UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Ambrož Homar

Izrek o verjetnostnem preverjanju dokazov

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN MATEMATIKA

Mentor: prof. dr. Borut Robič

Ljubljana, 2011

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00002/2011
Datum: 01.09.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko ter Fakulteta za matematiko in fiziko izdaja naslednjo nalogo:

Kandidat: **AMBROŽ HOMAR**

Naslov: **IZREK O VERJETNOSTNEM PREVERJANJU DOKAZOV**
THE PCP THEOREM

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

Predstavite izrek o verjetnostnem preverjanju dokazov s stališča dokaznih sistemov in zahtevnosti aproksimacije in pokažite ekvivalenco obeh verzij izreka. Omenite glavne ideje pri algebraičnem in kombinatoričnem dokazu. Navedite primer aproksimacijskega rezultata, ki je bil dobljen na podlagi izreka o verjetnostnem preverjanju dokazov. Predstavite zgodovinski razvoj na področju verjetnostnih dokazovalnih sistemov.

Mentor:
prof. dr. Borut Robič

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan Fakultete za računalništvo in informatiko:
prof. dr. Nikolaj Zimic

Dekan Fakultete za matematiko in fiziko:
prof. dr. Andrej Likar

UNIVERZA V LJUBLJANI • Fakulteta za matematiko in fiziko • 6

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Ambrož Homar,

z vpisno številko 63080031,

sem avtor diplomskega dela z naslovom:

Izrek o verjetnostnem preverjanju dokazov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
  prof. dr. Boruta Robiča
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 5.9.2011 Podpis avtorja:

Zahvala

Zahvaljujem se mentorju prof. dr. Borutu Robiču za nasvete, popravke in prijaznost pri pisanju diplomskega dela.

Zahvaljujem se prof. dr. Andreju Bauerju za pojasnilo glede afinih in linearnih predikatov.

Zahvaljujem se družini, ki me je podpirala v času študija.

Kazalo

Seznam kratic in simbolov

PCP - Probabilistic Checking of Proofs, slovensko: verjetnostno preverjanje dokazov

TM - Turing Machine, slovensko: Turingov stroj

PTM - Probabilistic Turing machine, slovensko: verjetnostni Turingov stroj

RAM - Random Access Memory, slovensko: pomnilnik z enakovrednim dostopom

poly(n) - polynomial, slovensko: polinom spremenljivke n

CSP - Constraint Satisfaction Problem, slovensko: problem izpolnjevanja omejitev

Gap-CSP - Gap Constraint Satisfaction Problem, slovensko: problem izpolnjevanja omejitev z vrzeljo

3-CNF - 3-Conjunctive Normal Form, slovensko: konjunktivna normalna oblika, v kateri je vsak faktor tročlena disjunkcija

3-SAT - 3-Satisfiability Problem, slovensko: problem izpolnljivosti Boolovega izraza v obliki 3-CNF

MAX-3-SAT - Maximum-3-Satisfiability Problem, slovensko: problem največje izpolnljivosti Boolovega izraza v obliki 3-CNF

3-COL - 3-Coloring Problem, slovensko: problem 3-barvanja grafa

GNI - Graph Non-Isomorphism Problem, slovensko: problem neizomorfnih grafov

MAX-INDSET - Maximum Independent Set Problem, slovensko: problem največje neodvisne podmnožice

MIN-VERTEX-COVER - Minimum Vertex Cover Problem, slovensko: problem najmanjšega pokritja grafa

LTC - Locally Testable Codes, slovensko: lokalno preverljive kode

IP - Interactive Proofs (Class), slovensko: razred problemov, za katere obstajajo interaktivni dokazovalni sistemi

PSPACE - Polynomial Space (Class), slovensko: razred odločitvenih problemov, ki jih lahko rešijo Turingovi stroji s polinomsko prostorsko omejitvijo

MIP - Multi-prover Interactive Proofs (Class), slovensko: razred problemov, za katere obstajajo interaktivni dokazovalni sistemi z več dokazovalniki

NEXP - Non-deterministic Exponential (Class), slovensko: razred odločitvenih problemov, ki jih lahko rešijo nedeterministični Turingovi stroji z eksponentno časovno omejitvijo

Povzetek

Rešitve NP-problemov lahko deterministično preverjamo v polinomskem času. Vendar pa ima običajno preverjanje pomanjkljivost - tipično moramo pregledati celoten dokaz o pravilnosti rešitve, da lahko utemeljeno trdimo, da je pravilna oz. napačna. V diplomskem delu predstavljamo izrek o verjetnostnem preverjanju dokazov, ki trdi, da lahko rešitve NP-problemov učinkovito preverimo že z majhnim številom dostopov do bitov v dokazih njihove pravilnosti. Pokažemo tudi ekvivalentno verzijo izreka, na podlagi katere je bilo dokazano mnogo aproksimacijskih pragov za NP-težke probleme. Navajamo primer problemov največje neodvisne množice in najmanjšega pokritja grafa. Predstavljamo glavne ideje pri prvotnem algebraičnem dokazu izreka PCP in elegantnejšem kombinatoričnem dokazu. V zadnjem poglavju navajamo razburljivo zgodovino odkritij na področju verjetnostnega preverjanja dokazov.

Ključne besede:

računska zahtevnost, verjetnostno preverjanje dokazov, zahtevnost aproksimacije, aproksimacijski algoritmi

Abstract

Solutions to NP-problems are deterministically verifiable in polynomial time. But the classic verification process has a drawback - typically we need to (at least) read the entire proof to decide whether the proof is correct or incorrect. In the thesis we present the PCP theorem which claims that solutions to NP-problems can be checked by only a small number of queries to bits in their corresponding proof strings. We describe the equivalent version of the theorem which is at the heart of many approximation results for NP-hard problems. We illustrate this on two well-known problems, the maximum independent set problem and the minimum vertex cover problem. We present the main ideas of both versions of the PCP theorem proof: the original algebraic proof and later combinatorial version. In the last chapter we list some exciting discoveries related to the probabilistic checking of proofs.

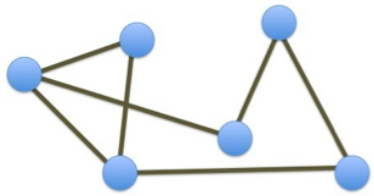
Key words:

computational complexity, probabilistic checking of proofs, hardness of approximation, approximation algorithms

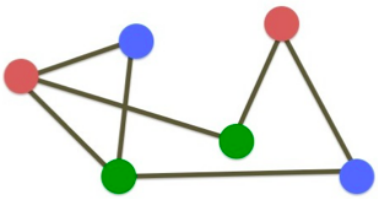
Poglavje 1

Uvod

Denimo, da sta nam predstavljeni situaciji na Slikah 1.1, 1.2. Na prvo lahko gledamo kot na trditev, druga pa bi lahko predstavljala njen dokaz.



Slika 1.1: Trditev: Dani graf je 3-obarvljiv.



Slika 1.2: Dokaz: 3-barvanje podanega grafa.

Veljavnost same trditve je težko preverjati, saj to od nas zahteva konstrukcijo dokaza. Naloga se zdi precej lažja, če nam je predstavljena tako trditev, kot tudi dokaz za njeno pravilnost. Preprosto pregledamo dokaz od začetka

do konca in če najdemo napako, dokaz proglasimo za neveljaven. Če pa pregledamo celoten dokaz in napake ne najdemo, ugotovimo, da trditev velja.

Vprašamo pa se lahko naslednje: ali lahko veljavnost trditve preverimo tudi kako drugače? Bolj učinkovito? Lahko o pravilnosti dokaza kaj rečemo, ne da bi nam ga bilo potrebno preučiti v celoti?

