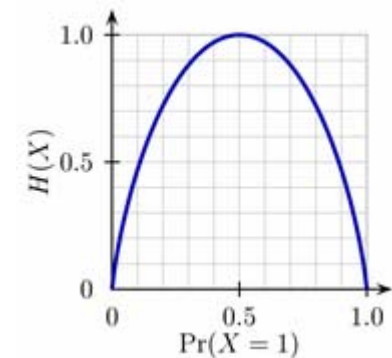


Information und Kommunikation

Hartmut Klauck
Universität Frankfurt
SS 07
9.7.



Monotone Spiele

- Ein monotones KW Spiel M_f für eine Funktion f ist wie folgt definiert:
 - Alice bekommt Eingabe x mit $f(x)=1$
 - Bob bekommt Eingabe y mit $f(y)=0$
 - Ziel ist es, einen Index i zu finden mit $x_i=1$ und $y_i=0$

Monotone Spiele

- **Theorem 24.1**

$D(M_f)$ ist gleich der minimalen Tiefe eines monotonen Schaltkreises für f für monotone Funktionen f

- Beweis ist völlig analog zum nichtmonotonen Fall

Das Matching Problem

- Wir betrachten im folgenden monotone Schaltkreise
- Eingabe ist ein ungerichteter ungewichteter Graph mit n Knoten als Adjazenzmatrix und ein Parameter m
- Ausgabe ist 1, wenn es eine Menge von m Kanten gibt, die paarweise keinen gemeinsamen Knoten haben

Das Matching Problem

- **Theorem 24.2**
 - Die monotone Tiefe des Matching Problems ist $O(n)$
- **Beweis:**
 - Wir betrachten der Einfachheit halber ein festes m
 - Der Schaltkreis ist ein ODER über alle Möglichkeiten, $2m$ Knoten zu wählen von Tests, ob es für diese ein perfektes Matching gibt
 - Es gibt $\binom{n}{2m} \leq 2^n$ viele Mengen von $2m$ Knoten
 - Das Oder trägt also n zur Tiefe bei
 - Wir müssen also zeigen, dass ein Test auf die Existenz eines perfekten Matchings in $O(n)$ Tiefe geht

Beweis 24.2

- **Fakt [Hall]**
 - Ein bipartiter Graph hat genau dann ein perfektes Matching, wenn für alle Teilmengen S der linken Knoten gilt: die Nachbarschaft von S (unter den rechten Knoten) ist mindestens so groß wie S
- Das unser Graph nicht bipartit ist, bilden wir ein großes Oder über alle Möglichkeiten m Knoten als linke Knoten zu wählen
- Die gibt uns zusätzliche Tiefe $\leq n$
- Jetzt müssen wir nur für alle Teilmengen von den m linken Knoten (ein Und mit $\leq n$ Tiefe) die Größe der Nachbarschaft unter den rechten Knoten bestimmen
- Genauer gesagt akzeptieren wir wenn die Nachbarschaft größer als die jeweilige Teilmenge ist für alle Teilmengen
- Dies ist (sogar in Tiefe $O(\log n)$) möglich

Matching Spiel

- Für das Matching Problem sieht das monotone Spiel M so aus:
 - Alice erhält einen Graphen G mit einem Matching der Größe $m=n/3$
 - Bob erhält einen Graphen H ohne ein Matching der Größe $n/3$
 - Ziel ist es, eine Kante zu finden die in G liegt aber nicht in H
- Wir werden zeigen, dass $D(M) \geq \Omega(n)$ in diesem Fall
- n ist die Knotenzahl der Graphen

Matching

- Wir schränken die Eingaben ein
- Alice erhält Graphen, die genau ein Matching der Größe $m=n/3$ sind
- Bob erhält Graphen der folgenden Form
 - Die Knotenmenge ist partitioniert in zwei Mengen S der Größe $m-1$ und T der Größe $2m+1$
 - Die Kanten sind alle Paare mit mindestens einem Knoten in S
 - Diese Graphen haben kein Matching der Größe m

Paar-Disjunktheit

- Wir wollen zeigen, dass $D(M)$ groß ist unter der obigen Einschränkung der Eingaben
- Dazu betrachten wir eine Variante des Disjunktheitsproblems
- Sei $n=3m$.
- X besteht aus allen *geordneten* Mengen P von m Paaren von Elementen von $\{1, \dots, n\}$ wobei alle Paare in P zusammen $2m$ unterschiedliche Elemente haben

