Approximations-Schemata, Estimator-Theorem, Expansion des Universums und Uniformes Sampling am Beispiel von # DNF

Sommerakademie 2007

Peter Zaspel

Idee

Ziel:

Approximative Lösung von *kombinatorischen Zählproblemen*

Lösung:

Verwendung der Methode der *Expansion des Universums* und von *Randomisierung* zur Approximation

Definition (kombinatorisches Optimierungsproblem Π)

Gegeben durch:

- D: Menge der (zulässigen) Eingaben
- $\mathcal{S}(\mathcal{I})$ mit $I \in D$: Menge der zulässigen Lösungen zu Eingabe I
- $f : S(I) \to \mathbb{N}^{\neq 0}$ (Bewertungsfunktion)
- $ziel \in \{min, max\}$

Gesucht:

für $I \in D$ eine Lösung $\sigma_{opt} \in S(I)$ so dass

$$f(\sigma_{opt}) = ziel\{f(\sigma) \mid \sigma \in S(I)\}$$

$OPT(I) = f(\sigma_{opt})$ ist der Wert einer optimalen Lösung.

Definition (kombinatorisches Zählproblem $\#\Pi$)

Gegeben:
kombinatorisches Optimierungsproblem Π
Gesucht:
$\#(I) = |S(I)|$ (Anzahl zulässiger Lösungen zu Probleminstanz I)

Beispiel (Rucksack-Zählproblem $\#RUCKSACK$)

Gegeben: *RUCKSACK* (Standard-Knappsack-Problem)
Gesucht: Anzahl der Rucksackfüllungen, die maximales Füllgewicht nicht überschreiten

Beispiel (Färbungen-Zählproblem $\#COL_k$)

Gegeben: COL_k (Graph-Färbbarkeitsproblem mit k Farben)
Gesucht: Anzahl der korrekten Knotenfärbungen mit $\leq k$ Farben

Beispiel (DNF-Zählproblem $\#DNF$)

Gegeben:
Probleminstanz Ψ ist Boolsche (n,m)-Formel in DNF über n Variablen $V = \{x_1, ..., x_n\}$
Also: $\Psi = C_1 \vee ... \vee C_m$ mit C_i Monome der Länge k_i bestehend aus Literalen x_j und $\bar{x}_k$. (Pro Monom kommt eine Variable maximal einmal vor.)
Gesucht:
Anzahl der zulässigen Belegungen, d.h. Belegung $b_\Psi : V \rightarrow \{TRUE, FALSE\}$ der Variablen so, dass Ψ erfüllt wird.

Komplexität (1)

Komplexitätsklasse $\#P$

In $\#P$ sind die Zählprobleme enthalten, deren zugehörige Entscheidungsprobleme in NP liegen.
Ein Problem p in $\#P$ ist $\#P$-vollständig, wenn sich alle Probleme aus $\#P$ mittels Polynomialzeitreduktion auf p reduzieren lassen.

Lemma

Wenn $P = NP$, dann lassen sich $\#P$-vollständige Probleme in Poly-Zeit lösen.

Komplexität (2)

Satz

$\#DNF$ ist $\#P$-vollständig.

Vermutung

$\nexists$ Polynomzeit-Algorithmus für $\#DNF$.

Expansion des Universums

Idee der Expansion

- Erweiterung der Menge der erlaubten Lösungen $S(I)$ auf eine Obermenge U_I
- Berechnung der Kardinalität von U_I
- Abschätzung des Verhältnisses der Kardinalitäten

Definition (Expansion des Universums)

Gegeben:

- I Instanz von $\#\Pi$
- U_I Menge mit $S(I) \subseteq U_I$ mit $|U_I|$ bekannt
  U_I ist das *Universum* von $S(I)$

$$\xi = \frac{|U_I|}{\#(I)} = \frac{|U_I|}{|S(I)|}$$

ξ ist die *Expansion des Universums*

Definition der relativen Güte

Gegeben:

- $\#\Pi$ kombinatorisches Zählproblem
- $\#(I)$ exakte Lösung
- Approximationsalgorithmus A
- $A(I)$ approximative Lösung des Problems
  (mit $A(I) \leq \#(I)$ und $A(I) \geq \#(I)$ erlaubt)

Definition (Relative Güte ρ_A)

relative Güte der Approximation:

$$\rho_A(I) = \max\left\{\frac{A(I)}{\#(I)}, \frac{\#(I)}{A(I)}\right\}$$

$\rho_A(n)$ ist relative Güte bei Eingaben der Länge $\leq n$.

