Komplexität des Maximums einer polynomzeitbeschränkten Funktion

Seminar Reelle Komplexität Sommersemester 2011

Holger Thies

06.06.2011

Inhalt

Berechenbarkeit von Funktionalen

- Bisher gesehen: Berechenbarkeit und Komplexität von reellen Zahlen und reellen Funktionen
- Jetzt: Berechnung eines numerisches Funktionals, d.h. einer Abbildung von einer reellen Funktion auf eine reelle Zahl.
- z.B. finde Abbildung die eine Funktion $f \in C[0,1]$ abbildet auf die reelle Zahl $\int_0^1 f(t)dt$.
- Wir benötigen zunächst ein formales Berechnungsmodell für solche Funktionale.

Berechenbarkeit von Funktionalen

In der numerischen Analysis wird oft implizit ein Modell mit folgenden Eigenschaften angenommen:

- Der Algorithmus kann nach dem Wert für $f(x)$ für alle x fragen
- Der Algorithmus bekommt das exakte Ergebnis $y = f(x)$ durch ein Orakel
- Jede solche Anfrage benötigt einen Zeitschritt

Mit diesem Modell ist es einfach, Komplexitätsschranken zu finden. Allerdings ist es kein realistisches Modell, da mit exakten reellen Zahlen gearbeitet wird.

Berechenbare Funktionale

Definition

Zwei Funktionen $m : \mathbb{N} \to \mathbb{N}$ und $\Phi : (\mathbb{D} \cap [0,1]) \times \mathbb{N} \to \mathbb{D}$ **repräsentieren** f, wenn

1. Die Funktion m ist eine Modul-Funktion für f auf $[0,1]$
   - $|x - y| \leq 2^{-m(n)} \implies |f(x) - f(y)| \leq 2^{-n}$ für $x, y \in [0,1]$
2. für alle $d \in \mathbb{D} \cap [0,1]$ und alle $n \in \mathbb{N}$ gilt $|\Phi(d,n) - f(d)| \leq 2^{-n}$

Beispiel

Beispiel

Die Funktionen $m : \mathbb{N} \to \mathbb{N}$, $m(k) = k + 1$ *und* $\Phi : (\mathbb{D} \cap [0,1]) \times \mathbb{N} \to \mathbb{D}$, $\Phi(d,n) = d^2$ *repräsentieren die Funktion* $f : [0,1] \to \mathbb{R}, f(x) = x^2$.

Berechenbare Funktionale

Definition

Ein numerisches Funktional F auf $D \subseteq C[0,1]$ heißt **berechenbar**, wenn es eine zwei-Orakel Turingmaschine M gibt, so dass für jede Funktion $f \in D$, alle Orakel-Funktionen m und Φ die f repräsentieren, und jede Eingabe $n \in \mathbb{N}$ gilt

$$|M^{m,\phi}(n) - F(f)| \leq 2^{-n}$$

Notwendigkeit des Modul-Orakels

- Die Maschine muss wissen, wie nah sie durch die Berechnung von $f(d)$ am gewünschten Wert von $f(x)$ ist.
- Betrachte das Funktional $F : C[0,1] \to \mathbb{R}, F(f) = f(\frac{\sqrt{2}}{2})$
- $\frac{\sqrt{2}}{2} \notin \mathbb{D}$, d.h. eine direkte Anfrage an Φ ist nicht möglich.
- Angenommen das Modul-Orakel m ist vorhanden
  - Eine Approximation $e \in D$ mit $|f(\frac{\sqrt{2}}{2}) - e| \leq 2^{-n}$ kann gefunden werden durch:
  - Berechne $k = m(n+1)$
  - Finde $d \in \mathbb{D}$, so dass $|d - \frac{\sqrt{2}}{2}| \leq 2^{-k}$
  - Frage Φ nach $e = \Phi(d, n+1)$

Notwendigkeit des Modul-Orakels

- Angenommen die Maschine verfügt nicht über das Modul-Orakel.
- Es kann vorkommen, dass das Orakel für alle Anfragen der Maschine $\Phi(d,k) = 0$ für alle $d < x$ und $\Phi(d,k) = 1$ für alle $d > x$.
- Sei d_1 der größte Punkt kleiner x und d_2 der kleinste Punkt größer x, nach dem die Maschine das Orakel befragt hat.
- Die Maschine kann nicht in endlich vielen Schritten zwischen den beiden Funktionen f_1 und f_2 unterscheiden:
- $f_1(0) = 0,\ f_1(d_1) = 0, f_1(x) = 1, f_1(1) = 1$
- $f_2(0) = 0,\ f_2(x) = 0, f_2(d_2) = 1, f_1(1) = 1$