Matching

- Also: Alice erhält P , eine Liste von m Paaren (die Paare sind paarweise disjunkt) aus der Menge $\{1, \dots, n\}$ (also natürlicherweise ein Matching)
- Bob erhält eine Menge S von $m-1$ Elementen aus $\{1, \dots, n\}$
- Wir transformieren Bobs Menge in einen Graphen:
 - S entspricht $m-1$ Knoten
 - Kanten sind alle Kanten zwischen Knoten in S und alle Kanten von S in das Komplement von S
- Alice und Bob benutzen ein Matching Protokoll um eine Kanten zu finden, die in Alices Matching liegt, aber nicht in Bobs Graph
- Eine solche Kante berührt die Menge S nicht
- Daher ist dies ein Element von P , das disjunkt von S ist wie gewünscht
- Alice sendet die Index von diesem Paar in P
- D.h. wir haben so das Paar-Disjunktheitsproblem gelöst
- $D(\text{Paar-Disj}) \leq D(M) + \log m$

Paar-Disj

- **Theorem 24.3**
 $D(\text{Paar-Disj}) \geq \Omega(R(\text{DISJ}) - \log n)$
- Da $R(\text{DISJ}) \geq \Omega(n)$ erhalten wir die gewünschte Schranke
- Dann folgt $MTiefe(\text{Matching}) \geq D(M) \geq D(\text{Paar-Disj}) - O(\log n) \geq \Omega(n)$

Beweis 24.3

- Wir zeigen $R(\text{DISJ}) \leq 2 D(\text{Paar-DISJ}) + 2 \log n$
- Gegeben sei also ein det. Protokoll P für Paar-Disj
- Sei P eine Menge von Paaren, so dass jedes Paar nur höchstens ein Element von S enthält
- Wie nennen die Relation Paar-Disj unter dieser Einschränkung der Eingaben PD'
- Jedes Protokoll für Paar-Disj löst auch PD'
- Also reicht es, $D(PD')$ zu beschränken

Beweis 24.3

- Wir betrachten eine neue Funktion f :
 - Für f erhält Bob eine Menge S der Größe m
 - Alice erhält P wie zuvor, aber unter der Einschränkung wie für PD'
- Wenn es ein Paar in P gibt, das kein Element von S enthält, dann gilt $f(P,S)=0$
- Wenn jedes Paar in P ein Element von S enthält, ist $f(P,S)=1$
- f ist eine partielle Funktion (d.h. nicht für alle Wahlen von P,S definiert)

Beweis 24.3

- Wir zeigen:
- **Lemma 24.4**
 - $R^{\text{pub}}(f) \leq 2D(\text{PD}') + 2\log n$
- Danach zeigen wir
- **Lemma 24.5**
 - $R^{\text{pub}}(\text{DISJ}) \leq R^{\text{pub}}(f)$ wobei DISJ auf $\{0,1\}^m \times \{0,1\}^m$ definiert ist
- Zusammen folgt dann, dass $M\text{Tiefe}(\text{Matching}) \geq D(\text{PD}') - O(\log n) \geq R^{\text{pub}}(\text{DISJ}) - O(\log n) = \Omega(n)$

Beweis 24.4

- Gegeben sei ein deterministisches Protokoll für PD'
- Wir suchen ein randomisiertes public coin Protokoll für f
- Eingaben seien P and S
- Bob löscht das kleinste Element von S und erhält S^* mit Größe $m-1$
- Mittels des public coin wird eine zufällige Permutation π von $\{1, \dots, n\}$ bestimmt
- Bob wendet diese Permutation auf S^* an und erhält S'
- [D.h. ein Element i wird zu $\pi(i)$]

Beweis 24.4

- Alice wendet π auf die Elemente von P an
- Zusätzlich permutiert Alice die Anordnung der m Paare mit einer weiteren zufälligen Permutation σ , und erhält P'
- Alice und Bob wenden das Protokoll für PD' auf die Eingaben P', S' an
- Sie erhalten einen Index i
- Für i gilt: das i -te Paar von P' enthält kein Element von S'
- Bob sendet das Element x , das er aus S entfernt hat an Alice ($\log n$ Bits)
- Ausgabe: Es wird akzeptiert, wenn $\pi(x)$ zum i -ten Paar von P' gehört
- Sonst wird verworfen

Beweis 24.4

- Es werden also $D(PD') + \log n$ Bits kommuniziert
- **Behauptung:** Das Protokoll hat Fehler $1/2$, wenn $f(P,S)=0$ ist und Fehler 0 sonst
- Daher kann zweifache Wiederholung den Fehler auf $1/4$ verringern

Beweis 24.4

- Fall 1: $f(P,S)=1$
- D.h. jedes Paar in P enthält genau ein Element von S
- Nach Entfernung von x gibt es genau 1 Paar, das kein Element von S hat
- Das Protokoll gibt i aus wenn i die Position dieses Paares in P' ist
- D.h. $\pi(x)$ steht in Paar i und es wird immer akzeptiert
- Bemerkung: Die Randomisierung hilft und schadet nichts in diesem Fall