Primer. Poglejmo si naslednjo življenjsko situacijo, kjer je uporabljen drugačen način dokazovanja [1]. Marko ima dve nogavici, zeleno in rdečo. Njegova prijateljica Ana je barvno slepa in mu ne verjame, da sta nogavici različnih barv. Kako naj se Ana prepriča, da Marko govori resnico? Dokaz (različni barvi) ni podan v njej razumljivi obliki. O Markovi trditvi se lahko prepriča le s pametnim postavljanjem vprašanj, na katera ji mora odgovoriti.

Kako torej ravna Ana? V vsako roko vzame po eno nogavico in vpraša Marka, kakšne barve je leva oz. desna. Nato se obrne stran od njega, vrže kovanec in izvede potezo glede na izid meta. Bodisi v rokah zamenja nogavici bodisi ju ohrani v trenutni poziciji. Nato ponovi vprašanje o barvi nogavic v posamezni roki. Če bi bili nogavici iste barve, bi Marko lahko le z verjetnostjo 1/2 uganil ali je Ana nogavici zamenjala ali ne. Tako je lahko Ana z vsakim metom kovanca bolj prepričana o resnicoljubnosti svojega prijatelja. Verjetnost, da se Marko laže, je namreč po vsakem pravilnem odgovoru dvakrat manjša (po desetih pravilnih odgovorih že pod 0.001).

Vzemimo zgornji primer za osnovo razmišljanja o drugačnem preverjanju dokazov. *Ali lahko poljuben (matematični) dokaz podamo v obliki, ki omogoča učinkovito verjetnostno preverjanje njegove pravilnosti?*

Izrek o verjetnostnem preverjanju dokazov (PCP) zagotavlja, da tak format dokaza obstaja, vendar je bistveno drugačen od klasičnega. Zaenkrat povejmo le, da je predstavljen v obliki zakodiranega dvojiškega niza, ki omogoča preverjanje pravilnosti že z branjem majhnega števila naključno izbranih bitov.

Posledice izreka PCP pa niso omejene le na konstrukcijo **lokalno preverljivih**

dokazovalnih sistemov (in s tem povezano novo klasifikacijo jezikov). Iz izreka sledijo tudi presenetljive ugotovitve na področju **zahtevnosti aproksimacije**.

V diplomskem delu želimo predstaviti izrek PCP in njegove posledice, skico dokaza ter zgodovinski razvoj od začetkov interaktivnih dokazovalnih sistemov do kombinatoričnega dokaza Irit Dinur.

Poglavje 2

Izrek PCP kot vir drugačnih dokazovalnih sistemov in nove karakterizacije razreda NP

2.1 Preverjanje dokazov

Razred NP-problemov je znan po tem, da je konstrukcija rešitve zanje včasih težavna oziroma dolgotrajna, po drugi strani pa lahko pravilnost problemu pripadajoče rešitve enostavno preverimo. Čeprav je preverjanje rešitve učinkovito (izvedljivo v polinomskem času), pa ima pomanjkljivost. Tipično moramo prebrati celoten dokaz o pravilnosti podane rešitve, da jo lahko brez dvomov razglasimo za pravilno. Razlika med pravilnim dokazom in napačnim dokazom je lahko zelo majhna.

Vzemimo za primer problem 3-SAT. Lahko se zgodi, da neka prireditev vrednosti spremenljivk zagotovi, da so vsi faktorji izraza, razen enega, pravilni. O njegovi napačnosti se ne bomo mogli prepričati drugače, kot da preverimo vsak del dokaza (pravilnost vsakega faktorja).

Izrek PCP pa pravi naslednje: dokaze je mogoče zapisati v taki obliki, da bomo

napake (če so prisotne) lahko zaznali **praktično povsod v dokazu** in ne le lokalno. Izrek pravi, da lahko naključno izberemo **konstantno število bitov** iz dokaza in z veliko verjetnostjo bomo lahko trdili, da je dokaz (rešitev NP-problema) pravilen oz. napačen. Preden nadaljujemo, natančneje opredelimo pomembne pojme in elemente **dokazovalnega sistema**.

2.2 NP-jeziki in preverjevalnik PCP

Definicija 1. *Jezik $L \subseteq \{0,1\}^*$ je v razredu NP, če obstaja polinom p: $\mathbb{N} \to \mathbb{N}$ in polinomsko časovno omejen Turingov stroj* M *(t.i.* **preverjevalnik***), tako da za vsak $x \in \{0,1\}^*$ velja: $x \in L \iff \exists u \in \{0,1\}^{p(|x|)}$: $M(x,u) = 1$.*

Nizu u pravimo tudi **certifikat** *ali dokazilo za x glede na jezik L in stroj M.*

Prepišimo definicijo v obliko, ki bo primerna za sklicevanje pri definiciji preverjevalnika PCP.

Definicija 2. *Označimo z $M^\pi(x)$ izhod Turingovega stroja M z dostopom do certifikata π za x. Jezik $L \subseteq \{0,1\}^*$ je v razredu NP, če obstaja polinomsko časovno omejen Turingov stroj V, ki za poljuben vhodni niz x preverja certifikate, ali dokazujejo trditev $x \in L$. Veljata naslednji ekvivalenci:*

$$x \in L \iff \exists \pi\text{: } V^\pi(x) = 1$$
$$x \notin L \iff \forall \pi\text{: } V^\pi(x) = 0.$$

Poglejmo, kaj moramo v zgornji definiciji dodati/spremeniti, da uvedemo nov dokazovalni sistem. Preverjevalnik V naj bo **verjetnostni Turingov stroj z enakovrednim dostopom** do certifikata π. Torej lahko neposredno preverja vsak bit certifikata preko posebnega naslovnega traku. Ta je logaritemske dolžine (glede na dolžino certifikata), kar omogoča preverjanje certifikatov eksponentne dolžine v polinomskem času.

Dostop poteka takole: Če preverjevalnik želi dostopati do i-tega bita v certifikatu π, na naslovni trak binarno zapiše i in prejme vrednost bita $\pi[i]$. Take

dostope do bitov preverjevalnik obravnava kot **dragocen vir** in želi karseda zmanjšati njihovo število.

Obstajajo adaptivni preverjevalniki, ki informacijo o že prebranih bitih uporabijo pri izboru naslednje poizvedbe, vendar se pri večini izrekov povezanih s PCP lahko omejimo na preverjevalnike, pri katerih je vsaka poizvedba neodvisna od predhodnjih. Definirajmo torej formalno, kaj je **preverjevalnik PCP**.

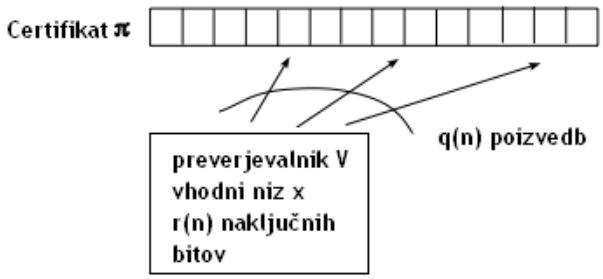
Definicija 3. *Naj bo $L \in \{0,1\}^*$ poljuben jezik in q, $r : \mathbb{N} \to \mathbb{N}$ funkciji. Za jezik L pravimo, da ima $[r(n), q(n)]$-preverjevalnik PCP, če obstaja polinomsko časovno omejen verjetnostni algoritem V z naslednjimi lastnostmi:*

1. **Učinkovitost (Efficiency)***: Naj bo stroju V dan niz $x \in \{0,1\}^*$ in omogočen enakovredni dostop do certifikata $\pi \in \{0,1\}^*$ dolžine največ $q(n)2^{r(n)}$. Stroj V lahko uporabi največ $r(n)$ naključnih bitov in izvede največ $q(n)$ neodvisnih poizvedb bitov v nizu π (Slika 2.1). Potem vrne 0 ali 1 oz. zavrne ali sprejme dokaz.*

2. **Popolnost (Completeness)***: Če je niz $x \in L$, potem obstaja certifikat $\pi \in \{0,1\}^*$, da velja $Pr[V^\pi(x) = 1] = 1$. Z drugimi besedami, če niz x pripada jeziku L, mora obstajati certifikat, ki ga bo preverjevalnik vedno sprejel.*

3. **Zdravje (Soundness):** *Če $x \notin L$, potem je za vsak certifikat $\pi \in \{0,1\}^*$ $Pr[V^\pi(x) = 1] \leq \rho$, za nek $\rho < 1$. Oziroma, če niz x ne pripada jeziku L, potem bo preverjevalnik vsak certifikat zavrnil z verjetnostjo vsaj $1 - \rho$.*

Za jezik L pravimo, da je v razredu $PCP[r(n), q(n)]$, če obstajata konstanti $c, d > 0$, tako da za L obstaja $[c \cdot r(n), d \cdot q(n)]$-preverjevalnik PCP.