Lemma

Mit $|I| = n$ gilt:

$$\frac{1}{\rho_A(n)}\#(I) \leq A(I) \leq \rho_A(n)\#(I)$$

Expansion des Universums

$$\xi = \frac{|U_I|}{\#(I)}$$

Satz

Sei s eine obere Schranke für ξ, also: $\xi \leq s$.
Dann hat die Approximation

$$A(I) = \frac{|U_I|}{\sqrt{s}}$$

für den Wert $\#(I)$ relative Güte von $\sqrt{s}$.

Approximationsschemata

Gegeben:

- $\#\Pi$ kombinatorisches Zählproblem
- A Algorithmus der $A(I, \epsilon)$ berechnet (unter Eingabe von Instanz I von $\#\Pi$ und von ϵ mit $0 < \epsilon < 1$

Definition (polynomielles Zähl-Approximationsschema (PASC))

A ist PASC für $\#\Pi$, falls A deterministisch, Laufzeit von A $T(A) = O(poly(|I|))$ und:

$$|A(I, \epsilon) - \#(I)| \leq \epsilon \#(I)$$

Definition (*streng* polynomielles Zähl-Approximationsschema (FPASC))

A ist PASC und $T(A) = O(poly(|I|, \frac{1}{\epsilon}))$

Definition (polynomielles randomisiertes Zähl-Approximationsschema (PRASC))

A hat Laufzeit $T(A) = O(poly(|I|))$ und es gilt:

$$Pr[|A(I,\epsilon) - \#(I)| \leq \epsilon \cdot \#(I)] \geq \frac{3}{4}$$

Definition (*streng* polynomielles randomisiertes Zähl-Approximationsschema (FPRASC))

A ist PRASC und hat Laufzeit $T(A) = O(poly(|I|, \frac{1}{\epsilon}))$

Definition ((ϵ,δ)-FPRASC)

A ist ein FPRASC mit zusätzlicher Eingabe δ mit $0 < \delta < 1$, hat Laufzeit $T(A) = O(poly(|I|, \frac{1}{\epsilon}, \log(\frac{1}{\delta})))$ und erzeugt eine Ausgabe $A(I,\epsilon,\delta)$ mit:

$$Pr[|A(I,\epsilon,\delta) - \#(I)| \leq \epsilon \cdot \#(I)] \geq 1 - \delta$$

Wahrscheinlichkeitsverstärkung

Es ist möglich, aus einem FPRASC ein (ϵ,δ)-FPRASC zu machen. Dies geschieht durch den $\Theta(\log(\frac{1}{\delta}))$-maligen Aufruf vom FPRASC-Algorithmus:

Algorithmus AMPL($A;\epsilon,\delta,I$)

for τ:= 1 to T_δ do

  N_τ := $A(I,\epsilon)$;

return $\frac{1}{T_\delta} \cdot \sum_{\tau=1}^{T_\delta} N_\tau$

Algorithmus AMPL(A;ϵ,δ,I)

```
for τ := 1 to T_δ do
      N_τ := A(I, ε);
return (1/T_δ) · Σ_{τ=1}^{T_δ} N_τ
```

for τ := 1 to T_δ do
$\quad N_\tau := A(I,\epsilon)$;
return $\frac{1}{T_\delta} \cdot \sum_{\tau=1}^{T_\delta} N_\tau$

Satz (Wahrscheinlichkeitsverstärkung)

Gegeben:

- *$\#\Pi$ kombinatorisches Zählproblem*
- *A ein FPRASC für $\#\Pi$*
- $\delta < 1$

Mit $T_\delta = 8\lceil \ln(\frac{1}{\delta}) \rceil$ ist AMPL(A;ϵ,δ,I) ein (ϵ,δ)-FPRASC für $\#\Pi$.

Monte-Carlo-Methode - Voraussetzungen

Gegeben:

- $\#\Pi$ kombinatorisches Zählproblem
- U_I Universum zu jeder Instanz I mit bekanntem $|U_I|$
  (Nenne hier U_I *Stichprobenraum*.)
- $\xi = \frac{|U_I|}{\#(I)}$ Expansion des Universums
- T Anzahl an Wiederholungen im Monte-Carlo-Algorithmus
- $\chi : U_I \to \{0, 1\}$ charakteristische Funktion von $S(I)$, mit:

$$\chi(u) = \begin{cases} 1 & \textit{falls } u \in S(I) \\ 0 & \textit{sonst.} \end{cases}$$

Forderungen

1. Es gibt einen Algorithmus UG der mit Laufzeit $O(Poly(|I|))$ Elemente (*Stichproben*) u aus U_I ausgibt, so dass für alle $u \in U_I$ gilt:

$$Pr[u\ wird\ von\ UG\ ausgegeben] = \frac{1}{|U_I|}$$

   Dieser Algorithmus heißt *Stichprobengenerator*.