Berechenbarkeit des Maximums

Theorem

Sei $f : [0,1] \to \mathbb{R}$ *berechenbar. Dann ist der Wert des Maximums* $max(f)$ *berechenbar.*

Komplexität der Maximumsberechnung

- In der diskreten Komplexitätstheorie wurde von vielen Maximierungsproblemen NP-Vollständigkeit nachgewiesen.
- Dagegen ist die Maximierung von reellwertigen Problemen häufig in polynomieller Zeit möglich.
- Wir betrachten nun in polynomieller Zeit berechenbare Funktionen $f : [0, 1] \to \mathbb{R}$.
- Ist der Wert $max(f)$ des Maximums von f eine in polynomieller Zeit berechenbare reelle Zahl?

Komplexität der Maximumsberechnung

Theorem

Die folgenden Aussagen sind äquivalent:

(a) $P = NP$

(b) *Die Funktion* $g(x) = max\{f(x,y) \mid 0 \leq y \leq 1\}$ *ist in* $P_{C[0,1]}$ *für alle* $f \in P_{C[0,1]^2}$

(c) *Die Funktion* $h(x) = max\{f(y) \mid 0 \leq y \leq x\}$ *ist in* $P_{C[0,1]}$ *für alle* $f \in P_{C[0,1]}$

(d) *Die Funktion* $k(x) = max\{f(y) \mid 0 \leq y \leq 1\}$ *ist in* $P_{C[0,1]}$ *für alle* $f \in P_{C^\infty[0,1]}$

Beweis $(a) \implies (b)$

$P = NP \implies g(x) = max\{f(x,y) \mid 0 \leq y \leq 1\}$ ist in $P_{C[0,1]}$

- Sei o.B.d.A. $im(f) \subseteq [0,1]$.
- Sei M eine Zwei-Orakel-Maschine die f in Zeit p(n) für ein Polynom p berechnet.
- $B \in NP$ gdw. $B = \{x \in \Sigma^* \mid \exists y \in \Sigma^{\leq p(|x|)} < x, y > \in K\}, K \in P$
- $A := \{< d_1, e > \mid e \in \mathbb{D}_{n+1} \cap [0,1], d_1 \in \mathbb{D}_{p(n+2)} \cap [0,1]$ für ein $n \geq 0$ für die gilt $\exists d_2 \in \mathbb{D}_{p(n+2)} \cap [0,1]$ und $e \leq M^{b_{d_1}, b_{d_2}}(n+2)\}$

Beweis $(a) \implies (b)$

Erinnerung:
$g(x) = max\{f(x,y) \mid 0 \leq y \leq 1\}$
$A := \{< d_1, e > \mid e \in \mathbb{D}_{n+1} \cap [0,1], d_1 \in \mathbb{D}_{p(n+2)} \cap [0,1]$ für ein $n \geq 0$
für die gilt $\exists d_2 \in \mathbb{D}_{p(n+2)} \cap [0,1]$ und $e \leq M^{b_{d_1}, b_{d_2}}(n+2)\}$

Sei $d_1 \in \mathbb{D}_{p(n+2)} \cap [0,1]$, $x \in [0,1]$ mit $|d_1 - x| \leq 2^{-p(n+2)}$ und
$e = max\{e_1 \in \mathbb{D}_{n+1} \cap [0,1] \mid < d_1, e_1 > \in A\}$
Dann gilt:

(i) $|e - g(d_1)| \leq 2^{-(n+1)}$

(ii) $|g(d_1) - g(x)| \leq 2^{-(n+2)}$

Daraus folgt $|e - g(x)| \leq 2^{-n}$.

Beweis $(a) \implies (b)$

- Aus $|e - g(d_1)| \leq 2^{-(n+1)}$ und $|g(d_1) - g(x)| \leq 2^{-(n+2)}$ folgt $|e - g(x)| \leq 2^{-n}$
- Eine Turingmaschine mit A als Orakel kann e durch binäre Suche in polynomieller Zeit bestimmen.
- Gilt P = NP ist inbesondere $A \in P$.
- D.h. g kann in polynomieller Zeit berechnet werden.

