

Übersicht

Einführung

Grundlagen

Methoden zum Entwurf von Approximationsalgorithmen

Approximationsklassen

Das PCP-Theorem

Probabilistische Algorithmen

Reduktion und Vollständigkeit

Approximations-
Algorithmen

Einführung

Grundlagen

Methoden zum
Entwurf

Approximationsklassen

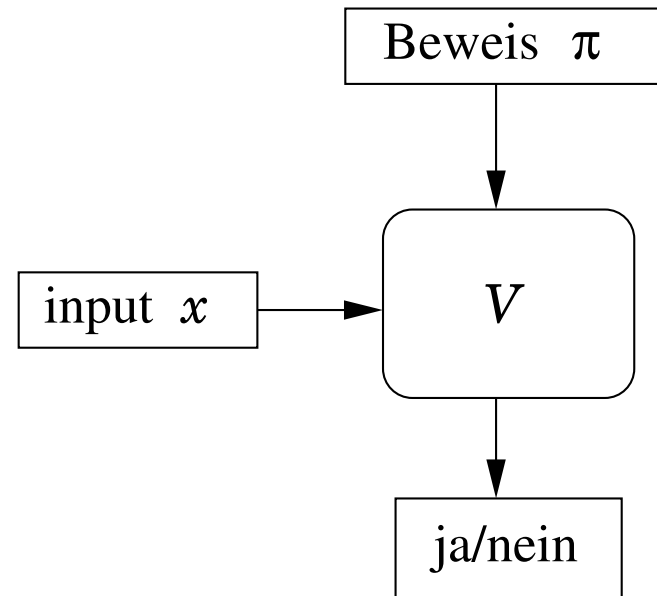
Das PCP-Theorem

Probabilistische
Algorithmen

Reduktion und
Vollständigkeit

NP = Prüfen von Beweisen

Nichtdeterministische TM (Verifikator):



Entscheidungsproblem P ist in NP gdw. es gibt Verifikator V mit

$$x \in Y_P \quad \rightarrow \quad \exists \pi \quad V(x, \pi) = \text{ja}$$

$$x \in N_P \quad \rightarrow \quad \forall \pi \quad V(x, \pi) = \text{nein}$$

Einführung

Grundlagen

Methoden zum
Entwurf

Approximationsklassen

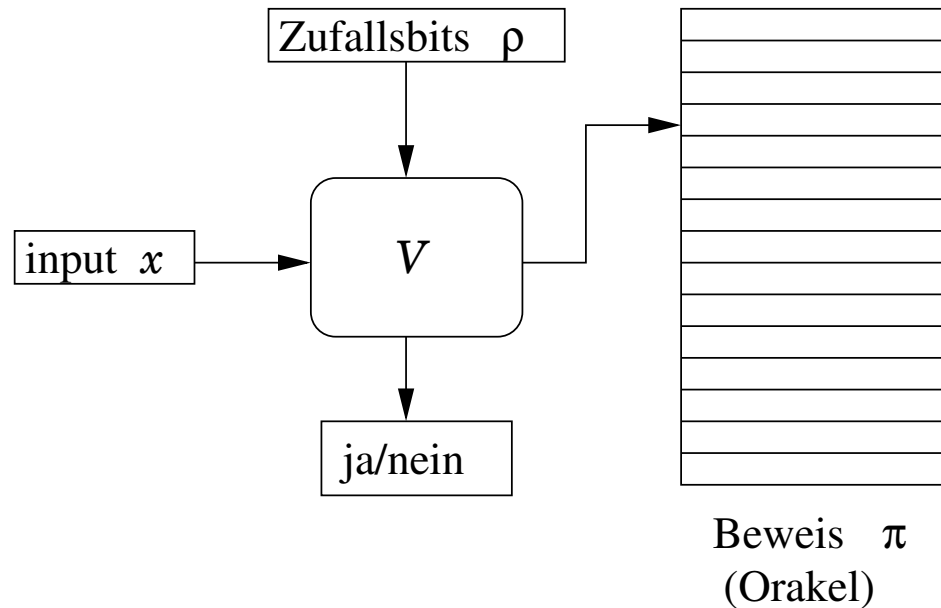
Das PCP-Theorem

Probabilistische
Algorithmen

Reduktion und
Vollständigkeit

Probabilistisch überprüfbare Beweise

Verifikator:



Verifikator V ist $(r(n), q(n))$ -beschränkt, falls bei $|x| = n$

- ▶ V liest $\leq r(n)$ Zufallsbits,
- ▶ berechnet daraus $\leq q(n)$ Orakelfragen
- ▶ entscheidet anhand der Antworten.

