$D(\mathrm{INDEX}') \in \Omega(\log n)$:

Fooling set bzgl. INDEX' der Größe n:

$$F = \{(0^{i-1}10^{n-i}, 0^{i-1}10^{n-i}) \mid i \in \{1, \ldots, n\}\}$$

F ist monochromatisch, da für alle $(x, y) \in F$ gilt $\mathrm{INDEX}'(x, y) = 1$.

Für $(x, y), (x', y') \in F$ gilt $\mathrm{INDEX}'(x', y) = \mathrm{INDEX}'(x, y') = 0$.

Ein algebraischer Ansatz für untere Schranken

Definition 13 (f-Matrix)

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

Die f-Matrix M_f ist eine $2^n \times 2^n$ Matrix mit Einträgen 0 oder 1, deren Zeilen und Spalten zu Elementen aus $\{0,1\}^n$ korrespondieren.

Für Zeile $x \in \{0,1\}^n$ und Spalte $y \in \{0,1\}^n$ ist $M_f[x,y] = f(x,y)$.

(Die Reihenfolge der Elemente aus $\{0,1\}^n$ ist beliebig aber gleich für Zeilen und Spalten.)

$\to$ Zusammenhang zwischen $D(f)$ und dem Rang von M_f [1]

Beispiel:

Die Matrix M_{EQ} ist die Identitätsmatrix.

[1] K. Mehlhorn und E. Schmidt, "Las-Vegas is better than Determinism in VLSI and Distributed Computing", Proc. of STOC, 1982

Kommunikationskomplexität $\leftrightarrow$ Matrix M_f

Lemma 14

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. *Es gilt* $D(f) \geq \log_2 \operatorname{rank}(M_f)$.

Beweis:

Nach Lemma 8 genügt es $C(f) \geq \operatorname{rank}(M_f)$ zu zeigen.

Sei $R_1, \ldots, R_m$ eine kleinste f-Partitionierung.

O.B.d.A. seien $R_1, \ldots, R_t$ 1-monochromatische Rechtecke.

Wir zeigen, dass bereits $t \geq \operatorname{rank}(M_f)$.

Sei M_i $(1 \leq i \leq t)$ die Matrix, so dass $M_i[x,y] = \begin{cases} 1 & \text{falls } (x,y) \in R_i \\ 0 & \text{sonst.} \end{cases}$

Es gilt $M_f = \sum_{i=1}^{t} M_i$.

Wegen der Subadditivität vom Rang gilt $\operatorname{rank}(M_f) \leq \sum_{i=1}^{t} \operatorname{rank}(M_i)$.

Offensichtlich gilt für $1 \leq i \leq t$, dass $\operatorname{rank}(M_i) = 1$. □

Untere Schranken für EQ und IP

Theorem 15

1. $D(\mathrm{EQ}) \geq n$
2. $D(\mathrm{IP}) \geq n$

Beweis:

Wir wenden Lemma 14 an.

1. Die Identitätsmatrix M_{EQ} hat Rang $\mathrm{rank}(M_{\mathrm{EQ}}) = 2^n$.
2. Die Matrix M_{IP} ist die sogenannte Hadamard-Matrix vom Grad 2^n (wobei die Einträge -1 durch 0 ersetzt werden).

   Es gilt $\mathrm{rank}(M_{\mathrm{IP}}) = 2^n$. □

Anmerkung: Punkt 2 wird in der Übung nochmal ausführlich bewiesen.

Obere Schranke mit Hilfe einer f-Partitionierung

- Kann man f-Partitionierungen auch für obere Schranken verwenden?
- Wie nah kommt man an $D(f)$ heran?

Man kann mit Hilfe von f-Partitionierungen recht nah an die untere Schranke heran:

Theorem 16

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. *Es gilt* $D(f) \leq (\log_2 C(f))^2$.

Geht es besser mit Hilfe von f-Partitionierungen?

Leider lässt sich $D(f)$ nicht exakt durch f-Partitionierungen berechnen[1].

Theorem 17

Es existiert eine Funktion f, so dass $D(f) \geq (\log_2 C(f))^{2-o(1)}$.

Anmerkung:

Beide Theoreme gelten auch, falls die Rechtecke nicht disjunkt sind.

Wie sieht es mit oberen Schranken m.H. von $\mathrm{rank}(M_f)$ aus?

[1] A. Ambainis, M. Kokainis und R. Kothari, "Nearly optimal separations between communication (or query) complexity and partitions", Proc. of CCC, 2016

Obere Schranke mit Hilfe von $\operatorname{rank}(M_f)$

Aus $C(f) \geq \operatorname{rank}(M_f)$ und Theorem 17 folgt direkt:

Theorem 18

Es existiert eine Funktion f, so dass $D(f) \geq (\log_2 \operatorname{rank}(M_f))^{2-o(1)}$.

Eine polylogarithmische obere Schranke gibt es derzeit nicht.