Slika 2.1: Preverjevalnik lahko izvede do $q(n)$ poizvedb v certifikatu in pri tem uporabi do $r(n)$ naključnih bitov.

Opombi:

- razred NP je po definiciji PCP$[0, poly(n)]$, saj lahko dokaze o pripadnosti deterministično preverimo v polinomskem času, torej brez uporabe naključnih bitov [3].
- Za konstanto ρ se ponavadi vzame $1/2$, vendar je važno le, da je manjša od 1. Vse vrednosti $\rho \in (0, 1)$ določajo isti razred jezikov. Primer: $[c \cdot r, c \cdot q]$-preverjevalnik PCP z $\rho = 2^{-c}$ dobimo iz $[r, q]$-preverjevalnika PCP z $\rho = 1/2$ preprosto tako, da njegovo izvajanje ponovimo c-krat [1].

Primer. Poglejmo si skico delovanja preverjevalnika za jezik GNI, ki ga sestavljajo pari (opisov) neizomorfnih grafov. Pripadajoči odločitveni problem je zelo težek. Po PCP klasifikaciji spada v razred PCP[poly(n),1] [1]. Vhod predstavlja par grafov z n vozlišči $\langle G_0, G_1 \rangle$. Preverjevalnik v certifikatu π za vsak graf H z n vozlišči pričakuje bit $\pi[H] \in \{0, 1\}$, ki označuje ali je izomorfen G_0 oziroma G_1. V primeru, da ne velja nič od naštetega, je vrednost bita poljubna. π je tako niz eksponentne dolžine, ki vsebuje tak bit za vsak graf na n vozliščih.

Preverjevalnik naključno izbere bit $b \in \{0, 1\}$ in generira naključno permutacijo dolžine n. To permutacijo uporabi nad vozlišči grafa G_b in dobi izomorfen graf G'_b. Preveri ustrezni bit v certifikatu π in dokaz sprejme natanko tedaj, ko je

bit enak b.

Če velja $G_0 \not\equiv G_1$, očitno obstaja π, ki ga bo preverjevalnik sprejel z verjetnostjo 1. V nasprotnem primeru bo kakršenkoli certifikat preverjevalnik lahko prepričal z verjetnostjo največ 1/2.

2.3 Izrek PCP

Izrek 1. $NP = PCP[log\, n,\, 1]$.

Izrek PCP trdi nekaj zares presenetljivega. Poglejmo si že omenjeni problem 3-SAT. Izrek trdi, da zanj obstaja preverjevalnik PCP, ki lahko za dan izraz φ v obliki 3-CNF iz certifikata π prebere samo konstantno število bitov. Kaj torej piše v teh bitih? Gotovo ne predstavljajo očitnega dokaza - prireditve vrednosti vsem spremenljivkam v izrazu. Izrazi, ki jim le malo manjka do izpolnljivosti, bi preverjevalnik zlahka prepričali, da so izpolnljivi.

Izrek trdi, da je za vsak NP-problem število bitov, ki jih potrebuje preverjevalnik omejeno z $O(1)$. Konstante se za posamezne probleme razlikujejo, lahko pa rečemo, da so omejene s konstanto problema 3-SAT [1], saj za vsak NP-problem obstaja polinomska prevedba nanj.

Poglavje 3

PCP in zahtevnost aproksimacije

3.1 Aproksimacijski algoritmi

Ker je v praksi mnogo optimizacijskih problemov NP-težkih, se je povsem naravno pojavila ideja o algoritmih, ki bi probleme reševali suboptimalno, vendar bi za tako reševanje rabili le polinomski čas.

Glavno vprašanje, ki se pri tem postavi je seveda, kako dobro aproksimacijo lahko dosežemo, če naj algoritem ostane **polinomsko časovno omejen**. V osemdesetih letih je bilo razvitih mnogo aproksimacijskih algoritmov za različne NP-težke probleme. Poznali niso nobenega natančnega **aproksimacijskega praga** in zdelo se je celo, da je mogoče vsak NP-težek problem aproksimirati do poljubne natančnosti [8].

3.2 Povezava z izrekom PCP

Feige in sodelavci so v [10] odkrili povezavo med izrekom PCP in zahtevnostjo aproksimacije, ki je popolnoma spremenila to področje. Za veliko znanih aproksimacijskih algoritmov so preko izreka PCP odkrili, da pravzaprav že

dosegajo najboljši možni približek pri polinomski časovni omejitvi [8].

Poglejmo si torej drugo obliko izreka PCP, ki je tesno povezana z **zahtevnostjo aproksimacije**. V izreku nastopa problem MAX-3-SAT, ki je definiran kot problem iskanja prireditve vrednosti spremenljivkam Boolovega izraza v obliki 3-CNF, ki maksimizira število izpolnjenih faktorjev. Sledita formalna definicija vrednosti izraza v obliki 3-CNF in aproksimacijskega algoritma za MAX-3-SAT.

Definicija 4. *Za vsak izraz φ v obliki 3-CNF definiramo njegovo* **vrednost** *$val(\varphi)$ kot maksimalni delež faktorjev v izrazu, ki jih lahko pravilno izpolnimo s prireditvijo vrednosti nastopajočim spremenljivkam. Izraz je* **izpolnljiv** *natanko tedaj, ko je $val(\varphi) = 1$.*

Za vsak $\rho \leq 1$ je algoritem A **ρ-aproksimacijski algoritem** *za* MAX-3-SAT*, če A za vsak izraz φ v obliki 3-CNF z m faktorji vrne prireditev vrednosti spremenljivkam, ki pravilno izpolni vsaj $\rho \cdot m \cdot val(\varphi)$ faktorjev izraza φ.*

Zapišimo torej izrek PCP s stališča zahtevnosti aproksimacije, za katerega bomo pokazali, da je ekvivalenten Izreku 1.

Izrek 2. *Obstaja $\rho < 1$, da za vsak jezik $L \in NP$ obstaja polinomsko časovno omejena funkcija f, ki slika nize v izraze v obliki 3-CNF, tako da velja:*

$$x \in L \Rightarrow val(f(x)) = 1$$

$$x \notin L \Rightarrow val(f(x)) < \rho$$

.

Iz izreka takoj sledi naslednje [1]:

Posledica 1. *Obstaja konstanta $\rho < 1$, za katero velja: če obstaja polinomski ρ-aproksimacijski algoritem za MAX-3-SAT, potem je $P = NP$.*

Izrek 2 torej implicira, da lahko (za vsak jezik $L \in NP$) ρ-aproksimacijski algoritem za MAX-3-SAT prilagodimo tako, da bo ugotavljal $x \stackrel{?}{\in} L$.

3.3 Ekvivalenca dokazne in aproksimacijske verzije izreka PCP

Za dokaz ekvivalence Izrekov 1, 2 moramo pred pomožnim Izrekom 3 definirati problem Gap-CSP preko nekoliko razširjene definicije problema CSP.