2. Es gibt einen Algorithmus BEANTWORTER, der χ in Zeit $O(Poly(|I|))$ berechnet.

Monte-Carlo-Algorithmus MC(T)

for $i := 1$ to T do
    (1) ziehe eine Stichprobe $u \in U_I$ mittels UG;
    (2) $Y_i := \chi(u)$ mittels BEANTWORTER
done;
$R := \frac{1}{T} \cdot \sum_{i=1}^{T} Y_i$;
gib $Z := R \cdot |U_I|$ aus.

Lemma

1. $E[R] = \xi^{-1}$ *($E[Y_i] = \xi^{-1}$ folgt aus Uniformität $\Rightarrow$ Beh.)*
2. $E[MC(T)] = \#(I)$ *($E[MC(T)] = E[Z] = \xi^{-1} \cdot |U_I| = \#(I)$)*

Wie groß muss T gewählt werden, damit der Fehler klein wird?

Lemma

1. $Var[R] = \frac{\xi^{-1} \cdot (1 - \xi^{-1})}{T}$
2. $Var[MC(T)] = |U_I|^2 \cdot \frac{\xi^{-1} \cdot (1 - \xi^{-1})}{T}$

Lemma

1. $Var[R] = \frac{\xi^{-1}\cdot(1-\xi^{-1})}{T}$
2. $Var[MC(T)] = |U_I|^2 \cdot \frac{\xi^{-1}\cdot(1-\xi^{-1})}{T}$

Beweis.

1. Y_i sind 0-1-Zufallsvariablen
   $\Rightarrow Var[Y_i] = E[Y_i]\cdot(1 - E[Y_i]) = \xi^{-1}\cdot(1-\xi^{-1})$
   $\Rightarrow Var[R] = Var\left[\frac{1}{T}\cdot\sum_{i=1}^{T} Y_i\right] = \frac{1}{T^2}\cdot\sum_{i=1}^{T} Var[Y_i] = \frac{\xi^{-1}\cdot(1-\xi^{-1})}{T}$
2. $Var[MC(T)] = Var[|U_I|\cdot R] = |U_I|^2\cdot Var[R] = |U_I|^2\cdot\frac{\xi^{-1}\cdot(1-\xi^{-1})}{T}$

□

Satz (Estimator-Theorem der Monte-Carlo-Methode)

Gegeben: $\epsilon > 0$, *beliebig, aber fest*
Mit $T_\xi(\epsilon) = \lceil \frac{4}{\epsilon^2} \cdot (\xi - 1) \rceil$ *gilt:*

$$Pr[|MC(T_\xi(\epsilon)) - \#(I)| \leq \epsilon \cdot \#(I)] \geq \frac{3}{4}$$

Beweis.

$Pr[|MC(T_\xi(\epsilon)) - \#(I)| \geq \epsilon \cdot \#(I)]$

$$\overset{Tschebyscheff}{\leq} \frac{1}{\epsilon^2} \cdot \frac{Var[MC(T_\xi(\epsilon)]}{E[MC(T_\xi(\epsilon)]^2} \overset{Lemmata}{=} \frac{1}{\epsilon^2} \cdot \frac{|U_I|^2 \cdot \frac{\xi^{-1} \cdot (1-\xi^{-1})}{T_\xi(\epsilon)}}{\#(I)^2}$$

$$\overset{Def.\xi}{=} \frac{1}{\epsilon^2} \cdot \xi^2 \cdot \frac{\xi^{-1} \cdot (1-\xi^{-1})}{T_\xi(\epsilon)} = \frac{1}{\epsilon^2} \cdot (\xi - 1) \cdot \frac{1}{T_\xi(\epsilon)} \overset{Def.T_\xi(\epsilon)}{\leq} \frac{1}{4}$$

□

Satz (Estimator-Theorem der Monte-Carlo-Methode)

Gegeben: $\epsilon > 0$, *beliebig, aber fest*
Mit $T_\xi(\epsilon) = \lceil \frac{4}{\epsilon^2} \cdot (\xi - 1) \rceil$ *gilt:*

$$Pr[|MC(T_\xi(\epsilon)) - \#(I)| \leq \epsilon \cdot \#(I)] \geq \frac{3}{4}$$

Bemerkung

- Das Theorem liefert mit $MC(T_\xi(\epsilon))$ *nicht* direkt für jedes Zählproblem, das Universum mit uniformem Generator und Beantworter ein FPRASC.
- Problem: $T_\xi(\epsilon)$ linear in ξ
  $\rightarrow$ Abhängigkeit von dem Wert $\#(I)$
- Deshalb: Abschätzung für ξ nötig.