Deren Existenz wird in der Log-Rank Vermutung[1] vorhergesagt:

Vermutung 19 (Log-Rank Vermutung)

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Es gilt $D(f) \leq (\log_2 \operatorname{rank}(M_f))^{O(1)}$.

[1] L. Lovasz und M. E. Saks, "Lattices, Möbius functions and communication complexity", Proc. of FOCS, 1988

Obere Schranke mit Hilfe von $\mathrm{rank}(M_f)$

Lange Zeit war die beste obere Schranke $D(f) \leq \log_2(4/3) \cdot \mathrm{rank}(M_f)$.

Mittlerweile verbessert auf[1]:

Theorem 20

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. *Es gilt*

$$D(f) \leq \sqrt{\mathrm{rank}(M_f)} \cdot \log_2 \mathrm{rank}(M_f).$$

Direkter Beweis von Rothvoß: https://arxiv.org/pdf/1409.6366.pdf

Weit weg von der unteren Schranke (und der Log-Rank Vermutung)!

[1] S. Lovett, "Communication is bounded by root of rank", Proc. of STOC, 2014

Erste Anwendung - Automatentheorie

Lemma 21

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *und* $L \subseteq \{0,1\}^*$. *Sei* M *ein deterministischer Automat mit Zustandsmenge* Q, *der* L *erkennt. Falls*

$$L \cap \{0,1\}^{2n} = \{xy \in \{0,1\}^{2n} \mid |x| = |y| = n, \ \ f(x,y) = 1\},$$

dann gilt $D(f) \leq \log_2 |Q| + 1$.

Beweis:

Protokoll für f basierend auf dem Automaten M:

Alice simuliert M bei Eingabe x. Sei $q \in Q$ der erreichte Zustand.
Mit $\leq \log_2 |Q| + 1$ Bits sendet Alice q an Bob.

Bob simuliert M beginnend in q bei Eingabe y.
Er sendet 1, falls M akzeptiert und 0 sonst. □

Größe von Automaten $\leftrightarrow$ Kommunikationskomplexität

Beispiel 1: $L = \{xx \mid x \in \{0,1\}^*\}$ ist nicht regulär.

Sei $L_n = \{xx \mid x \in \{0,1\}^n\}$.

Es gilt $L_n = \{xy \in \{0,1\}^{2n} \mid |x| = |y| = n, \ \mathrm{EQ}(x,y) = 1\}$.

L_n ist endlich, also gibt es einen DFA für L_n.

Sei M ein solcher DFA mit Zustandmenge Q.

Aus Lemma 21 folgt $|Q| \geq 2^{D(\mathrm{EQ})-1}$.

Zusammen mit $D(\mathrm{EQ}) = n+1$ (Übung) folgt $|Q| \geq 2^n$.

Für $L = \{xx \mid x \in \{0,1\}^*\}$ gilt $L \cap \{0,1\}^{2n} = L_n$ für alle $n \in \mathbb{N}$.

$\rightarrow$ L wird von keinem endlichen Automaten erkannt.

Größe von Automaten $\leftrightarrow$ Kommunikationskomplexität

Beispiel 2: Exponentieller Blow-up vom NFA zum DFA.

Sei $K_n = \{xy \in \{0,1\}^{2n} \mid |x| = |y| = n, \ \ x \neq y\}$. Es gilt:

1. Es existiert ein NFA mit $O(n)$ Zuständen (zum selbst überlegen).
2. Jeder DFA hat mindestens 2^n Zustände.

Sei $\mathrm{NEQ}(x,y) = 1 - \mathrm{EQ}(x,y)$.

Es ist leicht zu sehen, dass $D(\mathrm{NEQ}) = D(\mathrm{EQ})$.

Ausserdem ist $K_n = \{xy \in \{0,1\}^{2n} \mid |x| = |y| = n, \ \ \mathrm{NEQ}(x,y) = 1\}$.

$\rightarrow$ Jeder DFA für K_n hat $2^{D(\mathrm{EQ})-1} = 2^n$ Zustände.

Die Potenzmengenkonstruktion ist (asymptotisch) optimal.

Zweite Anwendung - Turing-Maschinen

Mit Kommunikationskomplexität kann man Platz/Zeit Tradeoffs zeigen.

Hier betrachten wir ein konkretes Beispiel:
$L_{\text{pal}} = \{w \in \{0,1\}^* \mid w = w^{\text{rev}}\}$

Theorem 22

Sei M eine Turing-Maschine für L_{pal}, die t-zeitbeschränkt und s-platzbeschränkt ist. Dann gilt:

$$t(m) \cdot s(m) \in \Omega(m^2)$$

Beweis:

Ein Protokoll P für EQ:

Strategie: Simulation von M bei Eingabe $x0^n y^{\text{rev}}$ (der Länge $m = 3n$).

M akzeptiert genau dann, wenn $\text{EQ}(x, y) = 1$.