Definicija 5. *Naj bo q naravno število. Primerek qCSP naj bo množica funkcij (omejitev)* $\varphi_1, ..., \varphi_m : \{0,1\}^n \to \{0,1\}$*, vendar je posamezna omejitev odvisna samo od q vhodnih parametrov.*
Prireditev vrednosti $u \in \{0,1\}^n$ *izpolnjuje omejitev* φ_i*, če je* $\varphi_i(u) = 1$*. Delež omejitev, ki jih u izpolnjuje je* $\frac{\sum_{i=1}^{m} \varphi_i(u)}{m}$ *in* $val(\varphi)$ *naj označuje maksimalni delež za* $u \in \{0,1\}^n$*.* φ *je izpolnljiva, če je* $val(\varphi) = 1$*. Številu q pravimo mestnost primerka* φ*.*

Definicija 6. *Za vsak* $q \in \mathbb{N}$ *in* $\rho < 1$ *definiramo* ρ-GAPqCSP *kot problem določanja, katera od naslednjih možnosti:* $val(\varphi) = 1$ *ali* $val(\varphi) < \rho$*, velja za dani primerek qCSP* φ*.*
Pravimo, da je problem ρ-GAPqCSP *NP-težek za vsak jezik* $L \in NP$*, če obstaja polinomsko časovno omejena funkcija* f*, ki slika nize v (predstavitve) qCSP primerkov, tako da velja:*

$$x \in L \Rightarrow val(f(x)) = 1$$

$$x \notin L \Rightarrow val(f(x)) < \rho$$

.

Izrek 3. *Obstajata konstanti* $q \in \mathbb{N}$ *in* $\rho < 1$*, tako da je* ρ-GAPqCSP *NP-težek problem.*

Dokaz ekvivalence Izrekov 1,2 bomo sedaj izvedli tako, da pokažemo, da je pomožni Izrek 3 ekvivalenten obema.

3.3.1 Izrek 1 ⇒ Izrek 3

Predpostavka je $NP \subseteq PCP[log\, n, 1]$, pokazati pa moramo, da lahko NP-jezike prevedemo na problem ρ-GAPqCSP, ki je NP-težek. Pokazali bomo prevedbo NP-polnega jezika 3-SAT na 1/2-GAPqCSP za neko konstanto q [1].

Po predpostavki za 3-SAT obstaja dokazovalni sistem PCP, v katerem preverjevalnik izvede konstantno število poizvedb, kar označimo s q. Pri tem uporabi $c \cdot log\, n$ naključnih bitov za neko konstanto c. Označimo z $V_{x,r}$ funkcijo, ki pri danem certifikatu π vrne 1, kadar preverjevalnik sprejme certifikat π pri vhodu x in množici naključnih bitov $r \in \{0,1\}^{c \cdot log\, n}$. Pomembno je poudariti, da je izhod $V_{x,r}$ odvisen od **največ** q **bitov** certifikata π.

Za vhodni niz $x \in \{0,1\}^n$ definirajmo množica omejitev $\varphi = \{V_{x,r}\}_{r \in \{0,1\}^{c \cdot log\, n}}$. φ predstavlja primerek qCSP polinomske velikosti. Transformacija certifikata x je polinomska, ker je preverjevalnik V polinomsko časovno omejen. Iz zahtev po popolnosti in zdravju sistema PCP sledi naslednje: če $x \in$ 3-SAT, potem je pripadajoča množica omejitev φ izpolnljiva ($val(\varphi) = 1$), medtem ko v primeru $x \notin$ 3-SAT velja $val(\varphi) \leq 1/2$.

3.3.2 Izrek 3 ⇒ Izrek 1

Predpostavimo, da je ρ-GAPqCSP NP-težek za neki konstanti $q \in \mathbb{N}$, $\rho < 1$. Izvedli bomo pretvorbo v dokazovalni sistem PCP s q poizvedbami, zdravjem ρ in logaritemskim številom naključnih bitov za katerikoli jezik L. Poglejmo, kako bo izgledal preverjevalnik V.

Pri vhodnem nizu x bo najprej izvedel prevedbo $f(x)$ v pripadajoči primerek qCSP $\varphi = \{\varphi_i\}_{i=1}^m$. Certifikat π bo sestavljen iz prireditve vrednosti spremenljivk, ki nastopajo v φ. Preverjevalnik bo izbral naključno omejitev φ_i, za katero bo s q poizvedbami preveril, ali je izpolnjena. Očitno je, da bo preverjevalnik z verjetnostjo 1 sprejel dokaz, če je $x \in L$, medtem ko bo v primeru $x \notin L$ to storil kvečjemu z verjetnostjo ρ [1].

3.3.3 Izrek 2 ⇒ Izrek 3

Implikacija je očitna, saj so izrazi v obliki 3-CNF le poseben primer primerkov 3-CSP.

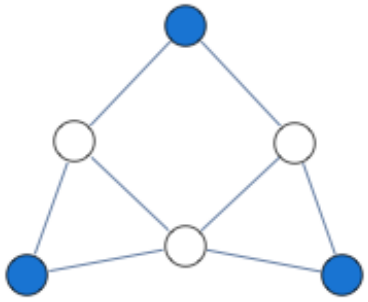
3.3.4 Izrek 3 ⇒ Izrek 2

Prevedba iz primerkov qCSP v izraze v obliki 3-CNF poteka v v dveh stopnjah. Najprej se omejitve izrazijo z Boolovimi izrazi, ki vsebujejo samo negacijo, disjunkcijo in konjunkcijo. Ti trije operatorji tvorijo poln nabor za Boolove izraze, torej je taka prevedba mogoča. V drugem koraku izvedemo pretvorbo v izraz, ki je konjunkcija tročlenih disjunkcij (morda negiranih) spremenljivk. Več detajlov na tem mestu ne bomo navedli, postopek je bolj podrobno opisan v [1].

3.4 Primer aproksimacijskega rezultata na podlagi PCP

V literaturi [1, 8] se kot ilustracija pogosto navaja presenetljiv primer aproksimacije najmanjšega pokritja grafa in največje neodvisne množice. V vsakem grafu $G = \langle V, E \rangle$ velja naslednje: če je S neodvisna množica, je $V \setminus S$ pokritje grafa. Iz omenjenega sledi, da sta natančni rešitvi problemov pravzaprav ena delitev vozlišč grafa na dve množici (Slika 3.1). Poglejmo, kaj se zgodi pri iskanju približnih rešitev.

Označimo z VC velikost najmanjšega pokritja grafa in z I velikost največje neodvisne množice. Velja $VC = n - IS$. ρ-aproksimacija za MAX-INDSET bi hkrati našla pokritje velikosti $n - \rho \cdot IS$. Aproksimacijski faktor za MIN-VERTEX-COVER bi bil $\frac{n-IS}{n-\rho \cdot IS}$, kar je lahko **zelo majhno število**, če največja neodvisna množica vsebuje skoraj vsa vozlišča grafa. Dejansko je situacija obrnjena: za MAX-INDSET **ni** polinomskega ρ-aproksimacijskega algoritma za



Slika 3.1: Največja neodvisna množica (modra vozlišča) in najmanjše pokritje grafa (bela vozlišča).

$\rho < 1$, medtem ko lahko najmanjše pokritje grafa enostavno 1/2-aproksimiramo.

3.4.1 1/2-aproksimacija minimalnega pokritja grafa

Za MIN-VERTEX-COVER obstaja zelo enostaven algoritem, ki najde pokritje, ki je kvečjemu **dvakrat** večje od minimalnega [1].

Postopek je naslednji:

- $S \leftarrow \emptyset$
- izberi povezavo e_1
- dodaj njeni krajišči v_1, v_2 v S
- odstrani vse povezave, v katerih nastopata v_1, v_2
- ponavljaj postopek, dokler graf ni prazen

Vsaki povezavi iz grafa očitno pripada vsaj eno vozlišče iz S, tako da je S pokritje. Zaporedje izbranih povezav je po vozliščih paroma **disjunktno**, torej je $|S| = 2 \cdot |\{e_1, ..., e_k\}|$. Po definiciji mora minimalno pokritje vsebovati vsaj eno od krajišč vsake povezave, torej je opisani algoritem 1/2-aproksimacija za MIN-VERTEX-COVER (nekateri, npr. [8], takemu algoritmu

pravijo 2-aproksimacija).