Beweis $(b) \implies (c)$

(b) Die Funktion $g(x) = max\{f(x,y) \,|\, 0 \leq y \leq 1\}$ ist in $P_{C[0,1]}$ für alle $f \in P_{C[0,1]^2}$
(c) Die Funktion $h(x) = max\{f(y) \,|\, 0 \leq y \leq x\}$ ist in $P_{C[0,1]}$ für alle $f \in P_{C[0,1]}$

Beweis.

Für jedes $f \in P_{C[0,1]}$ definiere die Funktion f_1 auf $[0,1]^2$ durch:

$$f_1(x,y) = \begin{cases} f(0), & \text{falls } y > x \\ f(x-y), & \text{falls } y \leq x \end{cases}$$

Dann gilt $f_1 \in P_{C[0,1]^2}$ und für alle $x \in [0,1]$

$$g(x) = max\{f_1(x,y) \,|\, 0 \leq y \leq 1\} = max\{f(y) \,|\, 0 \leq y \leq x\} = h(x)$$

Beweis $(c) \implies (d)$

(c) Die Funktion $h(x) = max\{f(y) \mid 0 \leq y \leq x\}$ ist in $P_{C[0,1]}$ für alle $f \in P_{C[0,1]}$
(d) Die Funktion $k(x) = max\{f(y) \mid 0 \leq y \leq 1\}$ ist in $P_{C[0,1]}$ für alle $f \in P_{C^\infty[0,1]}$

Beweis.

Jedes $f \in P_{C^\infty[0,1]}$ ist auch in $P_{C[0,1]}$. □

Beweis $(d) \implies (a)$

Lemma

Es gibt eine Funktion $f \in P_{C^\infty[0,1]}$ *mit folgenden Eigenschaften:*

1. $f(0) = 0$ *und* $f(1) = 1$,
2. $f^{(n)}(0) = f^{(n)}(1) = 0$ *für alle* $n \geq 1$,
3. f *ist monton wachsend auf* $[0, 1]$,
4. $f^{(n)}$ *ist in* $P_{C[0,1]}$ *für alle* $n \geq 1$.

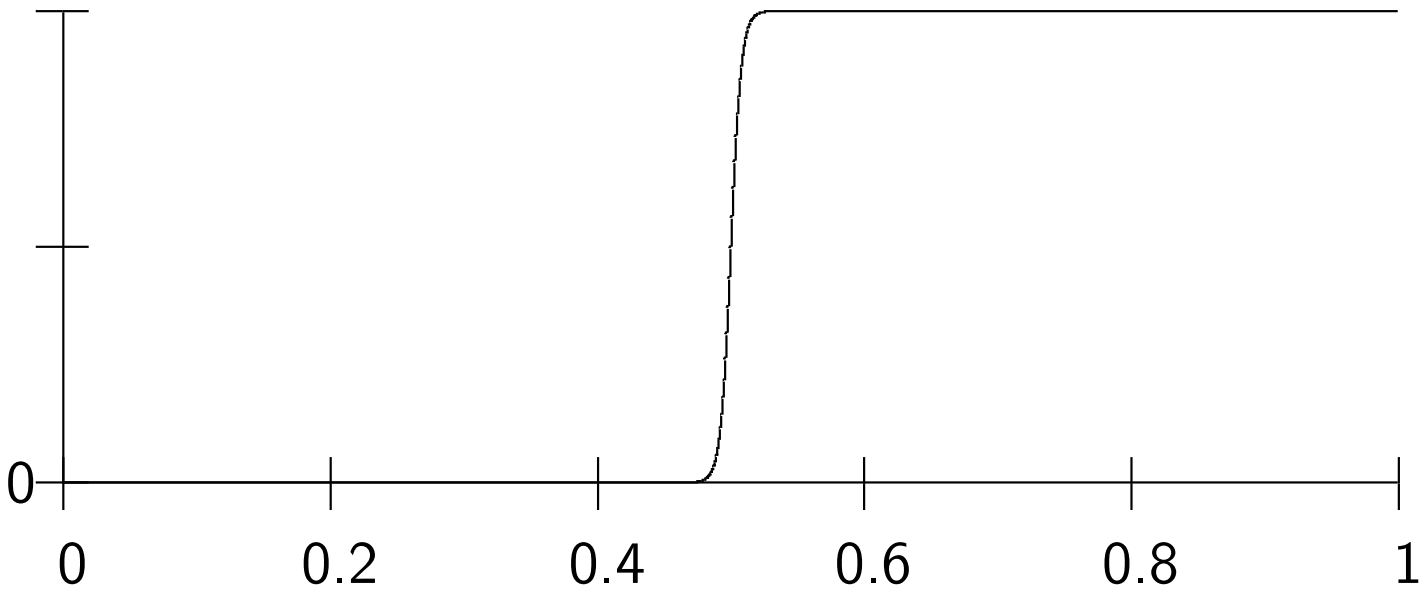
Beweis.