Zeit/Platz Tradeoffs von Turing-Maschinen

Problem: Alice kennt y nicht, Bob kennt x nicht.

Lösung:

- Alice simuliert M wenn der Lesekopf im $x0^n$-Bereich ist.
- Geht der Lesekopf in den y-Bereich, übergibt sie an Bob.
- Analog simuliert Bob während die Maschine in der $0^n y$-Region ist.
- Geht der Lesekopf in den x-Bereich, übergibt er an Alice.

Bei jedem Wechsel müssen $O(s(m))$ Bits transferiert werden.

Vor jedem Wechsel wird der 0^n-Bereich durchquert.
$\rightarrow$ Die Anzahl der Wechsel ist höchstens $t(m)/n$.

Es gilt $D_n(P) \in O(s(m) \cdot t(n)/n)$ und $D_n(P) \geq n$ wegen $D(\mathrm{EQ}) \geq n$.
$\rightarrow$ $t(m) \cdot s(m) \in \Omega(n^2) = \Omega(m^2)$ □

Zeit/Platz Tradeoffs von Turing-Maschinen

Anmerkungen:

Theorem 22 ist scharf.

Eine Turing-Maschine, die sowohl $O(n)$ Zeit als auch Platz benötigt,

1. kopiert die erste Hälfte auf das Band und
2. prüft ob die zweite Hälfte zur ersten passt.

Eine Turing-Maschine, die $O(n^2)$ Zeit und $O(\log n)$ Platz benötigt,

1. prüft für das i-te Bit, ob es mit dem $(n-i)$-ten Bit übereinstimmt
2. bis i in der Mitte des Wortes ankommt.

Randomisierte Kommunikationskomplexität

Definition 23 (Randomisiertes Kommunikationsprotokoll)

Ein randomisiertes (r-Runden) Kommunikationsprotokoll ist ein Tupel $P = (A, B)$ mit

- A (Alice) ist eine Funktion $A : \bigcup_{i=0}^{(r-1)/2} (\{0,1\}^*)^{2+2i} \to \{0,1\}^*$ mit der Einschränkung dass $|A(w_1, \ldots, w_{2+2i})| \geq 1$ für $i > 0$,
- B (Bob) ist eine Funktion $B : \bigcup_{i=0}^{(r-1)/2} (\{0,1\}^*)^{3+2i} \to \{0,1\}^+$.

Man unterscheidet public-coins und private-coins:

- Im public-coin Modell teilen sich Alice und Bob einen Randomstring.
- Im private-coin Modell haben Alice und Bob eigene Randomstrings.

Public-coin Modell

Wir beginnen mit dem **public-coin Modell**.

Die Kommunikation $s_P(x, y, z)$ bzgl. P bei Eingabe $x, y \in \{0,1\}^*$ und Zufallsstring $z \in \{0,1\}^*$ ist

$$s_P(x, y, z) = (m_1, m_2, \ldots, m_r, m_{r+1})$$

wobei für $i = 0, \ldots, (r-1)/2$ gilt:

- $m_{1+2i} = A(x, z, m_1, m_2, \ldots, m_{2i})$
- $m_{2+2i} = B(y, z, m_1, m_2, \ldots, m_{1+2i})$

Die Ausgabe $o_P(x, y, z)$ von P bzgl. $x, y, z \in \{0,1\}^*$ ist m_{r+1}.

Public-coin Modell

Sei m die maximale Anzahl von Zufallsbits, die in dem randomisierten Protokoll P von Alice und Bob genutzt werden.

Die Wahrscheinlichkeit, dass P für x, y den Wert $a \in \{0,1\}^*$ ausgibt ist:

$$\text{Prob}[o_P(x,y) = a] = \frac{|\{z \in \{0,1\}^m \mid o_P(x,y,z) = a\}|}{2^m}$$

Definition 24

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

P berechnet f falls für alle $x, y \in \{0,1\}^n$ gilt:

$$\text{Prob}[o_P(x,y) = f(x,y)] \geq 2/3$$

Randomisierte Kommunikationskomplexität

Definition 25 (Kommunikationskomplexität im public-coin Modell)

Sei P ein randomisiertes Kommunikationsprotokoll.

Die randomisierte (worst-case) Kommunikationskomplexität von P ist

$$R_n^{\mathrm{pu}}(P) = \max_{x,y\in\{0,1\}^n,\, z\in\{0,1\}^m} |s_P(x,y,z)|.$$

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ eine Funktion auf Eingaben der Länge n.

Die randomisierte Kommunikationskomplexität von f ist

$$R^{\mathrm{pu}}(f) = \min_{P\text{: } P \text{ berechnet } f} R_n^{\mathrm{pu}}(P).$$

Private-coin Modell

Betrachten wir nun das **private-coin Modell**.