Einführung

Grundlagen

Methoden zum
Entwurf

Approximationsklassen

Das PCP-Theorem

Probabilistische
Algorithmen

Reduktion und
Vollständigkeit

Die Klasse PCP

Ein Entscheidungsproblem P ist in $\text{PCP}[r(n), q(n)]$ gdw. es einen $(r(n), q(n))$ -beschränkten Verifikator V gibt mit

$$x \in Y_P \quad \rightarrow \quad \exists \pi \mathbb{P}_\rho[V(x, \rho, \pi) = \text{ja}] = 1$$

$$x \in N_P \quad \rightarrow \quad \forall \pi \mathbb{P}_\rho[V(x, \rho, \pi) = \text{nein}] \geq \frac{1}{2}$$

Korollar: $\text{NP} = \text{PCP}[0, n^{O(1)}]$

$\text{co-RP} = \text{PCP}[n^{O(1)}, 0]$

sogar: $\text{NP} = \text{PCP}[O(\log n), n^{O(1)}]$

PCP-Theorem (Arora, Lund, Motwani, Safra, Sudan, Szegedy 92):

$$\text{NP} = \text{PCP}[O(\log n), O(1)]$$

[Einführung](#)[Grundlagen](#)[Methoden zum
Entwurf](#)[Approximationsklassen](#)[Das PCP-Theorem](#)[Probabilistische
Algorithmen](#)[Reduktion und
Vollständigkeit](#)

Folgerungen aus dem PCP-Theorem

[Einführung](#)[Grundlagen](#)[Methoden zum
Entwurf](#)[Approximationsklassen](#)[Das PCP-Theorem](#)[Probabilistische
Algorithmen](#)[Reduktion und
Vollständigkeit](#)

Satz: Falls $P \neq NP$, so ist $\text{MAXIMUM SAT} \notin \text{PTAS}$.

Satz: Falls $P \neq NP$, so sind

- ▶ $\text{MAXIMUM INDEPENDENT SET}$
- ▶ MAXIMUM CLIQUE
- ▶ $\text{MINIMUM VERTEX COVER}$

nicht in PTAS.

Korollar: Falls $P \neq NP$, so sind

- ▶ $\text{MAXIMUM INDEPENDENT SET}$
- ▶ MAXIMUM CLIQUE

nicht in APX

Beweis des PCP-Theorems

1. $\text{NP} \subseteq \text{PCP}[n^{O(1)}, O(1)]$

2. $\text{NP} \subseteq \text{PCP}[O(\log n), \log^{O(1)} n]$

3. Komposition:

$$\text{NP} \subseteq \text{PCP}[r_1(n), q_1(n)] \quad \text{und} \quad \text{NP} \subseteq \text{PCP}[r_2(n), q_2(n)]$$

$$\implies \text{NP} \subseteq \text{PCP}[r_1(n) + r_2(c \cdot q_1(n)), q_2(c \cdot q_1(n))]$$

Arithmetisierung

Klauseln \mapsto Polynome über \mathbb{Z}_2

$$x \mapsto p_x := (1 - x)$$

$$\bar{x} \mapsto p_{\bar{x}} := x$$

$$C = \ell_1 \vee \dots \vee \ell_k \mapsto p_C := p_{\ell_1} \cdot \dots \cdot p_{\ell_k}$$

$$\vec{r} \in \mathbb{Z}_2^m, F = C_1 \wedge \dots \wedge C_m \mapsto p_{\vec{r}} := r_1 \cdot p_{C_1} + \dots + r_m \cdot p_{C_m}$$

Lemma: Für $0 \neq \vec{v} \in \mathbb{Z}_2^m$ gilt: $\mathbb{P}_{\vec{r} \in \mathbb{Z}_2^m} [\sum_i r_i v_i = 1] = \frac{1}{2}$.

Für jede Bewertung α gilt:

$$\alpha \models C \quad \rightarrow \quad \mathbb{P}_{\vec{r} \in \mathbb{Z}_2^m} [p_{\vec{r}}(\alpha) = 0] = 1$$

$$\alpha \not\models C \quad \rightarrow \quad \mathbb{P}_{\vec{r} \in \mathbb{Z}_2^m} [p_{\vec{r}}(\alpha) = 0] = \frac{1}{2}$$

Codierung durch lineare Funktionen

Satz: Sei $a \in \mathbb{Z}_2^n$. Dann gibt es lineare Funktionen

$$A_a : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2 \quad B_a : \mathbb{Z}_2^{n^2} \rightarrow \mathbb{Z}_2 \quad C_a : \mathbb{Z}_2^{n^3} \rightarrow \mathbb{Z}_2$$

so dass \forall Polynome $p \in \mathbb{Z}_2[x_1, \dots, x_n]$ vom Grad 3 gilt

$$p(a) = \alpha_p + A_a(q_{p,1}) + B_a(q_{p,2}) + C_a(q_{p,3})$$

für geeignete $\alpha_p \in \mathbb{Z}_2$, $q_{p,1} \in \mathbb{Z}_2^n$, $q_{p,2} \in \mathbb{Z}_2^{n^2}$ und $q_{p,3} \in \mathbb{Z}_2^{n^3}$, die nur von p abhängen.