Sei

$$h(x) = \begin{cases} e^{-\frac{1}{x^2}}, & \text{falls } x > 0 \\ 0, & \text{falls } x \leq 0 \end{cases}$$

und

$$f(x) = \frac{h(x - \frac{1}{4})}{h(\frac{3}{4} - x) + h(x - \frac{1}{4})}$$



Beweis $(d) \implies (a)$

$k(x) = max\{f(y) \mid 0 \leq y \leq 1\} \in P_{C[0,1]}$ für alle
$f \in P_{C^\infty[0,1]} \implies P = NP$

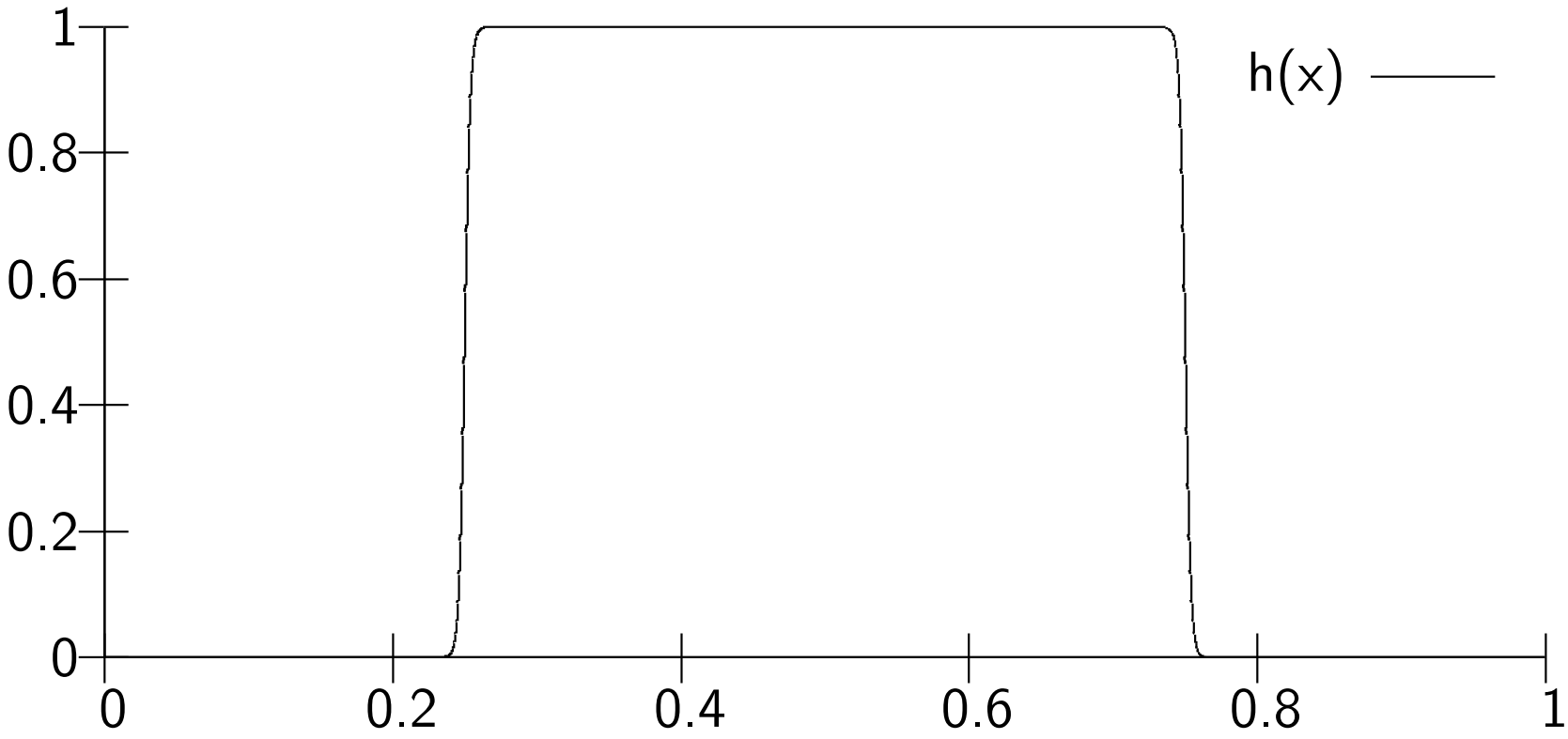
- Sei $A \in NP$. Wir suchen eine Funktion $f \in P_{C^\infty[0,1]}$, so dass $k \in P_{C[0,1]} \implies A \in P$
- $A \in NP : A = \{x \in \Sigma^* \mid \exists y \in \Sigma^{p(|x|)} : < x, y > \in K\}, K \in P$