Die Kommunikation $s_P(x, y, z_A, z_B)$ bzgl. P bei Eingabe $x, y \in \{0,1\}^*$ und Zufallsstrings $z_A, z_B \in \{0,1\}^*$ ist

$$s_P(x, y, z_A, z_B) = (m_1, m_2, \ldots, m_r, m_{r+1})$$

wobei für $i = 0, \ldots, (r-1)/2$ gilt:

- $m_{1+2i} = A(x, z_A, m_1, m_2, \ldots, m_{2i})$
- $m_{2+2i} = B(y, z_B, m_1, m_2, \ldots, m_{1+2i})$

Die Ausgabe $o_P(x, y, z_A, z_B)$ von P bzgl. $x, y, z_A, z_B \in \{0,1\}^*$ ist m_{r+1}.

Private-coin Modell

Sei m die maximale Anzahl von Zufallsbits, die in dem randomisierten Protokoll P von Alice und Bob genutzt werden.

Die Wahrscheinlichkeit, dass P für x, y den Wert $a \in \{0,1\}^*$ ausgibt ist:

$$\text{Prob}[o_P(x,y) = a] = \frac{|\{(z_A, z_B) \in (\{0,1\}^m)^2 \mid o_P(x,y,z_A,z_B) = a\}|}{2^{2m}}$$

Definition 26

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

P berechnet f falls für alle $x, y \in \{0,1\}^n$ gilt:

$$\text{Prob}[o_P(x,y) = f(x,y)] \geq 2/3$$

Randomisierte Kommunikationskomplexität

Definition 27 (Randomisierte Kommunikationskomplexität)

Sei P ein randomisiertes Kommunikationsprotokoll.

Die randomisierte (worst-case) Kommunikationskomplexität von P ist

$$R_n^{\mathrm{pr}}(P) = \max_{x,y\in\{0,1\}^n,\ z_A,z_B\in\{0,1\}^m} |s_P(x,y,z_A,z_B)|.$$

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ eine Funktion auf Eingaben der Länge n.

Die randomisierte Kommunikationskomplexität von f ist

$$R^{\mathrm{pr}}(f) = \min_{P\text{: } P \text{ berechnet } f} R_n^{\mathrm{pr}}(P).$$

Die Wahl der Wahrscheinlichkeit ...

Wir fordern eine Erfolgswahrscheinlichkeit von $p \geq 2/3$.

Man hätte aber auch jedes feste p mit $1/2 < p < 1$ wählen können.

(Wir nennen $1 - p$ die Fehlerwahrscheinlichkeit)

Probability amplification

Sei P ein randomisiertes Protokoll mit Fehlerwahrscheinlichkeit $\varepsilon < 1/2$.

Sei P'_k das Protokoll, welches

- k-mal das Protokoll P ausführt (mit verschiedenen Zufallsstrings) und
- als Ausgabe den häufigsten Output der k Ausführungen nimmt.

Es gilt $R^{\mathrm{x}}_n(P'_k) \in O(R^{\mathrm{x}}_n(P))$, da k nicht von n abhängt $(\mathrm{x} \in \{\mathrm{pu}, \mathrm{pr}\})$.

Mit Hilfe der Chernoff-Ungleichung gilt:

Die Fehlerwahrscheinlichkeit von P'_k ist $\leq 2^{-\Omega\left(k(1/2-\varepsilon)^2\right)}$.

Public-coin $\leftrightarrow$ Private-coin

Theorem 28 (Public-coin versus private-coin)

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. *Es gilt*

$$R^{\mathrm{pu}}(f) \leq R^{\mathrm{pr}}(f) \in O(R^{\mathrm{pu}}(f) + \log n).$$

Beweis:

Wir zeigen zuerst $R^{\mathrm{pu}}(f) \leq R^{\mathrm{pr}}(f)$.

Sei P **ein Protokoll für** f **im private-coin Modell** und sei m die maximale Länge der Zufallsstrings, die Alice und Bob benutzen.

Wir konstruieren ein **Protokoll** P' **für** f **im public-coin Modell**:

- Beide Spieler haben einen Zufallsstring $z = z_1 \dots z_{2m} \in \{0,1\}^{2m}$.
- Alice simuliert das Protokoll P mit dem Zufallsstring $z_1 \dots z_m$.
- Bob simuliert das Protokoll P mit dem Zufallsstring $z_{m+1} \dots z_{2m}$.

Die Behauptung folgt direkt.

Public-coin $\leftrightarrow$ Private-coin

Jetzt zeigen wir $R^{\mathrm{pr}}(f) \in O(R^{\mathrm{pu}}(f) + \log n)$.

Sei P **ein Protokoll für** f **im public-coin Modell** mit Fehlerwahrscheinlichkeit $\varepsilon < 1/4$ (probability amplification).
Sei m die maximale Länge eines Zufallsstrings, der in P genutzt wird (o.B.d.A. sei $m \geq n$, andernfalls setzen wir $m = n$).