Zgornji postopek zaradi enostavnosti daje vtis, da minimalnega pokritja ni težko aproksimirati. Vendar iz posledice Leme 1 v naslednjem razdelku sledi, da tudi za najmanjše pokritje obstaja konstanta ρ', za katero ni polinomskega ρ'-aproksimacijskega algoritma [1].

3.4.2 ρ-aproksimacija največje neodvisne množice je NP-težka za vsak $\rho < 1$

Lema 1. *Obstaja polinomsko časovno omejena transformacija f, ki izraze v obliki 3-CNF pretvori v grafe tako, da je za vsak izraz φ njegova slika $f(\varphi)$ graf z n vozlišči. Največja neodvisna množica tega grafa je velikosti $val(\varphi) \cdot \frac{n}{7}$.*

Dokaz leme je klasična NP-prevedba 3-SAT na MAX-INDSET, ki ima želeno lastnost (več v [1]).

Posledica 2. *Če je $P \neq NP$, obstaja konstanta $\rho < 1$, da problema največje neodvisne množice ne moremo ρ-aproksimirati v polinomskem času.*

Naj bo L NP-jezik. Po Izreku 2 lahko odločitveni problem za L prevedemo na aproksimacijo MAX-3-SAT. Še več, po prevedbi bomo dobili izraz φ v obliki 3-CNF, ki je bodisi izpolnljiv bodisi je $val(\varphi) < \rho$ za neko konstanto ρ. Nad tem izrazom lahko uporabimo **transformacijo** iz Leme 1. ρ-aproksimacija za MAX-INDSET bi nam tako omogočila reševanje problema MAX-3-SAT za izraz φ. Zaključimo, da je ρ-aproksimacija za največjo neodvisno podmnožico **NP-težek problem** [1].

Izrek 4. *Za vsak $\rho < 1$ je ρ-aproksimacija največje neodvisne podmnožice NP-težek problem.*

Za dokaz zgornjega izreka moramo pokazati, da je ρ iz Posledice 2 poljubno majhen. Dani graf G z n vozlišči pretvorimo v graf G^k z $\binom{n}{k}$ vozlišči. Posamezno vozlišče predstavlja neko k-podmnožico vozlišč grafa G. Vozlišči v G_k sta

povezani, če unija vozlišč iz G, ki jih predstavljata, tvori neodvisno množico. Največjo neodvisno množico S' v G_k sestavljajo vozlišča, ki ustrezajo k-podmnožicam največje neodvisne množice v grafu G. S' je moči $\binom{|S|}{k}$, kjer $|S|$ označuje število vozlišč največje neodvisne množice v G. Aproksimacijski faktor za MAX-INDSET v grafu G_k je $\frac{\binom{|S|}{k}}{\binom{\rho \cdot |S|}{k}}$, kar je približno ρ^k. Če izberemo dovolj velik k, bomo torej lahko dosegli, da je ρ^k manjši od vsake izbrane konstante. Opisana transformacija zahteva n^k časa, kar je polinomsko za vsak konstantni k [1].

3.4.3 Povzetek

Izrek PCP je prvi pokazal, da je marsikateri NP-problem **težko aproksimirati** do nekega konstantnega faktorja natančno. To je imelo izjemne posledice na področje kombinatorične optimizacije. Izrek PCP je danes osnova praktično vsakega rezultata na področju zahtevnosti aproksimacije [8].

3.4.4 Primerjava obeh verzij izreka PCP [1]

Tabela 1. *Vzporednice med dokaznim in aproksimacijskim vidikom izreka PCP.*

Dokazni vidik		**Aproksimacijski vidik**
Preverjevalnik PCP (V)	$\Longleftrightarrow$	primerek CSP (φ)
certifikat (π)	$\Longleftrightarrow$	prireditev vrednosti spremenljivkam (u)
dolžina certifikata	$\Longleftrightarrow$	število spremenljivk (n)
število poizvedb (q)	$\Longleftrightarrow$	mestnost omejitev (q)
število naključnih bitov (r)	$\Longleftrightarrow$	logaritem števila omejitev ($log\, m$)
vrednost konstante ρ	$\Longleftrightarrow$	največja dovoljena vrednost ρ za $val(\varphi)$
Izrek 1	$\Longleftrightarrow$	Izrek 2, Izrek 3

Poglavje 4

Skica dokaza izreka PCP

4.1 Skica algebraičnega dokaza

Izrek 1 pravi, da lahko razlikujemo pravilen in napačen dokaz o pripadnosti neke besede jeziku $L \in NP$ že, če iz certifikata preberemo konstantno število bitov. Iz tega lahko sklepamo, da je certifikat zapisan na tak način, da so morebitne napake prisotne praktično povsod in jih bomo z omenjenim načinom preverjanja lahko odkrili.

Ob zahtevi po bolj očitni prisotnosti napak morda najprej pomislimo na **kode za odpravljanje napak**. To so preslikave, ki poljubna različna niza x, y preslikajo v kodi $C(x), C(y)$, ki se gotovo razlikujeta v npr. petini istoležnih bitov. Problem je, da običajne kode nimajo lastnosti, ki jo pri certifikatu nujno potrebujemo, t.j. **lokalne preverljivosti**. Zato je potrebno uporabiti t.i. **lokalno preverljive kode (LTC)**, ki predstavljajo jedro konstrukcije PCP [8].

4.1.1 Lokalno preverjanje

O lastnostih neke velike, morda neizmerljive celote želimo velikokrat sklepati na podlagi nekega majhnega (lokalnega) vzorca. Okoljevarstveniki zajamejo

nekaj decilitrov vode in lahko sklepajo na onesnaženost reke. Bistveno je, da je opazovana lastnost **stabilna** - da se ohranja na vzorcih oz. podmnožicah.

Za skupino omejitev, ki definira problem, pravimo, da množico rešitev definira na stabilen način, če so približne rešitve problema le perturbacije neke točne rešitve. Na primer, skupina omejitev nad Boolovimi spremenljivkami je stabilna, če se prireditev, ki izpolnjuje veliko omejitev, od neke prireditve, ki izpolni vse, razlikuje le v vrednosti nekaterih spremenljivk. Posebej nas zanima stabilnost rešitve, t.j. za kolikšen del rešitve moramo preveriti ali izpolnjuje neko omejitev. Če moramo na primer pogledati le neko konstantno število bitov, potem rečemo, da je množica omejitev lokalno preverljiva. Poglejmo si formalno definicijo.

Definicija 7. *Lastnost binarnih nizov $L \subset \{0,1\}^*$ je lokalno preverljiva s k poizvedbami in napako ϵ, če obstaja množica k-mestnih omejitev $C_1, ..., C_m$ nad n biti, tako da za vsak $x \in \{0,1\}^n$ velja:*

- *če $x \in L$, potem je $Pr_{j\in[m]}$ [x izpolnjuje omejitev C_j]$= 1$*
- *če $x \notin L$, potem je $Pr_{j\in[m]}$ [x izpolnjuje omejitev C_j]$\leq \epsilon$*

Sedaj lahko definiramo lokalno preverljive kode.

Definicija 8. *Lokalno preverljiva koda je koda za odpravljanje napak $C : \{0,1\}^k \to \{0,1\}^n$, katere slika $Im(C) = \{C(x)|x \in \{0,1\}^k\}$ je lokalno preverljiva.*

Pri lokalno preverljivih kodah imamo dve zahtevi. Želimo, da je relativna razdalja med kodnimi besedami velika. Vrednost nekega bita naj bo tako odvisna le od konstantnega števila drugih bitov v besedi. Hkrati lokalna preverljivost zahteva, da so korelacije med biti lokalne. Med redkimi konstrukcijami, ki obema zahtevama ugodijo v zadostni meri, je **Hadamardova koda**.