Beweis 24.4

- Fall 2: $f(P,S)=0$
- D.h. es gibt ein Paar p_k in P das kein Element von S enthält, alle anderen höchstens eines
- S kann Elemente enthalten, die in keinem Paar von P liegen
 - Wenn x ein solches ist, dann ist $\pi(x)$ niemals in Paar i in P' und es wird immer verworfen
- Sei also x in einem der Paare in P
- D.h. nach Entfernen von x gibt es mind. 2 Paare in P , die kein Element von S enthalten
- Durch die Randomisierung wird mit Wahrscheinlichkeit $\geq 1/2$ ein i ausgegeben dessen Paar nicht $\pi(x)$ enthält

Beweis 24.4

- Seien die Paare p_k (enthält kein Element von S und nicht x) und p_l (enthält x aber kein Element von S^*)
- Zu jeder Permutationen π, σ des Protokolls gibt es Permutationen π', σ' , so dass P' und S' unverändert sind, aber $\pi(x)$ und $\pi'(x)$ liegen in unterschiedlichen Paaren von P
- Da die Permutationen zufällig sind, gilt mit Ws. mind. $1/2$, dass für eine beliebige Ausgabe i des Protokolls für $P \cap D'$ $\pi(x)$ nicht in i liegt

Beweis 24.4

- Sei (a',b') das Bild von $p_l=(a,b)$ unter π und σ
- (c',d') das Bild von $p_k=(c,d)$
- π' ist wie π definiert, nur
 - $\pi'(a)=c'$
 - $\pi'(b)=d'$
 - $\pi'(c)=a'$
 - $\pi'(d)=b'$
- σ' ist wie σ definiert, außer
 - $\sigma'(l)=k'$
 - $\sigma'(k)=l'$
- Unter π,σ ist $\pi(x)\in (a',b')$
- Unter π',σ' ist $\pi'(x)\in (a,b)$
- Weiterhin gilt dass die erzeugten P',S' gleich sind

Beweis 24.5

- Wir haben ein rand. Protokoll für f und wollen eines für DISJ finden, mit derselben Kommunikation und Fehlerwahrscheinlichkeit
- Alice erhält also ein $x \in \{0,1\}^m$ und konstruiert m Paare aus $\{1, \dots, n=3m\}$
 - Für jedes i ist das i -te Paar $(3i-x_i-1, 3i)$
- Bob erhält $y \in \{0,1\}^m$ und konstruiert eine Menge S der Größe m
 - S enthält Elemente $s_i = 3i - y_i$
- Alice und Bob wenden das Protokoll für f an

Beweis 24.5

- Wenn $\text{DISJ}(x,y)=0$ gibt es ein i , so dass $x_i=y_i=1$
- Dann gibt es ein Paar $p_i=(3i-2,3i)$ und S enthält das Element $s_i=3i-1$
- s_j für $j \neq i$ liegt niemals in einem Paar p_i
- Also gibt es ein Paar, das kein Element von S enthält und $f(S,P)=0$

Beweis 24.5

- Wenn $\text{DISJ}(x,y)=1$, dann ist für alle i entweder $x_i=0$ oder $y_i=0$
- Also entweder $s_i=3i$ und das Paar p_i ist $(3i-2,3i)$ [wenn $x_i=1$] oder $(3i-1,3i)$ [wenn $x_i=0$]
- Oder $s_i=3i-1$ und $p_i=(3i-1,3i)$
- In beiden Fällen ist $f(S,P)=1$
- D.h. die Fehlerwahrscheinlichkeiten für DISJ sind genau wie für f

Weitere Anwendungen

- Ein weiteres Resultat zu monotonen Schaltkreistiefe
- ST-conn ist das Problem, zu einem gerichteten Graphen mit n Knoten, einer Quelle und einer Senke zu entscheiden, ob es einen Weg von der Quelle zur Senke gibt
- ST-conn ist monoton
- Man kann ein monotones Spiel für st-conn definieren und die zugrundeliegende Relation analysieren und erhält:
- **Theorem 24.6**
 - $MTiefe(st-conn) = \Omega(\log^2 n)$

Monotone Tiefenhierarchien

- Wir haben mit Matching eine Funktion, deren monotone Tiefenkomplexität $\Theta(n)$ bei einer Eingabelänge von n^2 ist
- Man kann die Eingabe künstlich verlängern und für alle Funktionen $f(n)$ kleiner als Wurzel n Funktionen finden, deren monotone Tiefenkomplexität genau $\Theta(f(n))$ ist bei Eingabelänge n
- Weiterhin kann man für alle k eine Funktion f_k mit n^k Eingaben finden, die in linearer Größe und Tiefe k monoton berechenbar ist durch eine Formel mit UND/ODER Gattern von fan-in n , und für die alle monotonen Schaltkreise mit Tiefe $k-1$ und beliebigem fan-in exponentielle Größe haben