Idee:

Wir zeigen die Existenz einer Menge $R = \{r_1, \ldots, r_t\} \subseteq \{0,1\}^m$ mit $|R| \in O(n)$, die wir in einem **private-coin Protokoll** P' **für** f verwenden:

- Alice nutzt ihren privaten Zufallsstring und wählt zufällig ein $r_i \in R$.
- Alice sendet i an Bob ($O(\log n)$ Bits), so dass dieser r_i kennt.
- Beide nutzten r_i als Zufallsstring um P zu simulieren.
- Die Fehlerwahrscheinlichkeit von P' ist $2\varepsilon < 1/2$.

Public-coin $\leftrightarrow$ Private-coin

Die **probabilistische Methode** wird die Existenz von R zeigen:

Seien $r_1, \ldots, r_t$ unabhängig, zufällig gewählte Strings aus $\{0,1\}^m$.

Für eine feste Eingabe $(x, y) \in \{0,1\}^n \times \{0,1\}^n$ gilt:

Die Wahrscheinlichkeit (bzgl. r_i) ist $\geq 1 - \varepsilon$, dass $o_P(x, y, r_i) = f(x, y)$.
Sei $\ell_{x,y} \leq t$ die Anzahl der r_i, so dass $o_P(x, y, r_i) \neq f(x, y)$.
Mit Hilfe der Chernoff-Ungleichung folgt:

$$\operatorname*{Prob}_{r_1,\ldots,r_t}[\ell_{x,y} \geq (1 - 2\varepsilon)t] \leq 2^{-\Omega(\varepsilon^2 t)}$$

Wir wählen $t \in O(n/\varepsilon^2)$, so dass $\operatorname*{Prob}_{r_1,\ldots,r_t}[\ell_{x,y} \geq (1 - 2\varepsilon)t] < 2^{-2n}$.

Public-coin $\leftrightarrow$ Private-coin

Nun betrachten wir alle möglichen Eingaben $(x, y) \in \{0,1\}^n \times \{0,1\}^n$.

Aus Boole's Ungleichung (auch bekannt als *union bound*) folgt:

$$\operatorname*{Prob}_{r_1,\ldots,r_t}[\exists (x,y) : \ell_{x,y} \geq (1-2\varepsilon)t] \leq \sum_{(x,y)} \operatorname*{Prob}_{r_1,\ldots,r_t}[\ell_{x,y} \geq (1-2\varepsilon)t]$$

Wegen $\operatorname*{Prob}_{r_1,\ldots,r_t}[\ell_{x,y} \geq (1-2\varepsilon)t] < 2^{-2n}$ gilt

$$\sum_{(x,y)} \operatorname*{Prob}_{r_1,\ldots,r_t}[\ell_{x,y} \geq (1-2\varepsilon)t] < \sum_{(x,y)} 2^{-2n} = 1.$$

Public-coin $\leftrightarrow$ Private-coin

In Worten:

Die Wahrscheinlichkeit bzgl. $r_1, \ldots, r_t$ ist kleiner als 1, dass es eine Eingabe (x, y) gibt, so dass der Anteil der falschen Antworten $\geq (1 - 2\varepsilon)$ ist.

Anders herum:

Die Wahrscheinlichkeit bzgl. $r_1, \ldots, r_t$ ist größer 0, dass für jede Eingabe (x, y) der Anteil der falschen Antworten $< (1 - 2\varepsilon)$ ist.

$\rightarrow$ Es existiert $R = \{r_1, \ldots, r_t\}$ mit den geforderten Eigenschaften.

P' ist ein Protokoll für f mit Fehlerwahrscheinlichkeit $< 2\varepsilon$. $\square$

Randomisierte Kommunikationskomplexität von EQ

Randomisierte Protokolle können deutlich weniger Kommunikation benötigen als deterministische Protokolle.

Theorem 29

$R^{\mathrm{pu}}(\mathrm{EQ}) \in O(1)$

Beweis:

Alice und Bob haben einen Zufallsstring $z_1 z_2 \dots z_{2n} \in \{0,1\}^{2n}$.

Alice sendet $b_1 = \mathrm{IP}(x, z_1 z_2 \dots z_n)$ und $b_2 = \mathrm{IP}(x, z_{n+1} z_{n+2} \dots z_{2n})$.

Bob gibt 1 aus, falls

$$\mathrm{IP}(y, z_1 z_2 \dots z_n) = b_1 \text{ und } \mathrm{IP}(y, z_{n+1} z_{n+2} \dots z_{2n}) = b_2,$$

ansonsten gibt Bob 0 aus.

Randomisierte Kommunikationskomplexität von EQ

1. Fall: $x = y$

Dann gibt Bob unabhängig von z den korrekten Wert 1 aus.

D.h. mit Wahrscheinlichkeit 1 gibt Bob den richtigen Wert aus.