Definition: Für $x \in \mathbb{Z}_2^m$, $y \in \mathbb{Z}_2^k$ sei $x \circ y := z \in \mathbb{Z}_2^{mk}$ mit $z_{im+j} := x_i y_j$.

Aus Beweis: $A_a = a$, $B_a = a \circ a$, $C_a = a \circ (a \circ a)$.

[Einführung](#)[Grundlagen](#)[Methoden zum
Entwurf](#)[Approximationsklassen](#)[Das PCP-Theorem](#)[Probabilistische
Algorithmen](#)[Reduktion und
Vollständigkeit](#)

Funktionen $f, g : D \rightarrow R$ sind δ -nah, falls

$$\mathbb{P}_{x \in D}[f(x) \neq g(x)] \leq \delta.$$

Satz:

Für $g : \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2$ sei

$$\mathbb{P}_{x, y \in \mathbb{Z}_2^m}[g(x + y) \neq g(x) + g(y)] \leq \frac{\delta}{2}.$$

Dann ist g δ -nah zu einem *linearen* $f : \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2$.

Korrektureigenschaft

Sei $f : \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2$ linear, und $g : \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2$ δ -nah zu f .

Dann gilt für alle $x \in \mathbb{Z}_2^m$:

$$\mathbb{P}_{y \in \mathbb{Z}_2^m} [g(x + y) - g(y) = f(x)] \geq 1 - 2\delta .$$

Konsistenztest

A, B, C heißen *konsistent*, falls $\exists a$ mit $A = A_a, B = B_a, C = C_a$.

Konsistenztest:

Gegeben: A', B', C' sind δ -nah zu linearen Funktionen A, B, C

Wähle zufällig $x, x' \in \mathbb{Z}_2^n$;

$a := A(x); a' := A(x'); b := B(x \circ x')$;

if $aa' \neq b$ then reject;

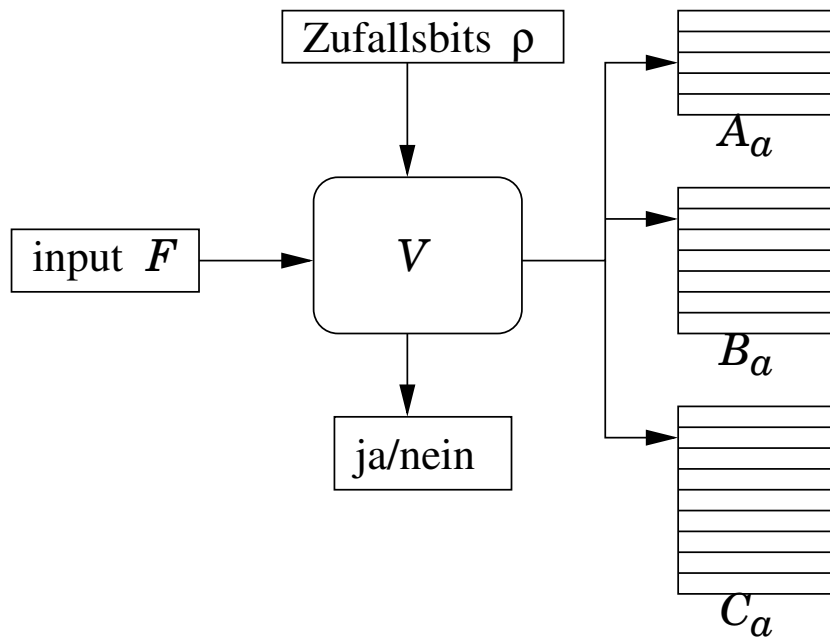
Wähle zufällig $x \in \mathbb{Z}_2^n, y \in \mathbb{Z}_2^{n^2}$;

$a := A(x); b := B(y); c := C(x \circ y)$;

if $ab \neq c$ then reject else accept;

(Berechne A, B, C jeweils mittels Korrektüreigenschaft)

Falls A, B, C konsistent sind, akzeptiert der Konsistenztest, andernfalls weist er mit hinreichender Wahrscheinlichkeit zurück.



Verifikator V :

Gegeben: Orakel A, B, C , 3CNF F
 Linearitätstest: sind A, B, C δ -nah
 zu linearen Funktion A', B', C'

Konsistenztest: gibt es a mit

$$A' = A_a, B' = B_a, C' = C_a$$

Wähle zufällig \vec{r} , setze $p := p_F^{\vec{r}}$

Berechne $\alpha_p, q_{p,1}, q_{p,2}, q_{p,3}$

$$\text{Berechne } v := \alpha_p + A_a(q_{p,1}) \\ + B_a(q_{p,2}) + C_a(q_{p,3})$$

if $v = 0$ then accept else reject

Größe des Beweises: $|A_a| + |B_a| + |C_a| = 2^{n^3} + 2^{n^2} + 2^n$

Also: Anzahl der Zufallsbits = $O(n^3)$.

Anzahl der Orakelfragen ist konstant.