Beweis $(d) \implies (a)$

- Teile das Intervall $[0,1]$ in Subintervalle, so dass jedes Intervall zu einem String $s \in \{0,1\}^*$ gehört.
- Für jedes $n \geq 1$, sei $a_n = 1 - 2^{-(n-1)}$
- Für $s \in \{0,1\}^n$, $u_s = a_n + i2^{-2n}$, $v_s = u_s + 2^{-2n}$
  - i ist die zu s gehörige Zahl, wenn man s als n-bit Binärdarstellung auffasst.
- Teile dann die Intervalle $[u_s, \frac{u_s+v_s}{2}]$ in $2^{p(n)}$ Subintervalle
- $y_{s,t} = u_s + i2^{-(p(n)+2n+1)}$ und $z_{s,t} = y_{s,t} + 2^{-(p(n)+2n+1)}$
  - Das Intervall $[u_s, \frac{u_s+v_s}{2}]$ hat Länge $2^{-(2n+1)}$, also wird es so in $2^{p(n)}$ gleich große Intervalle geteilt.

Beweis $(d) \implies (a)$

- Sei g_1 die Funktion aus dem Lemma
- $h_1 : [0,1] \to \mathbb{R}, h_1(x) = \begin{cases} g_1(2x), \text{ falls } 0 \leq x \leq \frac{1}{2} \\ g_1(2-2x), \text{ falls } \frac{1}{2} \leq x \leq 1 \end{cases}$



Beweis $(d) \implies (a)$

- Definiere f durch $f(x) = u_s + g_1(\frac{2}{v_s - u_s}x - \frac{v_s + u_s}{v_s - u_s})$ für $x \in [\frac{u_s + v_s}{2}, v_s]$
- $f(x) =$
$$\begin{cases} u_s & , \text{falls } < s,t > \notin K \\ u_s + 2^{-(p(n)+2n+2)} h_1(2^{p(n)+2n+1}(x - y_{s,t})) & , \text{falls } < s,t > \in K \end{cases}$$
für $x \in [y_{s,t}, z_{s,t}]$
- f hat Hügel der Höhe $2^{-(p(n)+2n+2)}$ in $[y_{s,t}, z_{s,t}]$, wenn t Zertifikat dafür ist, dass $s \in A$. Andernfalls ist f flach.
- $f \in P_{C^\infty[0,1]}$
  - $h_1 \in P_{C^\infty[0,1]}$
  - $h^{(n)}(0) = h^{(n)}(1) = 0 \, \forall n \geq 0$

Beweis $(d) \implies (a)$

- Angenommen $k \in P_{C[0,1]}$, dann lässt sich $s \in A$ folgendermaßen entscheiden:
  - Berechne Approximation e für $k((u_s + v_s)/2)$ mit Fehler $\leq 2^{-(p(n)+2n+4)}$
  - $s \in A$ gdw. $e > u_s$.

Zusammenfassung

- Das Maximum jeder berechenbaren Funktion ist berechenbar.
- Wenn $P \neq NP$ gibt es in polynomieller Zeit berechenbare, stetig differenzierbare Funktionen, deren Maximum nicht in polynomieller Zeit berechnet werden kann.
- Das Maximierungsproblem für eine polynomiell berechenbare, stetig differenzierbare Funktionen kann so schwer sein, wie ein NP-hartes Problem.