2. Fall: $x \neq y$

Sei $M = \{r \in \{0,1\}^n \mid \mathrm{IP}(x,r) = \mathrm{IP}(y,r)\}$.

Es gilt $|M| = 2^{n-1}$ (Beweis folgt).

Mit Wahrscheinlichkeit $1/2$, liegt ein Zufallsstring $z \in \{0,1\}^n$ in M.

Mit Wahrscheinlichkeit $1/4$, liegen zwei (unabhängige) Zufallsstrings $z, z' \in \{0,1\}^n$ in M.

Mit Wahrscheinlichkeit $3/4$ gibt Bob den korrekten Wert 0 aus.

Randomisierte Kommunikationskomplexität von EQ

Noch zu beweisen: $|M| = 2^{n-1}$

Seien $x = x_1 x_2 \dots x_n$, $y = y_1 y_2 \dots y_n$ und $I = \{i \in \{1, \dots, n\} \mid x_i \neq y_i\}$.

Es gilt $|I| \geq 1$, da $x \neq y$.

Für $i \in \{1, \dots, n\} \setminus I$ gilt $x_i = y_i$ und für $r = r_1 r_2 \dots r_n \in \{0,1\}^n$ gilt:

$$\mathrm{IP}(x, r) = \mathrm{IP}(y, r) \Leftrightarrow \sum_{i \in I} x_i \cdot r_i \mod 2 = \sum_{i \in I} y_i \cdot r_i \mod 2.$$

Gelte o.B.d.A. $x_i = 0$ und $y_i = 1$ für alle $i \in I$
(tauschen von Bits x_i und y_i ändert nichts daran ob $\mathrm{IP}(x, r) = \mathrm{IP}(y, r)$).

Es gilt $\sum_{i \in I} x_i \cdot r_i \mod 2 = 0$ und $\sum_{i \in I} y_i \cdot r_i \mod 2 = \sum_{i \in I} r_i \mod 2$.

Für wieviele $r \in \{0,1\}^n$ gilt $\sum_{i \in I} r_i \mod 2 = 0$?

Dies gilt für die Hälfte aller $r \in \{0,1\}^n$ (Übung). □

Randomisierte Kommunikationskomplexität von EQ

Folgendes Theorem folgt direkt aus den Theoremen 28 und 29.

Wir geben trotzdem einen alternativen, direkten Beweis an.

Theorem 30

$R^{\mathrm{pr}}(\mathrm{EQ}) \in O(\log n)$

Beweis:

Sei Alice Eingabe $a = a_0 \dots a_{n-1}$ und Bobs Eingabe $b = b_0 \dots b_{n-1}$.

Sei p eine Primzahl mit $4n^3 < p < 8n^3$
(die Existenz von p folgt aus Sätzen zur Primzahldichte).

Wir interpretieren x, y als Polynome über dem Körper $\mathrm{GF}(p) = \mathbb{Z}/p\mathbb{Z}$:

$$q_a(x) = \sum_{i=0}^{n-1} a_i x^i \pmod{p}, \quad q_b(x) = \sum_{i=0}^{n-1} b_i x^i \pmod{p}$$

Randomisierte Kommunikationskomplexität von EQ

Alice wählt nun zufällig $t \in \mathrm{GF}(p)$.

Nun sendet Alice t und $q_a(t)$ an Bob ($O(\log n)$ Bits).

Bob gibt 1 aus, falls $q_a(t) = q_b(t)$, andernfalls gibt er 0 aus.

1. Fall: $a = b$

Dann ist $q_a = q_b$ und somit $q_a(t) = q_b(t)$ unabhängig von t.

$\rightarrow$ Bob gibt 1 aus mit Wahrscheinlichkeit 1.

2. Fall: $a \neq b$

Es folgt $q_a \neq q_b$, wobei q_a und q_b jeweils Grad $\leq n-1$ haben.

Somit kann $q_a(t) = q_b(t)$ für höchstens $n-1$ Punkte t gelten.

Die Wahrscheinlichkeit einer falschen Antwort ist also höchstens

$$\frac{n-1}{p} \leq \frac{n}{4n^3} = \frac{1}{4n^2} \leq \frac{1}{4}.$$

Untere Schranken mittels Determinisierung

Untere Schranken sind im randomisierten Setting viel schwerer zu zeigen.

Eine Möglichkeit im private-coin Modell basiert auf Determinisierung:

Theorem 31

Sei $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. *Es gilt*

$$R^{\mathrm{pr}}(f) \in \Omega(\log D(f)).$$

Beweis:

Wir beweisen:

$$D(f) \leq 2^{R^{\mathrm{pr}}(f)} \cdot \left(\log \left(\frac{1}{2} - \varepsilon \right)^{-1} + R^{\mathrm{pr}}(f) \right)$$

Sei $P = (A, B)$ ein randomisiertes Protokoll für f mit Fehlerwahrscheinlichkeit $\varepsilon < 1/2$.