Definicija 9. *Hadamardova koda je koda za odpravljanje napak* $H : \{0,1\}^k \to \{0,1\}^{2^k}$*, ki niz* $a = (a_1, ..., a_k) \in \{0,1\}^k$ *zakodira v niz* $H(a) = H(a_1, ..., a_k) \in \{0,1\}^{2^k}$*, ki predstavlja resničnostno tabelo za linearno funkcijo* f*, definirano kot* $f(x) = \sum_{i=1}^{k} a_i x_i$*.*

Koda ima minimalno relativno razdaljo $1/2$ in je lokalno preverljiva (dokaz sledi iz lokalne preverljivosti linearnosti, glej [8]). Problem je njena eksponentna dolžina. Bolj učinkovite konstrukcije so precej bolj zapletene in jih tukaj ne bomo navajali, omenjena koda pa nam bo služila za ilustracijo problemov pri kodiranju PCP.

4.1.2 Poenostavljen opis dokazovalnega postopka

Preverjevalnik za jezik linearnega CSP

Za osnovo vzemimo primer, ko želimo konstruirati $(log\, n, 1)$-preverjevalnik za jezik CSP, definiran z množico linearnih in afinih omejitev nad npr. dvema spremenljivkama in naša naloga je, da preverimo ali dana prireditev vrednosti a izpolnjuje vse navedene omejitve. Tako definiran problem ni zadosti kompleksen, da bi s prevedbami zajel vse NP-probleme, vendar pa nam bo služil za koristno referenco pri splošnem primeru.

Prireditev vrednosti lahko zakodiramo s Hadamardovo kodo. Zakodirana prireditev bo predstavljala certifikat, do katerega bo dostopal preverjevalnik PCP. Ker je Hadamardova koda lokalno preverljiva, je preverjanje, ali je certifikat veljavno zakodirana prireditev in ali izpolnjuje vse omejitve, enostavno [8].

Prilagoditve pri splošnem primeru (za vse NP-jezike)

- Dovoliti moramo bolj splošen tip predikatov kot v primeru linearnega CSP. Izkaže se, da zadoščajo predikati stopnje 2. Jezik CSP s predikati stopnje 2 je NP-poln. Z njim lahko predstavimo vse NP-jezike.

- Problem je tudi že omenjena eksponentna dolžina Hadamardove kode. Kodiranje PCP mora biti **učinkovito** in preslikava niza dolžine n v niz dolžine 2^n gotovo ni ustrezna. Kodiranje poteka preko polinomov višjih stopenj (npr. $d = log\, n$) nad velikimi polji (velikosti npr. $(log\, n)^{O(1)}$). To razreši problem eksponentne dolžine kodiranja, vendar poveča število poizvedb na stopnjo polinoma, kar v omenjenem primeru $d = log\, n$ ni več konstantno. Pomanjkljivost se odpravi v več zahtevnih korakih, ki jih tukaj ne navajamo.

- Pokazati je potrebno tudi, kako poteka preverjanje ali zakodiran izraz res predstavlja **veljaven dokaz**, t.j. takega, ki izpolnjuje dane omejitve. Podrobnosti so navedene v [3, 2].

4.2 Skica kombinatoričnega dokaza

Kombinatorični dokaz temelji na aproksimacijski verziji izreka PCP, torej Izreku 2. Za jezik L v izreku izberemo NP-poln jezik 3-COL, z njim bomo preko prevedb zajeli vse NP-jezike.

Podobno kot pri algebraičnem dokazu je naša naloga, da prisotnost napak v dokazu o 3-obarvljivosti **razširimo** po celotnem certifikatu, da bo ta lokalno preverljiv. Pri problemu 3-obarvljivosti so napake enobarvne povezave, t.j. take, katerih krajišči sta enako pobarvani. Definirajmo minimalni delež enobarvnih povezav.

Definicija 10. *Naj bo dan graf G. Z $unsat(G)$ označimo minimalni delež enobarvnih povezav po vseh 3-barvanjih grafa G.*

$$unsat(G) = \min_{c:V\to\{1,2,3\}} Pr_{(u,v)\in E}[c(u) = c(v)]$$

Očitno je, da je graf **3-obarvljiv** natanko tedaj, ko je $unsat(G) = 0$. Za graf, ki ga ne moremo 3-obarvati, pa velja $unsat(G) \geq 1/|E|$. $unsat(G)$ ustreza

vrednosti $1 - \rho$ iz Izreka 2. Želimo si, da bi vrednost ρ preko nekega postopka zmanjšali in posledično lažje razločevali med pravilnim 3-barvanjem in barvanjem z zelo malo enobarvnimi povezavi, morda le eno. Iščemo torej algoritem, ki bo preko zaporedja transformacij grafa

$$G \to G_1 \to G_2 \to ... \to G'$$

povečal vrednost $unsat(G)$.

4.2.1 Transformacija grafa, ki poveča vrednost $unsat(G)$

Na koraku $G_i \to G_{i+1}$ bomo vrednost povečali za najmanj dvakrat (dokler ne dosežemo neke zgornje meje), razen v primeru $unsat(G) = 0$, ker želimo da se lastnost 3-obarvljivosti ohrani. Po $log\, n$ korakih bo vrednost $unsat(G)$ postala neka (dovolj velika) konstanta, torej bomo lahko po seriji transformacij iz začetnega 3-barvanja z malo enobarvnimi povezavami dobili barvanje, kjer bo takih povezav veliko.

Skica transformacije [8]

Opišimo približen potek transformacije iz grafa G_i v G_{i+1}. Zaradi preglednosti ju preimenujmo v G in H. Transformacija je sestavljena iz dveh korakov, t.i. **zbiranja** in **razprševanja**:

- Najprej graf G transformiramo v G' in 3-barvanje grafa G zakodiramo kot k-barvanje G' za nek konstantni $k > 3$. G' naj ima ista vozlišča kot G, barva nekega vozlišča $v \in G'$ pa naj nosi informacijo o **barvah sosedov** tega vozlišča v G. Omejitve pri barvanju G' niso običajne omejitve za k-barvanje, ampak preverjajo **neprotislovnost** lokalnega barvanja. Taka omejitev preverja, ali se podatki o barvi nekega skupnega soseda dveh vozlišč ujemajo.

  Potem v G' dodamo povezave med vozlišči, ki so med sabo oddaljena za

manj kot npr. 100 povezav [8].

Po konstrukciji bo G' ustrezal navedenim omejitvam, če je bil originalni graf G 3-obarvljiv. Težje je dokazati obratno: če G' izpolnjuje omejitve, jih tudi G. V primeru neizpolnjevanja omejitev mora transformacija vrednost *unsat* vsaj **podvojiti**.

- V drugem koraku k-barvanje grafa G' pretvorimo v 3-barvanje grafa H, ne da bi preveč pokvarili (zmanjšali) vrednost *unsat*. Ta korak je izveden s pomočjo zahtevnih tehnik, opisanih v [9].

Poglavje 5

Zgodovinski pregled

5.1 Interaktivni dokazovalni sistemi [14]

Verjetnostno preverjanje dokazov se je razvilo kot veja interaktivnih dokazovalnih sistemov z omejitvami. Interaktivne dokazovalne sisteme so definirali Goldwasser, Micali in Rackoff v **osemdesetih letih** [11].

