

SE Kommunikationskomplexität ^{*}

Randomisierung, Teil 1

Kornelius Kalnbach

15. November 2005

Einleitung

Um randomisierte Kommunikations-Szenarien abzudecken, erweitern den Protokollbegriff. Sowohl die Kommunikationskomplexität als auch die Ausgabe selbst sind dann nicht mehr deterministisch, sondern zufällig in bestimmten Grenzen.

Es gibt zwei Protokollvarianten, die wir betrachten:

- **Las-Vegas-Protokolle**, die *immer* den richtigen Funktionswert liefern
- **Monte-Carlo-Protokolle**, die den richtigen Wert *mit großer Wahrscheinlichkeit* liefern

In ähnlicher Weise kategorisieren wir auch Kosten, Laufzeit und Kommunikationskomplexität bei randomisierten Protokollen.

3.1 Grundlegende Definitionen

Das Szenario wird um zwei **Zufalls-Strings** r_A und r_B für Alice und Bob erweitert. Diese erlauben es den Beiden, zufällige Entscheidungen zu treffen.

Der Protokollbaum enthält dann Funktionen von x und r_A (für Alice) nach $\{0, 1\}$. Jede Kombination aus x, y, r_A, r_B führt zu genau einem Blatt des Baumes. Es ist also möglich, dass für dieselben Werte x, y und verschiedene r_A, r_B auch verschiedene Werte berechnet werden; die Protokolle können sich irren.

^{*}<http://www.informatik.hu-berlin.de/ldb/lehre/WS05-06/Kommunikationskomplexität/>

Definition 3.1: Protokolle und ε -Fehler

Ein **randomisiertes Protokoll** $\tilde{\mathcal{P}}^{[1]}$ berechnet eine Funktion f ...

- **ohne Fehler**, wenn für alle Eingaben (x, y) :

$$\Pr[\tilde{\mathcal{P}}(x, y) = f(x, y)] = 1$$

- **mit einem Fehler ε** , wenn für alle Eingaben (x, y) :

$$\Pr[\tilde{\mathcal{P}}(x, y) \neq f(x, y)] \leq \varepsilon$$

- **mit einem einseitigen Fehler ε** , wenn für alle Eingaben (x, y) :

$$\Pr[\tilde{\mathcal{P}}(x, y) = f(x, y) = 0] = 1 \quad \text{falls } f(x, y) = 0$$

und

$$\Pr[\tilde{\mathcal{P}}(x, y) = f(x, y) = 0] \leq \varepsilon \quad \text{falls } f(x, y) = 1$$

Eine solche Funktion liefert also mit einer Wahrscheinlichkeit ε ein *False Negative*.

Da Protokolle natürlich entweder 0 oder 1 liefern, gilt:

$$\Pr[\tilde{\mathcal{P}}(x, y) = 1 - f] = 1 - \Pr[\tilde{\mathcal{P}}(x, y) = f]$$

Definition 3.2: Randomisierte Laufzeit und Kosten

Für ein randomisiertes Protokoll ist...

- die **Worst-Case-Laufzeit** die *maximale* Anzahl kommunizierter Bits für alle r_A, r_B
- die **Average-Case-Laufzeit** die *zu erwartende* Anzahl kommunizierter Bits f.a. r_A, r_B

Außerdem sind...

- die **Worst-Case-Kosten** die *maximale* Worst-Case-Laufzeit für alle Eingaben (x, y)
- die **Average-Case-Kosten** die *maximale* Average-Case-Laufzeit f.a. Eingaben (x, y)

Die drei Fehlervarianten liefern drei Komplexitätsmaße. In jeder Variante ist die Komplexität gleich den Kosten des besten Protokolls, das die Fehlerschranke erfüllt.

^[1]Die Notation $\tilde{\mathcal{P}}$ stammt von mir; im Buch steht nur \mathcal{P} .

Definition 3.3: Randomisierte Kommunikationskomplexität: $R_0, R_\varepsilon, R_\varepsilon^1, R, R^1$

Sei f eine Funktion. Wir definieren folgende Komplexitätsmaße für f :

- $R_0(f)$ sind die *minimalen Average-Case-Kosten* für ein Protokoll, das f ohne Fehler berechnet.
- $R_\varepsilon(f)$ ($0 < \varepsilon < \frac{1}{2}$) sind die *minimalen Worst-Case-Kosten* für ein Protokoll, das f mit Fehler ε berechnet.
- $R_\varepsilon^1(f)$ ($0 < \varepsilon < 1$) sind die *minimalen Worst-Case-Kosten* für ein Protokoll, das f mit einseitigem Fehler ε berechnet.