Untere Schranken mittels Determinisierung

Ziel: Ein deterministisches Protokoll P' für f basierend auf P.

Sei m die maximale Anzahl genutzter Zufallsbits in P.

Sei x die Eingabe von Alice und y die Eingabe von Bob.

Seien $s_1, \ldots, s_r$ alle möglichen Kommunikationsverläufe für x, y bzgl. P.

D.h. für jedes s_i gibt es Zufallsstrings z_A, z_B, so dass $s_i = s_P(x, y, z_A, z_B)$.

Es gilt $r \in O(2^{R^{\mathrm{pr}}(f)})$.

Sei $s_i = (m_1^i, \ldots, m_{r+1}^i)$ für $i \in \{1, \ldots, r\}$.

Wir definieren Wahrscheinlichkeiten p_A^i, p_B^i für $i \in \{1, \ldots, r\}$ bzgl. x, y.

Untere Schranken mittels Determinisierung

Sei p_A^i die Wahrscheinlichkeit, dass Alice (bzgl. P) wie s_i kommuniziert:

$$\frac{|\{z_A \mid \forall j \in \{0,\ldots,(r-1)/2\} : \ m^i_{1+2j} = A(x, z_A, m^i_1, m^i_2, \ldots, m^i_{2j})\}|}{2^m}$$

Sei p_B^i die Wahrscheinlichkeit, dass Bob (bzgl. P) wie s_i kommuniziert:

$$\frac{|\{z_B \mid \forall j \in \{0,\ldots,(r-1)/2\} : \ m^i_{2+2j} = B(y, z_B, m^i_1, m^i_2, \ldots, m^i_{2j+1})\}|}{2^m}$$

In P' soll Alice an Bob die Wahrscheinlichkeiten $p_A^1, \ldots, p_A^r$ senden.

Untere Schranken mittels Determinisierung

Bob kann dann die Wahrscheinlichkeit berechnen, dass s_i realisiert wird:

$$\text{Prob}[s_i] = p_A^i \cdot p_B^i$$

Die Wahrscheinlichkeit, dass $b \in \{0,1\}$ von P ausgegeben wird ist

$$\text{Prob}[o_P(x,y) = b] = \sum_{i:\, m_{r+1}^i = b} \text{Prob}[s_i].$$

Somit könnte Bob $f(x,y)$ berechnen,
da $\text{Prob}[o_P(x,y) = f(x,y)] \geq 1 - \varepsilon$ und $\text{Prob}[o_P(x,y) \neq f(x,y)] < \varepsilon$.

Problem: Wie überträgt Alice die rationalen Zahlen $p_A^1, \ldots, p_A^r$?

Untere Schranken mittels Determinisierung

Jedes p_A^i wird durch eine Wahrscheinlichkeit q_A^i approximiert, so dass

- q_A^i mit $k = \log\left(\frac{1}{2} - \varepsilon\right)^{-1} + R^{\mathrm{pr}}(f)$ Bits repräsentiert wird,
- d.h. $|p_A^i - q_A^i| \leq 2^{-k} = \left(\frac{1}{2} - \varepsilon\right)/2^{R^{\mathrm{pr}}(f)}$.

Somit ist auch der Fehler $|\mathrm{Prob}[s_i] - q_A^i \cdot p_B^i|$ beschränkt:

$$|p_A^i \cdot p_B^i - q_A^i \cdot p_B^i| = |p_A^i - q_A^i| \cdot p_B^i \leq \left(\frac{1}{2} - \varepsilon\right)/2^{R^{\mathrm{pr}}(f)} \quad (\text{da } 0 \leq p_B^i \leq 1)$$

Sei p_b die von Bob berechnete Schätzung für $\mathrm{Prob}[o_P(x,y) = b]$.
Es gilt $|p_b - \mathrm{Prob}[o_P(x,y) = b]| \leq 1/2 - \varepsilon$,

$$p_{f(x,y)} \geq 1 - \varepsilon - (1/2 - \varepsilon) = 1/2 \text{ und } p_{1-f(x,y)} < \varepsilon + (1/2 - \varepsilon) = 1/2.$$

Somit kann Bob mit Hilfe von p_0, p_1 den Wert $f(x,y)$ berechnen. □

Kommunikationskomplexität von EQ

Theorem 32

- $D(\mathrm{EQ}) \in \Theta(n)$
- $R^{\mathrm{pr}}(\mathrm{EQ}) \in \Theta(\log n)$
- $R^{\mathrm{pu}}(\mathrm{EQ}) \in \Theta(1)$

$R^{\mathrm{pr}}(\mathrm{EQ}) \in \Omega(\log n)$ folgt aus Theorem 31 und $D(\mathrm{EQ}) \in \Omega(n)$.

Ein weiteres Beispiel, wo Randomisierung von Vorteil ist:

Theorem 33

$R^{\mathrm{pr}}(\mathrm{GT}) \in O((\log n)^2)$

Beweis:

Wir nutzen ein Protokoll für EQ (Theorem 30) und binäre Suche.