Anwendungsbeispiel:

DNF

Zutaten

- „vernünftiges“ Universum für $S(I)$
- Abschätzung für ξ
- BEANTWORTER-Algorithmus
- Uniformer Generator für das Universum
- Monte-Carlo-Algorithmus

Ergebnis

Es gibt ein (ϵ,δ)-FPRASC für $\#DNF$ mit der Laufzeit $O(const \cdot n \cdot \frac{1}{\epsilon} \cdot \log(\frac{1}{\delta}))$

Wiederholung

Beispiel (DNF-Zählproblem $\#DNF$)

Gegeben:

Probleminstanz Ψ ist Boolsche (n,m)-Formel in DNF über n Variablen $V = \{x_1, ..., x_n\}$

Also: $\Psi = C_1 \vee ... \vee C_m$ mit C_i Monome der Länge k_i bestehend aus Literalen x_j und $\bar{x}_k$. (Pro Monom kommt eine Variable maximal einmal vor.)

Gesucht:

Anzahl $\#(\Psi)$ der zulässigen Belegungen, d.h. Belegung $u : V \rightarrow \{TRUE, FALSE\}$ der Variablen so, dass Ψ erfüllt wird ($u(\Psi)$=TRUE).

Wahl des Universums für # DNF (1)

Wichtig

1. Erfüllende Belegungen für Ψ einfach bestimmbar, da die Erfüllung einer Klausel ausreicht
2. Anzahl $\#(C_j)$ der erfüllenden Belegungen einer einzelnen Klausel C_j unmittelbar berechenbar

Zu (1): Wähle erfüllende Belegung für eine Klausel und würfele verbleibende, unbelegte Variablen aus.

Zu (2):

Lemma

Gegeben: *$C = l_1 \wedge \ldots \wedge l_k$ Klausel der Boolschen (n,m)-Formel Ψ in DNF aus k Literalen.*
Dann gilt: Es gibt genau 2^{n-k} Belegungen, die C erfüllen.
$\Rightarrow \#(C) = 2^{n-k}$.

Wahl des Universums für # DNF (2)

Neue Menge mit gleicher Kardinalität wie $S(\Psi)$

$$
\begin{aligned}
S(\Psi) &= \bigcup_{j=1}^{m} \{u \mid u\ erfüllt\ C_j\} \\
&= \bigcup_{j=1}^{m} \{u \mid u\ erfüllt\ C_j, aber\ kein\ C_k, k < j\} \\
S'(\Psi) &= \bigcup_{j=1}^{m} \{(u,j) \mid u\ erfüllt\ C_j, aber\ kein\ C_k, k < j\}
\end{aligned}
$$

$$\#(\Psi) = |S(\Psi)| = |S'(\Psi)|$$

Wahl des Universums für # DNF (3)

$$S'(\Psi) = \bigcup_{j=1}^{m} \{(u, j) | u \ erf\ddot{u}llt \ C_j, aber \ kein \ C_k, k < j\}$$

Universum

$$\begin{aligned} U_\Psi &= \{(u, j) | u \ erf\ddot{u}llt \ C_j\} \\ &= \dot{\bigcup_{j=1}^{m}} \{(u, j) | u \ erf\ddot{u}llt \ C_j\} \end{aligned}$$

Es gilt: $S'(\Psi) \subseteq U_\Psi$

$$U_\Psi = \dot{\bigcup_{j=1}^{m}} \{(u,j) | u \text{ erfüllt } C_j\}$$

mit Lemma folgt:

Kardinalität des Universums

$$|U_\Psi| = \sum_{j=1}^{m} 2^{n-k_j}$$

Abschätzung für ξ

Lemma (Expansionslemma)

$$\xi = \frac{|U_\Psi|}{|S'(\Psi)|} \leq m$$

(m die Anzahl der Klauseln)

Beweis.