Zusätzlich definieren wir die **Abkürzungen**

$$R := R_{\frac{1}{3}} \quad R^1 := R_{\frac{1}{2}}^1$$

Begründung der Begriffe

Warum wählen wir für R_0 die Average-Case-Kosten? Die Worst-Case-Kosten sind nicht sonderlich interessant, da wir hier genau die deterministische Kommunikationskomplexität erhalten würden.

Warum wählen wir dann für die anderen Maße den Worst Case? Im Allgemeinen kann man mit diesem Wert besser rechnen; und das folgende Lemma macht deutlich, dass der Unterschied nicht problematisch ist:

Faktor-Lemma für fehlerbehaftete Protokolle

Die Worst-Case-Kosten eines fehlerbehafteten Protokolls unterscheiden sich von den Kosten im Average-Case maximal um einen konstanten Faktor.

Beweis

Gegeben sei ein Protokoll $\tilde{\mathcal{P}}$, das f mit einem Fehler von ε berechnet und dabei im Average Case eine Kommunikation von t Bits erfordert.

Wir konstruieren nun ein zweites Protokoll $\tilde{\mathcal{P}}'$, bei dem $\tilde{\mathcal{P}}$ ausgeführt wird, bis maximal $\frac{t}{\varepsilon}$ Bits kommuniziert wurden. Wenn $\tilde{\mathcal{P}}$ bis dahin beendet ist, benutzen wir die Ausgabe; andernfalls geben wir 0 aus.

Die Wahrscheinlichkeit, dass wir vorzeitig abbrechen, dass also $\tilde{\mathcal{P}}$ mehr als $\frac{t}{\varepsilon}$ Bits benötigt, ist höchstens ε aufgrund der Markoff-Ungleichung:

$$\Pr \left[|\tilde{\mathcal{P}}(X)| \geq \frac{t}{\varepsilon} \right] \leq \frac{\mathbb{E}[|\tilde{\mathcal{P}}(X)|]}{\frac{t}{\varepsilon}} = \frac{t}{\frac{t}{\varepsilon}} = \varepsilon$$

Der Fehler des neuen Protokolls kann also höchstens $\varepsilon + \varepsilon = 2\varepsilon$ betragen (Fehler von $\tilde{\mathcal{P}}$ + Fehler durch vorzeitigem Abbruch), und die Kommunikationskomplexität im Worst Case ist jetzt $\frac{t}{\varepsilon} = t \cdot \frac{1}{\varepsilon}$. □

Beispiel 3.5

Betrachte die Gleichheits-Funktion EQ. Diesmal seien $a = a_0a_1 \dots a_{n-1}$ und $b = b_0b_1 \dots b_{n-1}$ die Eingaben von Alice und Bob.

Dann wählen wir eine Primzahl $n^2 < p < 2n^2$ und interpretieren die Eingabewerte als Polynome über dem Restklassenring \mathbb{F}_p , also

$$A(x) := a_0 + a_1x + \dots + a_{n-1}x^{n-1} \pmod{p}$$

Analog ist $B(x)$ definiert.

Alice wählt eine Zahl $t \in \mathbb{F}_p$ und sendet t und $A(t)$ an Bob. Dieser antwortet mit 1, wenn $A(t) = B(t)$ und sonst mit 0.

Die Anzahl der kommunizierten Bits ist

$$2 \log p < 2 \log 2n^2 = 4 \log n + 2 \log 2$$

also lediglich $O(\log n)$. Das ist ein großer Fortschritt gegenüber $D(\text{EQ}) = n + 1$. Wie korrekt arbeitet dieses Protokoll?

Für $a = b$ ist $A(t) = B(t)$ für alle t , also gibt es keine *False Negatives*.