Definicija 11. *Interaktiven dokazovalni sistem sestavljata verjetnostni polinomsko omejen preverjevalnik in neomejen dokazovalnik. Komunicirata tako, da si polinomsko mnogokrat izmenjata sporočila. Dokazi pravilnih trditev morajo biti sprejeti z verjetnostjo 1, kakršenkoli dokaz napačne trditve pa mora biti zavrnjen z verjetnostjo vsaj* $1/2$*. Z IP označimo razred vseh jezikov, za katere obstaja interaktivni dokazovalni sistem.*

Leta 1988 so Ben-Or, Goldwasser, Killan in Wigderson v [6] definirali še interaktivne dokazovalne sisteme z **več dokazovalniki**.

Definicija 12. *Razred MIP sestavljajo tisti jeziki, za katere obstajajo interaktivni dokazovalni sistemi z več dokazovalniki, ki med seboj ne morejo komunicirati.*

Najpomembnejša rezultata na področju interaktivnih dokazovalnih sistemov

sta iz leta 1990: Izrek 5, ki ga je dokazal Adi Shamir [16] in Izrek 6 (avtorji Babai, Fortnow in Lund [5]).

Izrek 5. $IP = PSPACE.$

Izrek 6. $MIP = NEXP.$

Ker sta zgornja rezultata povezana z razredi visoko v hierarhiji kompleksnosti, se je naravno pojavilo vprašanje, kako bi lahko na podoben način definirali tudi tiste, ki nas v praksi bolj zanimajo, predvsem razred NP. Jasno je bilo, da bo v sistem potrebno dodati **omejitve**, ni pa bilo takoj jasno, kakšne [14].

Anne Condon je leta 1991 pokazala, da razred NP lahko definiramo tudi kot razred jezikov z interaktivnimi dokazovalnimi sistemi, v katerih je preverjevalnik omejen na **logaritemski prostor** in ima samo **enosmerni bralni dostop** do dokaza [7]. Babai, Fortnow, Levin in Szegedy so istega leta pokazali, da lahko NP definiramo tudi kot razred jezikov, za katere lahko dokaze preverja **polilogaritemsko časovno omejen** preverjevalnik [4].

5.2 Verjetnostno preverjanje dokazov

Največji napredek je bil dosežen, ko sta Arora in Safra leta 1992 definirala sistem, v katerem je preverjevalnik hkrati omejen pri **številu naključnih bitov** in **številu dostopov** do certifikata [3].

Definicija 13. *PCP[r(n),q(n)] je razred jezikov, katerim pripadnost lahko dokažemo z dokazovalnim sistemom PCP, kjer preverjevalnik uporabi* $O(r(n))$ *naključnih bitov in izvede* $O(q(n))$ *poizvedb v dokazu. Pravilni dokazi trditev morajo biti sprejeti z verjetnostjo 1, kakršenkoli napačen dokaz pa mora biti zavrnjen z verjetnostjo vsaj* $1/2$.

5.2.1 Razvoj verjetnostnega preverjanja dokazov glede na parametre [17]

Faza 1

V prvo fazo spadata dve očitni ugotovitvi. $NP = PCP[0, poly(n)]$ smo že navedli v prvem poglavju. Velja tudi $NP = PCP(log\, n, poly(n))$. Dodatni naključni biti torej **ne povečajo razreda jezikov**, pač pa postane preverjanje bistveno bolj **učinkovito** [17].

Faza 2

Prvi pomembnejši rezultat na področju PCP ni bil povezan z razredom NP. Babai, Fortnow in Lund so v [5] pokazali, da je $NEXP = PCP[poly(n), poly(n)]$. To je bil **prelomen razultat**, ki je pokazal, da lahko z uporabo naključnosti zmanjšamo število poizvedb za **polilogaritemski faktor**. Podoben rezultat $NP \subseteq PCP[poly\, log\, n, poly\, log\, n]$ je bil za razred NP pokazan v [4]. Ostal je še problem **točne karakterizacije NP**, saj je bila ugotovljena le navedena vsebovanost.

Faza 3

Arora in Safra sta predstavila karakterizacijo $NP = PCP[O(log\, n), o(log\, n)]$ [3]. Tehnike, ki sta jih uporabila, so predstavljale **temelj vseh poznejših izboljšav**. Izrek PCP kot ga poznamo danes $NP = PCP[O(log\, n), O(1)]$, so dokazali Arora, Lund, Motwani, Sudan in Szegedi v [2].

Opisani rezultati so bili zares presenetljivi. Dokazano je bilo, da je **število poizvedb neodvisno od dolžine certifikata!** Obe omejitvi (naključni biti, poizvedbe) sta tesni do konstantnega faktorja ($NP \subseteq PCP[o(log\, n), o(log\, n)]$ bi pomenilo $NP = P$ [3]).

Faza 4

V četrto fazo Madhu Sudan uvršča rezultate, ki so dosegli meje bodisi pri **številu poizvedb** bodisi pri **številu naključnih bitov**. Polischuk in Spielman sta pokazala, da je $SAT \in PCP[(1+\epsilon)log\, n, O(1)]$ za vsak $\epsilon > 0$ [15]. Håstad je v [13] pokazal, da za vsak $\epsilon > 0$ velja $NP = PCP[O(log\, n), 3)]$ pri parametrih popolnosti $1-\epsilon$ in zdravju $1/2$. Kasneje so Guruswami, Lewin, Sudan in Trevisan v [12] dokazali še rezultat s popolnostjo 1 in zdravjem $1/2+\epsilon$.

5.2.2 Kombinatorični dokaz izreka PCP

Zadnji večji dosežek na področju verjetnostnega preverjanja dokazov je kombinatorični dokaz (skica v četrtem poglavju) Irit Dinur iz leta 2005 [9]. Pomemben je predvsem zato, ker na izrek PCP gleda s področja zahtevnosti aproksimacije, kjer so njegove posledice trenutno najbolj uporabne. Tudi sam postopek dokazovanja je bolj **eleganten** in ne zahteva več toliko poglabljanja v detajle kot originalni algebraični dokaz [14].

Poglavje 6

Sklepne ugotovitve

Izrek o verjetnostnem preverjanju dokazov je **temelj ugotovitev na področju zahtevnosti aproksimacije**. V diplomskem delu smo izrek najprej predstavili s stališča drugačnih dokazovalnih sistemov, iz katerih je zgodovinsko izšel. Pokazali smo **ekvivalenco med obema verzijama** izreka in navedli primer aproksimacijskega rezultata na podlagi izreka PCP. Opisali smo **glavne ideje pri algebraičnem in kombinatoričnem dokazu**, bolj poglobljena obravnava bi lahko predstavljala primeren izziv za podiplomskega študenta z zanimanji na področju računske zahtevnosti.

V zgodovinskem pregledu smo navedli, da je izrek PCP povezan s področjem **interaktivnih dokazovalnih sistemov**, ki bi ga lahko bolj podrobno predstavili v kakem drugem diplomskem delu. V pričujočem smo se ga le dotaknili in navedli motivacijski primer takega dokazovanja. Poznavanje interaktivnih dokazovalnih sistemov omogoča za lažje razumevanje izreka PCP.

Aproksimacijska verzija izreka o verjetnostnem preverjanju dokazov je povezana s pojmom **stabilnosti matematičnih sistemov**. Dinur v [8] navaja, da obstajajo povezave med izrekom PCP in rezultati o stabilnosti na področjih kot so diskretna Fourierova analiza, geometrija, verjetnost in aritmetična kombinatorika. Zanimivo bi bilo raziskati in predstaviti ugotovitve na teh področjih.