1. $U_\Psi = \dot\bigcup_{j=1}^{m} \{(u,j)|u\ erf\ddot{u}llt\ C_j\}$ enthält Tupel aus allen erfüllenden Belegungen u und Werten j mit $j = 1, ..., m$
2. Anzahl der erfüllenden Belegungen: $|S(\Psi)|$

$\Rightarrow |\mathbf{U_\Psi}| \leq \mathbf{m} \cdot |\mathbf{S(\Psi)}|$ □

BEANTWORTER-Algorithmus

Gesucht

Deterministischer Polynom-Zeit-Algorithmus zur charakteristischen Funktion χ von $S'(\Psi)$ mit

$$\chi((u,j)) = \begin{cases} 1 & falls\ j = \min\{k | u\ erf\ddot{u}llt\ C_k\} \\ 0 & sonst. \end{cases}$$

Algorithmus

Einfaches Auswerten der DNF nötig

$\Rightarrow$ deterministisch in Zeit $O(m \cdot n)$ möglich

Uniformer Generator für das Universum (1)

Algorithmus UG

1. würfele ein $j \in \{1, \ldots, m\}$ mit Wahrscheinlichkeit $\frac{2^{n-k_j}}{|U_\Psi|}$;
2. setze die in C_j auftauchenden Variablen so, dass C_j wahr wird;
3. würfele eine Belegung der restlichen Variablen;
4. gib j und die teils berechnete, teils gewürfelte Belegung u aus.

Lemma

Algorithmus UG ist ein uniformer Generator für U_Ψ.

Uniformer Generator für das Universum (2)

Lemma

Algorithmus UG ist ein uniformer Generator für U_Ψ.

Beweis.

$Pr[(u,j) \in U_\Psi \textit{ wird gewählt}]$
$= Pr[j \in \{1,..,m\} \textit{ wird in Zeile } 1 \textit{ ausgewürfelt}$
$\quad \wedge \textit{ die } C_j \textit{ erfüllende Belegung } u \textit{ wird ausgegeben}]$

Mit Wahrscheinlichkeit $\frac{2^{n-k_j}}{|U_\Psi|}$ für j und Wahrscheinlichkeit $\frac{1}{2^{n-k_j}}$ für tatsächliche Wahl von u, folgt:

$$Pr[(u,j) \in U_\Psi \textit{ wird gewählt}] = \frac{2^{n-k_j}}{|U_\Psi|} \cdot \frac{1}{2^{n-k_j}} = \frac{\mathbf{1}}{|\mathbf{U_\Psi}|}$$

$\square$

Monte-Carlo-Algorithmus

Wiederholung:

Forderungen

1. uniformer Stichprobengenerator für das Universum
2. Algorithmus BEANTWORTER, der χ berechnet

Monte-Carlo-Algorithmus anwendbar

Satz

Mit $T(\epsilon, \delta) = \lceil m \cdot \frac{4}{\epsilon^2} \ln \frac{2}{\delta} \rceil$ *ist Algorithmus* $MC(T(\epsilon, \delta))$ *ein* (ϵ, δ)*-FPRASC für #DNF der Laufzeit* $O(m^2 \cdot n \cdot \frac{1}{\epsilon} \cdot \log \frac{1}{\delta})$.

Satz

Mit $T(\epsilon, \delta) = \lceil m \cdot \frac{4}{\epsilon^2} \ln \frac{2}{\delta} \rceil$ *ist Algorithmus* $MC(T(\epsilon, \delta))$ *ein* (ϵ, δ)*-FPRASC für* $\#DNF$ *der Laufzeit* $O(m^2 \cdot n \cdot \frac{1}{\epsilon} \cdot \log \frac{1}{\delta})$.

Beweis.

1. Verwende Monte-Carlo-Algorithmus
2. Estimator-Theorem liefert Abschätzung für Fehlerwahrscheinlichkeit

$$Pr[|MC(T_\xi(\epsilon)) - \#(I)| \leq \epsilon \cdot \#(I)] \geq \frac{3}{4}$$

3. Abschätzung für Expansion (*Expansionslemma*) macht aus dem Monte-Carlo-Algorithmus ein FPRASC
4. Anwendung des Satzes über die Wahrscheinlichkeitsverstärkung macht aus FPRASC ein (ϵ, δ)-FPRASC.

□

Zusammenfassung

Herausforderung

Kombinatorisches Zählproblem approximativ lösen

Methoden

- Expansion des Universums
- Monte-Carlo-Algorithmus
  - Uniformes Sampling
  - BEANTWORTER

Zentrales Ergebnis

Estimator-Theorem

Anwendung

Approximations-Algorithmus für $\#DNF$

Fragen?