Eine obere Schranke für GT

Seien $x = x_1 \dots x_n$ und $y = y_1 \dots y_n$ die Eingaben.

Alice und Bob suchen gemeinsam $k = \min\{i \in \{1, \dots, n\} \mid x_i \neq y_i\}$.

Sei zu Beginn $L = 1$ und $R = n$.

In jedem Schritt berechnen Alice und Bob $M = \lfloor (L + R)/2 \rfloor$
und führen gemeinsam das Protokoll für $\mathrm{EQ}(x_L \dots x_M, y_L \dots y_M)$ aus.

Falls 0 ausgegeben wird $(x_L \dots x_M \neq y_L \dots y_M)$, dann wird $R = M$ gesetzt.

Falls 1 ausgegeben wird, dann wird $L = M + 1$ gesetzt.

Dieses Vorgehen wird wiederholt bis $L = R$.

Eine obere Schranke für GT

Am Ende ($L = R$) sendet Alice x_L an Bob.

Bob gibt 1 aus, falls $x_L > y_L$ und sonst 0.

Falls das Protokoll für EQ immer die korrekte Antwort liefert,
dann ist dieses Protokoll GT korrekt (selbst überlegen).

Die Fehlerwahrscheinlichkeit hängt damit vom Protokoll für EQ ab.

Dieses wird $\lceil \log n \rceil$ mal aufgerufen.

Die Fehlerwahrscheinlichkeit für einen Aufruf ist $\leq 1/(4n^2)$.

Die gesamte Fehlerwahrscheinlichkeit ist $\leq \lceil \log n \rceil / (4n^2) \leq 1/4$ □

Kommunikationskomplexität von GT

Die obere Schranke lässt sich noch auf $O(\log n)$ verbessern.

Insgesamt ergibt sich für GT folgende Kommunikationskomplexität:

Theorem 34

- $D(\mathrm{GT}) \in \Theta(n)$
- $R^{\mathrm{pr}}(\mathrm{GT}) \in \Theta(\log n)$
- $R^{\mathrm{pu}}(\mathrm{GT}) \in \Theta(\log n)$

Die untere Schranke $R^{\mathrm{pu}}(\mathrm{GT}) \in \Omega(\log n)$ findet man hier:

S. N. Ramamoorthy, M. Sinha, "On the communication complexity of greater-than", Proc. of Allerton Conference, 2015

Diese untere Schranke überträgt sich auf $R^{\mathrm{pr}}(\mathrm{GT})$ (Theorem 28).

$R^{\mathrm{pr}}(\mathrm{GT}) \in O(\log n)$ überträgt sich entsprechend auf $R^{\mathrm{pu}}(\mathrm{GT})$.

Kommunikationskomplexität von $\mathrm{IP}, \mathrm{DISJ}$

Nicht immer führt Randomisierung zu erheblichen Verbesserungen.

Theorem 35

- $R^{\mathrm{pu}}(\mathrm{IP}) \in \Theta(n)$
- $R^{\mathrm{pu}}(\mathrm{DISJ}) \in \Theta(n)$

Beweise für die unteren Schranken von IP und DISJ findet man hier:

Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, "An information statistics approach to data stream and communication complexity", Journal of Computer and System Sciences, 2004

Ursprünglich wurden die Beweise für IP[1] und DISJ[2] in den 80ern geführt.

[1] B. Chor, and O. Goldreich, "Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity", Proc. of FOCS, 1985

[2] B. Kalyanasundaram, and G. Schnitger, "The Probabilistic Communication Complexity of Set Intersection", Proc. of Structure in Complexity Theory, 1987

Die Welt der Kommunikationskomplexität

- One-way Kommunikationskomplexität

  $\rightarrow$ Kommunikation auf eine Runde begrenzt.
  $\rightarrow$ Nützlich für untere Schranken im Streaming Modell.

- Multiparty Kommunikationskomplexität

  $\rightarrow$ Neben Alice und Bob gibt es weitere Parteien, die kommunizieren.
  $\rightarrow$ Die betrachteten Funktionen hängen von mehr als 2 Eingaben ab.

- Variable Partition Kommunikationskomplexität

  $\rightarrow$ Man betrachtet eine Funktion $f : \{0,1\}^n \rightarrow \{0,1\}$.
  $\rightarrow$ Bei Input $x_1 \dots x_n$ bekommt Alice $x_1 \dots x_k$ und Bob $x_{k+1} \dots x_n$.
  $\rightarrow$ Alice und Bob sollen $f(x_1 \dots x_n)$ berechnen.
  $\rightarrow$ Welchen Einfluss hat k auf die Kommunikationskomplexität?

- Quantum Kommunikationskomplexität

  $\rightarrow$ Das Berechnungsmodell basiert auf Quantenmechanik (Qubits).