Dodatek A

Definicije

Definicija 1. Verjetnostni Turingov stroj (PTM) *je Turingov stroj z dvema funkcijama prehodov δ_0 in δ_1. Delovanje PTM* M *pri vhodu x je sestavljeno iz izbiranja funkcij prehodov δ_0 oz. δ_1, pri čemer ima na vsakem koraku vsaka verjetnost* $1/2$*, da bo izbrana.*

Izhod stroja M *je bodisi 0 ali 1 (zavrne oz. sprejme vhod). Z $M(x)$ označimo slučajno spremenljivko, ki opisuje izhod stroja* M*.*

Pravimo, da ima M *časovno zahtevnost $T(n)$, pri čemer je $T : \mathbb{N} \to \mathbb{N}$, če se pri vsakem vhodu x dolžine n ustavi po kvečjemu $T(n)$ korakih, ne glede na naključno izbiranje funkcije prehodov.*

Definicija 2. Turingov stroj z enakovrednim dostopom (RAM TM) *je Turingov stroj z neskončnim poljem A, ki je inicializirano s simboli B. Do polja dostopa preko posebnega delovnega traku, ki ga imenujemo naslovni trak. Stroj ima v abecedi tudi dva posebna simbola, R in W, ter dodatno stanje q_{access}. Ko stroj preide v stanje q_{access}, stori naslednje:*

- *če naslovni trak vsebuje $\llcorner i \lrcorner R$ (kjer $\llcorner i \lrcorner$ označuje število i v binarnem zapisu), potem prebere vrednost $A[i]$ in jo zapiše v celico zraven simbola* R*.*

- *če naslovni trak vsebuje* $\llcorner i \lrcorner W\sigma$ *(kjer je* σ *nek simbol iz abecede), potem nastavi vrednost* $A[i]$ *na* σ*.*

Definicija 3. *Naj bo* $V = \{v_1, ..., v_m\}$ *množica spremenljivk, ki zavzemajo vrednosti v končni abecedi* Σ*.* (n-mestna) omejitev $C = (\psi, i_1, ..., i_n)$ *je sestavljena iz n-terice indeksov* $i_1, ..., i_n \in \{1, ..., m\}]$ *in predikata* $\psi : \Sigma^n \to \{0, 1\}$*. Omejitev* C *je izpolnjena s prireditvijo* $a : V \to \Sigma$ *natanko tedaj, ko je* $\psi(a(v_{i_1}), ..., a(v_{i_m})) = 1$*.*

Definicija 4. Problem izpolnljivosti Boolovega izraza v obliki 3-CNF (3-SAT) *je problem izpolnjevanja omejitev, kjer je* $\Sigma = \{0, 1\}$*,* $m = 3$ *in je predikat zapisan v obliki tročlene disjunkcije, npr.* $\psi(a, b, c) = \neg a \vee b \vee \neg c$*.*

Definicija 5. Problem 3-obarvljivosti (3-COL) *je ponavadi podan kot problem na grafih: za dani graf najdi 3-barvanje vozlišč* $\chi : V \to \{1, 2, 3\}$*, tako da ne bo noben par sosednjih vozlišč pobarvan z isto barvo.*
Na ta problem lahko gledamo kot na poseben primer problema izpolnjevanja omejitev, kjer je $\Sigma = \{1, 2, 3\}, m = 2$*, spremenljivke pa predstavljajo posamezna vozlišča. Vsaka povezava predstavlja omejitev s predikatom* $\psi(a, b) = 1$ *natanko tedaj, ko* $a \neq b$*, pri čemer sta* a *in* b *barvi krajišč te povezave.*

Definicija 6. Prevedba *med jezikoma* $L_1 \subseteq \{0, 1\}^*$ *in* $L_2 \subseteq \{0, 1\}^*$ *je preslikava* $r : \{0, 1\}^* \to \{0, 1\}^*$ *z naslednjima lastnostima: izračunljiva je v polinomskem času in zagotavlja* $x \in L_1 \iff r(x) \in L_2$*.*

Definicija 7. *Grafa* G_1 *in* G_2 *sta* izomorfna*, če obstaja preslikava* $h : V(G_1) \to V(G_2)$ *z naslednjima lastnostima:*

- h *je bijekcija*
- $uv \in E(G_1) \iff h(u)h(v) \in E(G_2)$*.*

Definicija 8. *Naj bo dan neusmerjen graf* $G = \langle V, E \rangle$. Najmanjše pokritje grafa *je najmanjša podmnožica vozlišč* $V' \subseteq V$*, za katero velja:*

$$uv \in E \Rightarrow (u \in V' \vee v \in V').$$

Definicija 9. *Naj bo dan neusmerjen graf* $G = \langle V, E \rangle$ *in naj za množico* $U \subseteq V$ *velja naslednje: Nobeni vozlišči v* U *nista sosednji. Množico z največ elementi, ki izpolnjuje zgornji pogoj, imenujemo* največja neodvisna množica.

Definicija 10. Hammingova razdalja $Dist(x, y)$ *med nizoma* $x, y \in \{0,1\}^n$ *je število istoležnih bitov, v katerih se razlikujeta.* Relativna Hammingova razdalja *dist(x,y) je* $\frac{Dist(x,y)}{n}$.
Koda za odpravljanje napak *z relativno razdaljo* δ *je preslikava* $C : \{0,1\}^k \to \{0,1\}^n$*, za katero velja naslednje:*

$$x, y \in \{0,1\}^k, x \neq y \Rightarrow dist(C(x), C(y)) \geq \delta$$

.

Literatura

[1] S. Arora, B. Barak, *Computational Complexity: A Modern Approach*, New York: Cambridge University Press, 2009.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, "Proof verification and intractability of approximation problems", *Journal of the ACM*, zv. 45, št. 3, 1998.

[3] S. Arora, S. Safra, "Probabilistic Checking of Proofs: A new Characterization of NP", *Journal of the ACM*, zv. 45, št. 1, 1998.

[4] L. Babai, L. Fortnow, L. Levin, M. Szegedy, "Checking computations in polylogarithmic time", v zborniku *23rd ACM Symposium on Theory of Computing*, New Orleans, 1991.

[5] L. Babai, L. Fortnow, C. Lund, "Non-deterministic exponential time has two-prover interactive protocols", *Computational Complexity*, št. 1, 1991.

[6] M. Ben-Or, S. Goldwasser, J.Kilian, A. Widgerson, "Multi-prover interactive proofs: How to remove intractability assumptions", v zborniku *20th ACM Symposium on Theory of Computing*, ZDA, 1988.

[7] A. Condon, "The complexity of space bounded interactive proof systems", *Complexity Theory: Current Research*, S. Homer, U. Schöning, K. Ambos-Spies (ured.), New York: Cambridge University Press, 1993.

[8] I.Dinur, "Probabilistically Checkable Proofs and Codes", v zborniku *International Congress of Mathematicians*, Hyderabad, 2010.

[9] I.Dinur, "The PCP Theorem by Gap Amplification", *Journal of the ACM*, zv. 54, št. 3, 2007.

[10] U.Feige, S. Goldwasser, L. Lovász, S. Safra, M. Szegedy, "Approximating clique is almost NP-complete", *Journal of the ACM*, zv. 43, št. 2, 1996.

[11] S. Goldwasser, S.Micali, C. Rackoff, "The knowledge complexity of interactive proof systems", *SIAM Journal of Computing*, zv. 18, št. 1, 1989.

[12] V. Guruswami, D.Lewin, M. Sudan, L. Trevisan, "A tight characterization of NP with 3-query PCPs", v zborniku *39th Annual Symposium on Foundations of Computer Science*, Palo Alto, 1998.

[13] J. Håstad, "Some optimal inapproximability results", v zborniku *29th ACM Symposium on Theory of Computing*, ZDA, 1997.

[14] R. O'Donnell, "A History of the PCP Theorem", *neobjavljeno*. Dostopno na: www.cs.washington.edu/education/courses/533/05au/pcp-history.pdf

[15] A. Polischuk, D. Spielman, "Nearly-linear size holographic proofs", v zborniku *26th ACM Symposium on Theory of Computing*, Montreal, 1994.

[16] A. Shamir, "IP=PSPACE", *Journal of the ACM*, zv. 39, št. 4, 1992.

[17] M. Sudan, "Advanced complexity theory", *Zapiski s predavanj*, 2002. Dostopni na: http://people.csail.mit.edu/madhu/ST02/scribe/all.pdf