Für $a \neq b$ ergeben sich zwei unterschiedliche Polynome A und B vom Grad $n - 1$. Solche Polynome können sich maximal an $n - 1$ Stellen treffen, da ihre Differenz ebenfalls ein Polynom höchstens $n - 1$. Grades ist und damit maximal $n - 1$ Nullstellen hat. Die Wahrscheinlichkeit eines *False Positive* (also $A(t) = B(t)$ obwohl $a \neq b$) beträgt also höchstens

$$\varepsilon \leq \frac{n-1}{p} < \frac{n}{n^2} = \frac{1}{n}$$

Damit haben wir gezeigt, dass $R(\text{EQ}) = O(\log n)$ und sogar $R_{\frac{1}{n}}(\text{EQ}) = O(\log n)$.

3.2 Randomisierung vs. Determinismus

Wie weit kann sich randomisierte Komplexität von deterministischer unterscheiden? Welche Arten unterer Schranken können wir beweisen?

Randomisierung ohne Fehler kann jedenfalls nicht stärker sein als Nichtdeterminismus, da man ja nichtdeterministisch die besten Zufalls-Strings wählen könnte. Daraus folgt:

Folgerung 3.7

Für jedes $0 \leq \varepsilon < 1$ gilt:

$$R_{\varepsilon}^1(f) \geq N^1(f)$$

Analog $R_0(f) \geq N(f)$.

Laut Satz 2.11 gilt $D(f) = O(N(f)^2)$, also kann $R(f)$ maximal quadratisch kleiner sein als $D(f)$. Später im Beispiel 3.16 wird diese Schranke auch erreicht.

Das Beispiel 3.5 oben hat hingegen gezeigt, dass der Unterschied von $R^1(f)$ und $R(f)$ zu $D(f)$ exponentiell sein kann. Folgerung 3.7 gibt eine untere Schranke für die einseitige Komplexität $R^1(f)$.

Lemma 3.8

Folgendes Lemma gibt eine untere Schranke für $R(f)$:

$$R(f) = \Omega(\log D(f))$$

Behauptung

Wir werden eine etwas komplexere Aussage beweisen:

$$D(f) \leq 2^{R_\varepsilon(f)} \cdot \left(R_\varepsilon(f) - \log\left(\frac{1}{2} - \varepsilon\right) \right)$$

Beweis

Sei $\tilde{\mathcal{P}}$ ein randomisiertes Protokoll zur Berechnung von f .

Sei $L := 2^{R_\varepsilon(f)}$ die Anzahl der Blätter des Protokollbaumes.

Wir konstruieren ein deterministisches Protokoll mit oben genannter Komplexität.

Für jedes Blatt l des Protokolls schickt Alice den Wert p_l^A - die Wahrscheinlichkeit, dass sie bei gegebener Eingabe x den Weg zum Blatt l auswählt. Bob berechnet dann p_l^B und $p_l := p_l^A \cdot p_l^B$, also die Wahrscheinlichkeit, dass das Blatt l erreicht wird.

Diese Berechnung wird für jedes der L Blätter durchgeführt. Zu jedem Blatt gehört ein Ausgabewert, und Bob kann jetzt ausrechnen, welcher Wert eine Wahrscheinlichkeit von mindestens $1 - \varepsilon$ hat. Dieser Wert ist dann $f(x, y)$.

Die Schwierigkeit besteht darin, dass Alice *reelle* Zahlen p_l^A senden muss. Es wird allerdings nur eine begrenzte Genauigkeit benötigt: Wir stellen sicher, dass sich der gesendete Wert vom echten um maximal

$$2^{-k} = \frac{\frac{1}{2} - \varepsilon}{L}$$

unterscheidet, indem Alice auf k Bit rundet. Dann ist der summierte Fehler für p_l über allen Blättern maximal $\frac{1}{2} - \varepsilon$. Bob wählt dann den Wert mit der größeren Wahrscheinlichkeit ($> \frac{1}{2}$).

Umstellen der obigen Gleichung führt zu

$$k = R_\varepsilon(f) - \log\left(\frac{1}{2} - \varepsilon\right)$$

Alice muss also $L \cdot k$ Bits senden, woraus die Behauptung folgt. □

Beispiel 3.9

Mit den bisher erreichten Mitteln und den Schranken aus Kapitel 2 können wir die randomisierte Komplexität von EQ und NE komplett bestimmen:

- $R^1(\text{EQ}) = \Theta(n)$, $R_0(\text{EQ}) = R_0(\text{NE}) = \Theta(n)$
- $R^1(\text{NE}) = \Theta(\log n)$, $R(\text{EQ}) = R(\text{NE}) = \Theta(\log n)$.